

UNIVERSITÀ POLITECNICA DELLE MARCHE
FACOLTÀ DI INGEGNERIA

Dipartimento di Ingegneria dell'Informazione
Corso di Laurea in Ingegneria Informatica e dell'Automazione



TESI DI LAUREA

**Progettazione e implementazione in Python di un sistema per la
gestione di un'impresa edile**

**Design and implementation in Python of a system for managing a
construction company**

Relatore

Prof. Domenico Ursino

Candidato

Domenico La Porta

ANNO ACCADEMICO 2023-2024

*Sii forte e coraggioso; non ti spaventare e non ti sgomentare,
perché il Signore, il tuo Dio, sarà con te dovunque andrai.*

Giosuè 1:9, La Sacra Bibbia

Sommario

Nell'ambito dell'automatizzazione del settore informativo di un'azienda, una delle principali attività da realizzare è il passaggio verso una gestione dei processi svolta attraverso l'utilizzo di un software. Da parte dell'azienda ci sono diverse alternative, che vanno dall'acquisto di un sistema già esistente sul mercato al commissionare ad un team di sviluppatori un prodotto su misura, a partire da una lista di requisiti concordati. In questo lavoro di tesi ci troviamo nel secondo scenario; in particolare, vogliamo creare un software per la gestione della "Impresa Edile La Porta Tommaso", che intende proiettarsi verso una gestione automatizzata dei propri processi aziendali.

Keyword: Ingegneria del Software, Processi Aziendali, Sistema Informatico, Dati, Impresa Edile, Analisi dei Requisiti, Database, Python, MySQL.

Introduzione	1
1 Descrizione del contesto e specifica dei requisiti	3
1.1 Descrizione del contesto di riferimento	3
1.2 Intervista	4
1.2.1 Intervista al titolare	4
1.2.2 Intervista alla segretaria	7
1.2.3 Intervista al commercialista	8
1.2.4 Intervista al fornitore	9
1.3 Analisi dei processi interni	10
1.4 Descrizione in linguaggio naturale	10
1.5 Glossario dei termini	12
1.6 Diagramma dei sistemi	12
1.7 Analisi dei requisiti	13
1.7.1 Requisiti funzionali	14
1.7.2 Requisiti non funzionali	20
1.8 Diagrammi dei casi d'uso	20
1.8.1 Gestione dei lavori e dei clienti	21
1.8.2 Gestione contratti e dipendenti	25
1.8.3 Gestione della sicurezza	27
1.8.4 Gestione del magazzino	30
1.8.5 Gestione dei fornitori	34
1.8.6 Gestione della contabilità	35
1.8.7 Calcolo statistiche	39
2 Progettazione del database	42
2.1 Scelta database	42
2.1.1 Perché utilizzare un database	42
2.1.2 Perché scegliere un modello relazionale	43
2.2 Entità fondamentali	44
2.3 Diagramma E/R	45
2.4 Vincoli sui dati	46
2.5 Traduzione verso il modello relazionale	47
2.5.1 Traduzione	47
2.5.2 Vincoli di riferimento	48

3	Progettazione della componente applicativa	49
3.1	Diagramma dell'architettura	49
3.2	Diagramma delle classi	52
3.2.1	Sezione "Lavori e Clienti"	53
3.2.2	Sezione "Dipendenti"	54
3.2.3	Sezione "Sicurezza sul lavoro"	54
3.2.4	Sezione "Magazzino"	54
3.2.5	Sezione "Fornitori"	55
3.2.6	Sezione "Contabilità"	55
3.2.7	Sezione "Statistiche"	55
3.3	Diagrammi di attività	55
3.3.1	Sezione "Lavori e Clienti"	56
3.3.2	Sezione "Dipendenti"	56
3.3.3	Sezione "Sicurezza sul lavoro"	56
3.3.4	Sezione "Magazzino"	57
3.3.5	Sezione "Fornitori"	57
3.3.6	Sezione "Contabilità"	57
3.3.7	Sezione "Statistiche"	58
3.3.8	Filtraggio	58
3.4	Diagrammi di sequenza	58
3.4.1	Create	59
3.4.2	Read	64
3.4.3	Update	67
3.4.4	Delete	68
3.4.5	Altre operazioni	72
4	Implementazione e testing	85
4.1	Tecnologie utilizzate	85
4.1.1	Python	85
4.1.2	Git & GitHub	87
4.1.3	PyCharm	89
4.1.4	SQLite3	91
4.1.5	PyQt5	93
4.1.6	Altre librerie	105
4.1.7	HTML	109
4.2	Testing	110
4.2.1	Test sui lavori	110
4.2.2	Test sui dipendenti	112
4.2.3	Test sull'aggiornamento del magazzino	112
4.2.4	Test sulle visite mediche	113
4.2.5	Test sui locali	114
5	Manuale utente	116
5.1	Autenticazione	116
5.2	Home Page	116
5.3	Sezione "Lavori"	118
5.4	Sezione "Preventivi"	122
5.5	Sezione "Dipendenti"	124
5.6	Sezione "Clienti"	128
5.7	Sezione "Corsi di sicurezza"	130
5.8	Sezione "Visite mediche"	133

5.9 Sezione "Fornitori"	137
5.10 Sezione "Materie prime"	140
5.11 Sezione "Locali"	144
5.12 Sezione "Veicoli"	146
5.13 Sezione "Entrate"	153
5.14 Sezione "Uscite"	156
5.15 Sezione "Bilanci"	158
5.16 Sezione "Statistiche"	160
5.17 Video Tutorial	162
Conclusioni	167
Bibliografia	169
Sitografia	170
Ringraziamenti	171

Elenco delle figure

1.1	Diagramma dei processi interni	11
1.2	Diagramma dei sistemi	14
1.3	Suddivisione dei requisiti in funzionali e non funzionali	14
1.4	Elenco dei requisiti funzionali	15
1.5	Diagramma dei casi d'uso relativo alla gestione dei lavori e dei clienti	21
1.6	Diagramma dei casi d'uso relativo alla gestione dei contratti e dei dipendenti	25
1.7	Diagramma dei casi d'uso relativo alla gestione della sicurezza	28
1.8	Diagramma dei casi d'uso relativi alla gestione del magazzino	30
1.9	Diagramma dei casi d'uso relativo alla gestione dei fornitori	34
1.10	Diagramma dei casi d'uso relativo alla gestione della contabilità	36
1.11	Diagramma dei casi d'uso relativo al calcolo delle statistiche	40
2.1	Esempio di diagramma E/R	46
3.1	Mappa dell'architettura del sistema	50
3.2	Mappa dell'architettura del sistema (parte 1)	50
3.3	Mappa dell'architettura del sistema (parte 2)	51
3.4	Mappa dell'architettura del sistema (parte 3)	51
3.5	Diagramma delle classi del sistema	52
3.6	Diagramma delle classi del sistema (parte 1)	53
3.7	Diagramma delle classi del sistema (parte 2)	53
3.8	Diagramma delle classi del sistema (parte 3)	54
3.9	Diagramma delle classi del sistema (parte 4)	55
3.10	Diagramma delle classi del sistema (parte 5)	56
3.11	Diagramma delle classi relativo alle sottosezioni "Lavori" e "Clienti"	57
3.12	Diagramma delle classi relativo alla sottosezione "Preventivi"	58
3.13	Diagramma delle classi relativo alla sezione "Dipendenti"	59
3.14	Diagramma delle classi relativo alla sezione "Sicurezza"	60
3.15	Diagramma delle classi relativo alla sottosezione "Materie Prime"	61
3.16	Diagramma delle classi relativo alle sottosezioni "Locali" e "Veicoli"	61
3.17	Diagramma delle classi relativo alla sezione "Fornitori"	62
3.18	Diagramma delle classi relativo alle sottosezioni "Entrate" e "Uscite"	63
3.19	Diagramma delle classi relativo alla sottosezione "Bilanci"	64
3.20	Diagramma delle classi relativo alla sezione "Statistiche"	65
3.21	Diagrammi di attività relativi ai casi d'uso della sezione "Lavori e Clienti"	66

3.22	Diagrammi di attività relativi ai casi d'uso della sezione "Dipendenti"	66
3.23	Diagrammi di attività relativi ai casi d'uso della sottosezione "Corsi di Sicurezza"	67
3.24	Diagrammi di attività relativi ai casi d'uso della sottosezione "Visite Mediche"	68
3.25	Diagrammi di attività relativi ai casi d'uso della sottosezione "Materie Prime"	69
3.26	Diagrammi di attività relativi ai casi d'uso della sottosezione "Locali"	69
3.27	Diagrammi di attività relativi ai casi d'uso della sottosezione "Veicoli"	70
3.28	Diagrammi di attività relativi ai casi d'uso della sezione "Fornitori"	70
3.29	Diagrammi di attività relativi ai casi d'uso della sezione "Contabilità"	71
3.30	Diagrammi di attività relativi ai casi d'uso della sezione "Statistiche"	71
3.31	Diagramma di attività generalizzato relativo ai casi d'uso che riguardano il filtraggio	72
3.32	Diagramma di sequenza relativo al caso d'uso "Inserisci bilancio"	73
3.33	Diagramma di sequenza relativo al caso d'uso "Inserisci fornitore"	73
3.34	Diagramma di sequenza relativo al caso d'uso "Inserisci lavoro"	74
3.35	Diagramma di sequenza relativo al caso d'uso "Inserisci veicolo"	75
3.36	Diagramma di sequenza relativo al caso d'uso "Visualizza dettaglio cliente"	76
3.37	Diagramma di sequenza relativo al caso d'uso "Filtra clienti per nome"	76
3.38	Diagramma di sequenza relativo al caso d'uso "Visualizza lista clienti"	77
3.39	Diagramma di sequenza relativo al caso d'uso "Visualizza lista veicoli a partire da un locale"	77
3.40	Diagramma di sequenza relativo al caso d'uso "Modifica lavoro"	78
3.41	Diagramma di sequenza relativo al caso d'uso "Modifica fornitore"	78
3.42	Diagramma di sequenza relativo al caso d'uso "Elimina bilancio"	79
3.43	Diagramma di sequenza relativo al caso d'uso "Elimina cliente"	79
3.44	Diagramma di sequenza relativo al caso d'uso "Elimina fornitore"	80
3.45	Diagramma di sequenza relativo al caso d'uso "Elimina lavoro"	80
3.46	Diagramma di sequenza relativo al caso d'uso "Contatta cliente a partire da un lavoro"	81
3.47	Diagramma di sequenza relativo al caso d'uso "Contatta fornitore"	81
3.48	Diagramma di sequenza relativo ai casi d'uso "Visualizza magazzino" e "Con- suma materie prime"	82
3.49	Diagramma di sequenza relativo all'interazione dell'utente con la lista della spesa e al caso d'uso "Conferma ordine"	83
3.50	Diagramma di sequenza relativo al caso d'uso "Scarica preventivo"	84
4.1	Home Page di GitHub	89
4.2	Repository del progetto	90
4.3	UI PyCharm	90
4.4	Funzionalità di Git integrate in PyCharm	91
5.1	Schermata di Login	117
5.2	Schermata per il cambio delle credenziali	117
5.3	Schermata relativa alla Home Page del sistema	118
5.4	Schermata relativa alla lista dei lavori	119
5.5	Schermata di pop-up che permette il filtraggio dei lavori per cliente	119
5.6	Schermata per l'inserimento di un nuovo lavoro (sezione "Lavoro")	120
5.7	Schermata per l'inserimento di un nuovo lavoro (sezione "Cliente")	121
5.8	Schermata relativa ai dettagli del lavoro selezionato (sezione "Lavoro")	121
5.9	Schermata relativa ai dettagli del lavoro selezionato (sezione "Cliente")	122
5.10	Schermata relativa alla lista dei preventivi	123
5.11	Schermata di pop-up che permette il filtraggio dei preventivi per cliente	123

5.12	Schermata per l'inserimento di un nuovo preventivo (sezione "Preventivo")	124
5.13	Schermata per l'inserimento di un nuovo preventivo (sezione "Cliente")	125
5.14	Schermata relativa ai dettagli del preventivo selezionato (sezione "Preventivo")	125
5.15	Schermata relativa ai dettagli del preventivo selezionato (sezione "Cliente")	126
5.16	Schermata relativa alla lista dei dipendenti	126
5.17	Schermata per l'inserimento di un nuovo dipendente (sezione "Dati Anagrafici")	127
5.18	Schermata per l'inserimento di un nuovo dipendente (sezione "Contratto")	128
5.19	Schermata relativa ai dettagli del dipendente selezionato (sezione "Dati Anagrafici")	129
5.20	Schermata relativa ai dettagli del dipendente selezionato (sezione "Contratto")	129
5.21	Schermata relativa alla lista dei clienti	130
5.22	Schermata di pop-up che permette il filtraggio dei clienti per nome	131
5.23	Schermata di pop-up che permette il filtraggio dei clienti per cognome	131
5.24	Schermata relativa ai dettagli del cliente selezionato	132
5.25	Avviso "Cliente impegnato"	132
5.26	Schermata relativa alla lista dei corsi di sicurezza sul lavoro	133
5.27	Schermata per l'inserimento di un nuovo corso di sicurezza sul lavoro	134
5.28	Schermata relativa ai dettagli del corso di sicurezza sul lavoro selezionato	134
5.29	Schermata relativa alla lista delle visite mediche	135
5.30	Schermata di pop-up che permette il filtraggio delle visite mediche per dipendente	136
5.31	Schermata per l'inserimento di una nuova visita medica	136
5.32	Schermata relativa ai dettagli della visita medica selezionata	137
5.33	Schermata relativa alla lista dei fornitori	138
5.34	Schermata di pop-up che permette la ricerca di un fornitore per nome	138
5.35	Schermata di pop-up che permette il filtraggio dei fornitori per tipologia	139
5.36	Schermata di inserimento di un nuovo fornitore	139
5.37	Schermata relativa ai dettagli del fornitore selezionato	140
5.38	Schermata relativa alle scorte di materie prime presenti in magazzino	141
5.39	Schermata di pop-up che permette il consumo della materia prima selezionata	141
5.40	Schermata relativa all'acquisto di nuove scorte di materie prime (lista della spesa vuota)	142
5.41	Schermata di pop-up che permette la ricerca per nome di una specifica materia prima	143
5.42	Schermata di pop-up che permette l'inserimento nella lista della spesa della materia prima selezionata	143
5.43	Schermata relativa all'acquisto di nuove scorte di materie prime (lista della spesa non vuota)	144
5.44	Schermata relativa alla lista dei locali	145
5.45	Schermata di pop-up che permette il filtraggio dei locali per tipologia	145
5.46	Schermata per l'inserimento di un nuovo locale	146
5.47	Schermata relativa ai dettagli del locale selezionato	147
5.48	Avviso "Locale occupato"	147
5.49	Menù che permette di selezionare il tipo di pagamento in uscita che riguarda il locale selezionato	148
5.50	Menù che permette di selezionare il tipo di pagamento in entrata che riguarda il locale selezionato	149
5.51	Schermata relativa alla lista dei veicoli	149
5.52	Schermata di pop-up che permette il filtraggio dei veicoli per tipologia	150
5.53	Schermata di pop-up che permette il filtraggio dei veicoli per locale	150

5.54	Schermata per l'inserimento di un nuovo veicolo	151
5.55	Avviso "Lista locali vuota"	151
5.56	Schermata relativa ai dettagli del veicolo selezionato	152
5.57	Menù che permette di selezionare il tipo di pagamento in uscita che riguarda il veicolo selezionato	152
5.58	Menù che permette di selezionare il tipo di pagamento in entrata che riguarda il veicolo selezionato	153
5.59	Schermata relativa alla lista delle entrate	154
5.60	Schermata di pop-up che permette il filtraggio delle entrate per provenienza .	154
5.61	Schermata di pop-up che permette il filtraggio delle entrate per data	155
5.62	Schermata per inserimento di una nuova entrata	155
5.63	Schermata relativa ai dettagli dell'entrata selezionata	156
5.64	Schermata relativa alla lista delle uscite	157
5.65	Schermata di pop-up che permette il filtraggio delle uscite per destinazione .	157
5.66	Schermata di pop-up che permette il filtraggio delle uscite per data	158
5.67	Schermata per l'inserimento di una nuova uscita	159
5.68	Schermata relativa ai dettagli dell'uscita selezionata	159
5.69	Schermata relativa alla lista dei bilanci	160
5.70	Schermata di pop-up per l'inserimento di un nuovo bilancio	161
5.71	Avviso "Bilancio già presente per l'anno selezionato"	161
5.72	Schermata di pop-up relativa ai dettagli del bilancio selezionato	162
5.73	Schermata relativa alle statistiche	163
5.74	Grafico dipendenti-stipendi	163
5.75	Schermata di pop-up che mostra la percentuale dei lavori pagati	164
5.76	Grafico lavori-importi	164
5.77	Schermata relativa alla lista delle entrate e delle uscite nel periodo di tempo selezionato dall'utente	165
5.78	Grafico bilanci-saldi complessivi	165
5.79	Schermata pop-up che mostra la percentuale dei preventivi accettati	166
5.80	Schermata Web che mostra il video tutorial di come utilizzare il software . . .	166
5.81	Foto in cantiere da solo	174
5.82	Foto in cantiere con nonno Mimì, papà e mio fratello Michele	175
5.83	Foto in cantiere con papà e i miei due fratelli Michele e Marco	175
5.84	Foto sul cestello con mio fratello Michele	176
5.85	Foto in cantiere con mio fratello Michele	177

Elenco delle tabelle

1.1	Glossario dei termini	13
2.1	Traduzione verso il modello relazionale	48
2.2	Vincoli di riferimento	48
4.1	Classi del modulo QtCore utilizzate nel progetto	94
4.2	Classi del modulo QtGui utilizzate nel progetto	95
4.3	Classi del modulo Qt utilizzate nel progetto	96
4.4	Descrizione di ulteriori librerie/moduli utilizzati nel progetto	106

4.1	Esempio di codice Python	86
4.2	Esempio di definizione di una classe	87
4.3	Esempio di utilizzo di <code>sqlite3</code>	92
4.4	Costruzione della "Home Page" del sistema	95
4.5	Costruzione di un'interfaccia per la visualizzazione di liste	98
4.6	Costruzione di un'interfaccia per l'inserimento di nuovi oggetti	99
4.7	Costruzione di un'interfaccia con switch tra due sezioni	101
4.8	Metodi utili per la costruzione di interfacce	104
4.9	Esempio di utilizzo di <code>pathlib</code>	105
4.10	Esempio di utilizzo di <code>webbrowser</code>	105
4.11	Esempio di utilizzo di <code>re</code>	106
4.12	Esempio di utilizzo di <code>codicefiscale</code>	106
4.13	Esempio di utilizzo di <code>numpy</code>	106
4.14	Esempio di utilizzo di <code>pyplot</code>	107
4.15	Esempio di utilizzo di <code>QPdfWriter</code> e di <code>QPainter</code>	107
4.16	Metodo per la visualizzazione del tutorial	109
4.17	Pagina web per la visualizzazione del tutorial	109
4.18	Test sui lavori	111
4.19	Test sui dipendenti	112
4.20	Test sulle materie prime	112
4.21	Test sulle visite mediche	113
4.22	Test sui locali	114

In Italia sono più di 5 milioni le imprese attive nel primo trimestre del 2024. Ogni impresa può essere divisa nei sottosectori aziendale, organizzativo, informativo e informatico. Il nostro lavoro si colloca nell'ambito dell'automatizzazione del settore informativo verso quello informatico; l'obiettivo è di aiutare un'impresa specifica nella gestione dei propri processi aziendali attraverso lo sviluppo di un software personalizzato che sappia lavorare con la grande mole di dati con cui l'impresa stessa si interfaccia ogni giorno.

La maggioranza delle imprese medio-piccole, come quella con cui ci interfacceremo nel nostro lavoro, non riesce, oppure non vuole, per una serie di motivi, comprendere i vantaggi che l'utilizzo di un software porterebbe nella gestione dei propri processi aziendali e, di conseguenza, non investe più di tanto in questo settore.

Tuttavia, qualsiasi azienda voglia essere gestita in modo corretto è obbligata ad avvalersi di un efficace sistema informatico; ogni scelta diversa è anacronistica e antieconomica. Questo non significa che un'azienda che non investe sull'automatizzazione della propria parte informativa non possa vivere ed esistere – lo dimostra il fatto che la maggioranza delle aziende faccia un utilizzo parziale o nullo della gestione informatizzata dei dati – vuol dire, invece, che tutte le aziende potrebbero trarne molti vantaggi.

Una volta comprese le opportunità di crescita dell'efficienza nell'ambito della gestione dei processi a seguito dell'introduzione di un software, procederemo con la realizzazione di "EdilGest", il prodotto finale che arricchirà il sistema informatico dell'"Impresa Edile La Porta Tommaso", portando una novità mai sperimentata prima dalla suddetta azienda.

Così come accade negli altri settori ingegneristici – l'esempio più chiaro è, per restare in tema, l'ingegneria edile – ci sono alcune fasi da seguire: dovremo capire, innanzitutto, cosa realizzare; successivamente studieremo come lavorare; dopo ci occuperemo della realizzazione vera e propria del sistema; infine, testeremo quanto prodotto, per assicurarci che quanto fatto rispecchi tutti i requisiti. L'ingegneria del software propone queste quattro fasi prevedendo i seguenti passi: specifica e analisi dei requisiti, progettazione del sistema, implementazione del codice, testing del prodotto. Pensare di non seguire questo programma equivarrebbe a costruire una casa partendo, ad esempio, prima dalla manodopera e poi, se avanza tempo, dedicarsi alla fase di progettazione. Sarebbe irrealistico! Sulla base di questo modello, svilupperemo, quindi, la nostra tesi, la quale si struttura così come di seguito specificato:

1. Nel Capitolo 1 approfondiremo lo studio sul campo di applicazione del sistema, ovvero l'edilizia, e ci occuperemo della specifica e dell'analisi dei requisiti, in modo da comprendere gli scenari di utilizzo che il committente desidera.

2. Nel Capitolo 2 affronteremo la fase di progettazione del database, partendo dalle entità fondamentali e dal diagramma E/R, fino ad arrivare alla traduzione verso il modello relazionale.
3. Nel Capitolo 3 ci occuperemo della progettazione del sistema, prima dal punto di vista strutturale, poi da quello comportamentale, al fine di avere delle buone fondamenta su cui appoggiare il lavoro delle fasi successive.
4. Nel Capitolo 4 passeremo alla realizzazione vera e propria del software, approfondendo le tecnologie e gli approcci utilizzati per implementare il codice del sistema.
5. Nel Capitolo 5 testeremo le componenti del sistema, in modo da assicurarci che i requisiti ricevuti dal cliente vengano soddisfatti e che il software faccia correttamente il lavoro per cui è stato realizzato. Inoltre, illustreremo, attraverso un manuale utente, tutte le possibili funzionalità del sistema, al fine di rendere chiara la User Experience e far comprendere a chi utilizzerà il software come navigare all'interno di esso.

Descrizione del contesto e specifica dei requisiti

In questo primo capitolo analizzeremo il contesto in cui lavoreremo, senza quindi entrare ancora nella parte tecnica. Come prima cosa cercheremo di avere una panoramica generale sul mondo dell'edilizia per poi scendere nello specifico e capire che tipo di digital attitude hanno le imprese italiane. Successivamente si entrerà a contatto con la ditta di interesse attraverso un'intervista che spazierà su diversi livelli aziendali in modo da avere una panoramica generale su ogni settore. In seguito, dopo aver opportunamente raggruppato le tante informazioni ricavate dalle varie interviste in una descrizione fatta in linguaggio naturale, si inizierà a parlare di software e funzionalità attraverso un documento che elenca tutti i requisiti del sistema ovvero tutte le operazioni che l'utente deve poter eseguire. Infine, attraverso i primi diagrammi UML, nello specifico quelli dei casi d'uso, entreremo nella parte tecnica (senza però ancora parlare di codice) passando dai requisiti ai casi d'uso ovvero descrizioni dettagliate delle interazioni tra l'utente e il sistema stesso, finalizzate a raggiungere un obiettivo specifico.

1.1 Descrizione del contesto di riferimento

In Italia sono più di 5 milioni le imprese attive nel primo trimestre del 2024. Ogni impresa può essere divisa in più sottosistemi:

- *sistema aziendale*: l'obiettivo è quello di produrre valore;
- *sistema organizzativo*: si occupa della gestione delle risorse;
- *sistema informativo*: riguarda la gestione di informazioni di interesse;
- *sistema informatico*: riguarda la porzione automatizzata di un sistema informativo.

Il sistema informativo (con la sua parte automatizzata) è di particolare interesse in questo progetto, in quanto si nota subito l'enorme mole di dati che popola tale settore di un'azienda.

Un sistema informativo può essere, a sua volta, diviso in tre sottosistemi: direzionale, gestionale e operativo, a cui si aggiunge anche la parte relativa ai processi fisici. Ciascuno di questi settori si occupa di enormi quantità di dati: la parte direzionale si occupa di dati relativi alla strategia, interni (budget, dati storici) ed esterni (mercato, dati macroeconomici); la parte gestionale si occupa di dati relativi al controllo (saldi mensili, produzione mensile), la parte operativa si occupa di dati relativi alle operazioni (distinta base, ore di presenza dei dipendenti), la parte dei processi fisici si occupa di dati transazionali (ordine, bolla prelievo/consegna, cartellino dipendenti).

Per gestire tutti questi dati spesso le aziende, soprattutto quelle medio-piccole (come quella di cui ci occuperemo in questo progetto), non riescono/vogliono investire sul proprio sistema informatico; i motivi sono molteplici (ne elenchiamo tre):

- scarse conoscenze e competenze di tipo informatico;
- mancanza di tempo (è un falso problema, ma molto sentito);
- mancanza di presa di coscienza degli enormi vantaggi derivanti da una gestione informatizzata.

Tuttavia, qualsiasi azienda voglia essere gestita in modo corretto è obbligata ad avvalersi di un efficace sistema informatico; ogni scelta diversa è anacronistica e antieconomica. Questo non significa che un'azienda che non investe sull'automatizzazione della propria parte informativa non possa vivere ed esistere – lo dimostra il fatto che la maggioranza delle aziende abbia un utilizzo parziale o nullo della gestione informatizzata dei dati – vuol dire, invece, che tutte le aziende potrebbero trarne molti vantaggi.

In Italia sono presenti circa 780mila imprese del settore costruzioni, il 65,3% impegnate in lavori di costruzione specializzati, il 33,1% nella costruzione di edifici e l'1,6% in opere di ingegneria civile. Dal punto di vista geografico la popolazione del settore edile si concentra per il 29% nel Nord-Ovest, per il 20,6% nel Meridione, per il 20,9% nell'Italia centrale, per il 20,4% nel Nord-Est e per il restante 9,1% nelle Isole. Le prime dieci regioni con il maggior numero di realtà imprenditoriali del settore sono la Lombardia (17,2%), il Lazio (10,1%), la Campania (8,8%), l'Emilia-Romagna (8,6%), il Veneto (8,1%), il Piemonte (8%), la Toscana (6,9%), la Sicilia (6,3%), la Puglia (5,5%) e la Liguria (3,5%).

Il settore edile nazionale è in continuo mutamento ed è caratterizzato per lo più da piccole e piccolissime realtà imprenditoriali, che costituiscono la stragrande maggioranza della popolazione analizzata.

Sotto il profilo occupazionale le realtà del settore edile a fine 2022 contavano circa 854mila dipendenti, dato in crescita rispetto allo stesso periodo del 2021 (+6,1%) e a quello del 2020 (+21,3%). Tre stabilimenti su cinque, il 62,9% del totale, impiega meno di due dipendenti, mentre il 10,2% ne impiega più di cinque.

A livello di forma legale, come è facile immaginare, è predominante la ditta individuale. Infatti, il settore è formato per il 57,9% da imprese individuali, per il 33,3% da società di capitali e per l'8,6% da società di persone. Tra le società di capitali è da segnalare la percentuale delle società a responsabilità limitata (20,8%), delle società a responsabilità limitata semplificata (7,9%) e di quelle con un unico socio (2%). Le società per azioni, invece, sono solo lo 0,2% delle imprese totali.

Le realtà del settore si caratterizzano per una bassissima dimestichezza con il canale digitale. Infatti, circa otto imprese su dieci, il 76,9%, fanno registrare una digital attitude bassa. Al contrario, sono rari i casi di imprese di costruzioni con un punteggio alto (0,8%), ovvero di realtà che mostrano una buona affinità con le tecnologie TIC, una propensione all'adozione di processi digitali e una maggiore reattività alle campagne di digital marketing.

1.2 Intervista

1.2.1 Intervista al titolare

Come primo approccio per la realizzazione del software, abbiamo ritenuto opportuno effettuare un'intervista al titolare dell'azienda. Ci siamo posti come obiettivo quello di riuscire a comprendere l'ambito in cui opera l'impresa e di avere alcune informazioni utili sulle prime entità da analizzare successivamente.

- *Buongiorno Tommaso. Siamo qui per farle alcune domande riguardanti la sua impresa, di cosa si occupa in generale, come riesce nella gestione dei processi aziendali, quanto ampio risulta essere il sistema informativo dell'azienda e in che percentuale tale sistema viene automatizzato. Partiamo dall'inizio. Di cosa si occupa la sua impresa?*

Buongiorno ragazzi. La mia è un'impresa edile artigiana, ci occupiamo principalmente di costruzioni edili e ristrutturazioni per conto di terzi o in proprio.

- *Entrando più nello specifico, quali sono i servizi che offre l'impresa al cliente?*

Al cliente offriamo diversi servizi come, ad esempio, costruzioni per civili abitazioni, ristrutturazioni murarie, restauri vari e rifacimento di fabbricati civili e industriali, pubblici e privati, lavori industriali, stradali, idrici, fognanti, di bonifica e movimento terra, pubblici e privati, illuminazione pubblica urbana ed extraurbana. Inoltre, offriamo anche carpenteria in legno, cemento armato, tamponatura, intonaco, pavimenti, cappotti termici, impianti idrici, fognanti, rivestimenti, tinteggiatura, demolizioni e scavi.

- *Ci riesce a spiegare, anche senza entrare troppo nello specifico, i processi aziendali di tali servizi?*

Certamente! (La spiegazione dei processi aziendali risulta essere molto lunga, per questo, anche per osservare come il flusso dei dati venga influenzato dalle procedure e come le varie parti dell'azienda interagiscano tra loro, abbiamo deciso di riassumere il tutto in uno schema dei processi aziendali riportato in Figura 1.1).

- *Riguardo i lavori effettuati, quale metodo utilizzate per la gestione di tale settore dell'azienda?*

Per ogni lavoro abbiamo delle cartelline in cui inseriamo tutti i fogli contenenti i dati di maggiore interesse.

- *Di che tipo di dati parliamo?*

Ci sono dati che riguardano l'identificazione e il classamento dell'appartamento in cui stiamo lavorando; nello specifico parliamo, rispettivamente, di foglio, particella sub e categoria, classe e consistenza. Abbiamo, anche, informazioni riguardo le date previste ed effettive di inizio e fine lavoro, oltre che, ovviamente, la denominazione, il cliente e l'importo. Per un'analisi riguardo i pagamenti completati per ogni lavoro siamo soliti anche evidenziare i lavori i cui clienti hanno interamente effettuato il pagamento dell'importo.

- *Come vengono gestiti gli operai? In media quanti ne risultano assunti in un anno? Quante ore lavorative giornaliere prevede il contratto? Quanti giorni a settimana?*

In media in un anno lavorano con me 5 operai, 8 ore al giorno, 5 giorni a settimana. Per quanto riguarda ulteriori dettagli puoi provare a rivolgerti al commercialista. È lui che si occupa dei contratti e delle buste paga.

- *Supponiamo che l'impresa sia provvista di un magazzino. Come gestisce e divide dal punto di vista pratico tutte le scorte di materie prime?*

L'impresa è provvista di alcuni locali adibiti a magazzino in cui teniamo tutte le materie prime, gli attrezzi, le macchine e i mezzi intestati alla ditta. Per la gestione, anche in questo caso, sarà meglio rivolgersi alla mia segretaria, in quanto è lei che se ne occupa.

- *Ci sa dire qualcosa riguardo l'identificazione a livello nazionale della sua impresa?*

Per esercitare un'attività economica sotto forma di impresa (individuale o societaria) in Italia è obbligatoria l'iscrizione al Registro Imprese, tenuto presso la Camera di Commercio della provincia ove ha sede l'azienda. La Visura camerale è un documento

informativo ufficiale con tutti i dettagli riguardanti l'impresa iscritta; sono riportati aspetti legali e anagrafici dell'impresa, la denominazione, la forma giuridica, la sede legale, il codice fiscale, il tipo di attività svolta e molti altri elementi relativi agli organi di amministrazione e alle cariche sociali. Nel Registro Imprese, a livello nazionale, ogni impresa è univocamente identificata da un Codice Fiscale e da una Partita IVA. Codice Fiscale e Partita IVA, tranne eccezioni, sono sempre coincidenti per le società, e sempre differenti per le imprese individuali. Presso l'Ufficio del Registro Imprese è istituito il Repertorio delle notizie Economiche e Amministrative (REA), con il quale la Camera di Commercio acquisisce e utilizza ogni altra notizia, sia di carattere economico che di natura statistica e amministrativa, anche ove non sia prevista ai fini dell'iscrizione al Registro Imprese.

- *Ci parla un po' dei fornitori/aziende terze che lavorano con lei? Riesce in qualche modo a tenere traccia di tutte le aziende con cui interagisce?*

Siamo a contatto settimanalmente sia con fornitori sia con aziende che lavorano con noi. Abituamente ci rivolgiamo sempre alle stesse aziende, quindi riusciamo a tenere bene a mente i loro riferimenti; tuttavia, il fatto che a volte per esigenza ci rivolgiamo anche ad altri potrebbe portarci a valutare di conservare un elenco con tutti i dati più importanti. Siccome, ripeto, siamo molto a contatto con fornitori e altre ditte, potreste valutare l'idea di andare ad intervistare anche loro per approfondire la questione.

- *Può parlarci della sicurezza sul lavoro?*

Certamente! Io sono il responsabile della sicurezza sul lavoro, quindi mi occupo in tutto e per tutto di questo settore. Siamo tenuti ad effettuare periodicamente visite mediche, essenziali per approvare l'idoneità o meno di ogni dipendente. Inoltre, è obbligatorio per tutti i dipendenti, compreso il sottoscritto, frequentare corsi di formazione e informazione in materia di sicurezza e salute sui luoghi di lavoro.

- *Adesso una domanda riguardante il sistema informatico dell'azienda. Qual è il livello di automatizzazione dei processi? Come vengono gestiti i diversi dati su vendite, clienti, fornitori, magazzino?*

Tutti i dati inerenti vengono mantenuti fisicamente da un commercialista per una durata di circa cinque anni. Dopo tale periodo tutto viene recuperato da me e mantenuto in ufficio, tuttavia senza avvalermi di nessun mezzo informatico.

- *Crede che le tornerebbe utile incrementare il livello di automatizzazione dei processi all'interno della propria azienda?*

Per quanto la mia sia un'azienda medio-piccola, i processi sono comunque di una grandezza non trascurabile. Il fatto di tracciare poco o nulla, e quel poco tenerlo su semplici fogli, potrebbe portare ad errori dovuti a dimenticanze, oltre a un'enorme difficoltà nel recuperare qualcosa in tempi brevi nel caso di eventuali ricerche. Certamente mi tornerebbe molto utile incrementare il livello di automatizzazione dell'azienda iniziando proprio dallo sviluppo di un software per la gestione dei processi e dei dati di maggiore importanza.

- *Ci ha detto in precedenza che lei è il responsabile della sicurezza sul lavoro. Riguardo alla gestione di questo settore come crede che un software potrebbe aiutarla?*

Sicuramente potrebbe essere utile avere un inventario di tutti i corsi di sicurezza sul lavoro, oltre che una lista di tutte le visite mediche effettuate; renderebbe la gestione più efficiente anche un'ipotetica ricerca delle visite mediche per ogni dipendente.

- *Quali sono i dati di maggiore interesse riguardo le visite mediche e i corsi di sicurezza sul lavoro?*
Per quanto riguarda le visite mediche è importante ricordare la data della visita, oltre che, ovviamente, il risultato e il dipendente che ha effettuato tale visita. Invece i corsi di sicurezza, siccome non sono riservati ad un singolo dipendente, ma a tutta la squadra, me compreso, non necessitano di un dato che evidenzia chi ha seguito tale corso; è importante, però, che vengano messi in luce la materia trattata, la data, l'ora di inizio e di fine, l'attività svolta, che può essere formazione, intervento oppure formazione e intervento, la durata e il docente.
- *Perfetto Tommaso, le risposte che ci ha dato ci hanno soddisfatto a sufficienza. Un'ultima richiesta: ci riesce ad individuare all'interno dell'impresa o anche al di fuori alcune figure alle quali possiamo porre ulteriori domande per approfondire i singoli campi che abbiamo analizzato oggi?*
Come detto in precedenza, potreste intervistare la segretaria per quanto riguarda l'accoglienza dei clienti, la documentazione e la gestione del magazzino; inoltre, potete sentire cosa hanno da dirvi anche il commercialista sulla contabilità e uno dei miei fornitori su dettagli riguardanti la sua impresa e il suo modo di lavorare.
- *Grazie tante per l'enorme disponibilità. Le auguriamo ancora una buona giornata.*
Figuratevi, per così poco. Auguro anche a voi buon lavoro e buona giornata.

1.2.2 Intervista alla segretaria

Dopo aver analizzato le risposte datoci dal signor Tommaso, ci siamo resi conto, anche sotto suo consiglio, di dover approfondire alcune questioni e per questo siamo stati obbligati a contattare anche altre figure che lavorano per e con l'impresa; in particolare, abbiamo dovuto intervistare la segretaria, il commercialista e un fornitore.

- *Buon giorno Caterina. Abbiamo intervistato Tommaso chiedendo informazioni e dettagli riguardanti la sua impresa. Alla fine, abbiamo chiesto se ci fossero ulteriori persone con cui potevamo parlare e ci ha mandato da lei. Per prima cosa volevamo chiederle di cosa si occupa.*
Salve ragazzi. Io mi occupo principalmente dell'accoglienza dei clienti, delle stipule di preventivi e della gestione del magazzino.
- *In cosa consiste la stipula di un preventivo?*
La stipula di un preventivo consiste nella stesura di alcuni dati inerenti al lavoro commissionato, come informazioni del cliente, dettagli dei lavori da effettuare, prezzi, data di stipula del preventivo stesso; inoltre, per puro fine statistico, siamo soliti evidenziare i preventivi accettati in modo che l'azienda riesca a tenere traccia, appunto, di quanti preventivi in percentuale vengano poi effettivamente accettati dalla clientela.
- *C'è un modo efficace in cui vengono gestiti i preventivi?*
I preventivi vengono mantenuti cartacei nell'ufficio. Siamo privi, tuttavia, di un inventario in cui è possibile visualizzare i dettagli fondamentali.
- *Tommaso ci ha riferito che lei è la prima figura dell'impresa con cui il cliente si interfaccia. Ci sa dire qualcosa di più anche su come viene gestita la clientela? Vengono tracciati alcuni dati e, se sì, in che modo?*
Non c'è un vero e proprio inventario; tuttavia, attraverso i preventivi e i lavori effettuati, teniamo traccia dei dati anagrafici e identificativi dei clienti.

- *Di quali dati parliamo nello specifico?*

Per ogni cliente memorizziamo il nome, il cognome, il codice fiscale, la residenza, la data di nascita, il telefono e l'e-mail; quest'ultima è importante in quanto è il mezzo attraverso cui io, che sono la segretaria, in rappresentanza dell'azienda, invio comunicazioni di ogni genere al cliente.

- *Ci è stato anche riferito che lei si occupa della gestione del magazzino, ma che quest'ultima risulta molto faticosa in quanto viene fatta con semplici appunti su carta; addirittura, alcune cose non vengono segnate e vengono ricordate soltanto a mente. Ci sa dire come si trova lei con questa tecnica?*

Effettivamente è molto dispendiosa. Molte volte capita di dimenticare di aggiornare i dati, o addirittura di smarrire fogli che risultano importanti, e tocca ricominciare da zero.

- *Crede che le tornerebbe utile un metodo più efficace e sicuro per la gestione del magazzino, e non solo?*

Sicuramente! Purtroppo, il fatto di trovarci in un'azienda medio-piccola ha sempre portato ad un certo rifiuto per l'automatizzazione di questi processi, vuoi per mancanza di tempo o per mancata consapevolezza degli enormi vantaggi che questa può portare. Credo proprio che ormai siamo arrivati ad un punto in cui non si possa più continuare ad andare avanti con fogli sparsi, carta e penna e memoria, nella peggiore delle ipotesi, ed è giusto guardare avanti e proiettarci verso lo sviluppo del sistema informatico nella nostra azienda.

- *Un software per la gestione di un'impresa edile come potrebbe aiutarla in questo settore?*

Riguardo al magazzino e alle materie prime, sicuramente deve essere in grado di mostrarmi tutte le scorte a disposizione dell'azienda; poi deve permettermi, ovviamente, di modificarne la quantità, quindi parliamo di acquisto e consumo di materie prime. Inoltre, credo sia importante anche tenere traccia sia di tutti i locali posseduti, tra i quali ce ne sono di adibiti a magazzino, sia dei mezzi intestati all'impresa.

- *Quali sono i dati di maggiore interesse per quanto riguarda i locali e i veicoli?*

Abbiamo la necessità di memorizzare i locali attraverso l'indirizzo, la data di acquisto e il tipo; quest'ultimo campo è importante per comprendere di che tipo di locale stiamo parlando; in particolare potremmo avere a che fare con un garage, un ufficio, un appartamento oppure un magazzino. I veicoli, invece, vengono memorizzati attraverso targa, classificazione, locale in cui si trovano, data di acquisto e costo. Potrebbe tornare utile, riguardo la domanda fatta in precedenza, che il software possa mostrarmi tutti i veicoli presenti in ogni singolo locale.

- *Perfetto, abbiamo informazioni a sufficienza. La ringraziamo e le auguriamo ancora buona giornata.*

Grazie a voi per le domande. Vi auguro buona giornata e un buon proseguimento per il lavoro.

1.2.3 Intervista al commercialista

- *Buongiorno Emanuele. Abbiamo avuto il piacere di fare alcune domande a Tommaso, proprietario dell'impresa edilizia a suo nome. Con le sue risposte è stato abbastanza esaustivo; tuttavia ci ha consigliato di parlare con lei su eventuali dettagli riguardanti il suo campo. Se possibile ci piacerebbe farle alcune domande partendo da cosa si occupa lei nello specifico.*

Buongiorno a voi. Io mi occupo della contabilità dell'impresa: entrate, uscite, analisi di bilancio, ma anche gestione del personale, per quanto riguarda la stesura dei contratti e le buste paga.

- *Per quanto riguarda i contratti con i dipendenti, come sono strutturati? Viene tenuta traccia degli operai in qualche modo?*

Nei contratti vengono specificati lo stipendio, il ruolo, la data di stipula, la durata, l'orario ed eventuali extra; ovviamente si tiene memoria anche di dati che riguardano il dipendente, come nome, cognome, codice fiscale, data di nascita, residenza, e-mail e recapito telefonico, oltre che la matricola, un numero che permette di identificare univocamente ogni persona che lavora per l'azienda. I contratti vengono mantenuti per una durata di circa cinque anni dal sottoscritto sia in maniera cartacea sia in un database. Dopo tale periodo il cartaceo viene consegnato al titolare mentre i dati online vengono cancellati.

- *Come viene gestita la comunicazione con i dipendenti?*

Per ogni dipendente viene memorizzata l'e-mail in modo che io, il titolare o la segretaria, sempre comunque in rappresentanza dell'impresa, possiamo inviare comunicazioni di qualsiasi tipo.

- *Quali sono le principali fonti di entrata e uscita dell'impresa? In che maniera viene gestito questo settore dell'impresa?*

Le principali fonti d'entrata sono certamente i guadagni che derivano dai lavori effettuati; tuttavia, l'azienda ottiene introiti anche attraverso vendita/affitto di locali, veicoli, mezzi aziendali posseduti. Per quanto riguarda i costi troviamo i costi fissi, le tasse, le buste paga dei dipendenti, le spese per la materia prima e per i servizi da imprese terze, i costi per l'acquisto di attrezzatura e macchine... Riguardo al metodo di gestione di questo settore dell'azienda posso dirvi che il titolare ha a disposizione una lista di tutte le entrate e le uscite con dettagli quali data di pagamento, importo, causale e provenienza.

- *Per quanto riguarda la stipula del bilancio, in cosa consiste? Tra i tanti dati presenti, quali sono quelli di maggiore interesse? Anche in questo caso le chiedo se viene tenuta traccia di tali dati.*

In un'analisi di bilancio sono tanti i dati che vengono studiati, tra stato patrimoniale e conto economico. Tuttavia, ci si sofferma nel nostro caso soltanto sul totale delle entrate, sul totale delle uscite e sul saldo complessivo. È di particolare interesse per l'azienda anche avere una panoramica sull'andamento anno per anno del saldo complessivo. Anche in questo caso il titolare ha a disposizione un elenco di tutte le analisi di bilancio effettuate e può, quando ha necessità, visionare anche gli andamenti del saldo complessivo per avere una panoramica generale su tale settore.

- *Ottimo. Abbiamo le idee molto più chiare. La ringraziamo per la disponibilità e le auguriamo una buona giornata.*

Buona giornata e buon lavoro anche a voi ragazzi.

1.2.4 Intervista al fornitore

- *Buona giornata Giovanni. Parlando con Tommaso negli scorsi giorni, nonostante le sue risposte siano state abbastanza esaustive, ci siamo resi conto, anche sotto il suo consiglio, che parlare con lei ci avrebbe portato ad avere maggiori delucidazioni sulla figura del fornitore/erogatore*

di servizi. Innanzitutto, volevamo chiederle con quali informazioni viene identificata la vostra impresa, se sono delle informazioni comuni con tutte le imprese o se ci sono delle differenze.

Salve ragazzi. Per quanto riguarda l'identificazione della nostra impresa, a livello nazionale è pressoché identica a quella delle altre imprese, compresa quella di Tommaso, che, sicuramente, vi avrà già spiegato come funziona. Tuttavia, per farci "pubblicità" tra la clientela utilizziamo semplicemente i dati che trovate sul nostro cartellino, quindi nome dell'impresa, titolare, tipo di attività, indirizzo, e-mail e telefono.

- *Per quanto riguarda le materie prime avete un listino prezzi? Ci sono dei dati di particolare interesse che un acquirente dovrebbe conoscere riguardanti tali materie prime?*

Abbiamo un listino prezzi; tuttavia non ci sono dati particolari oltre ovviamente al nome del prodotto e al prezzo.

- *Per quanto riguarda le fatture, come sono strutturate?*

In una fattura potete trovare i nomi delle materie prime acquistate, la quantità, il prezzo unitario e il prezzo totale.

- *La ringraziamo per le informazioni che ci ha dato. Sono state molto esaustive. Le auguriamo buona giornata.*

Grazie a voi ragazzi. Buona giornata e buon lavoro.

1.3 Analisi dei processi interni

Come già accennato prima, la grande quantità di informazioni che siamo riusciti ad estrapolare dalle interviste, non ci hanno ancora chiarito le idee su come strutturare il nostro lavoro. Per riuscire a comprendere meglio le dinamiche dell'azienda e per capire come i vari dati entrino in gioco nei vari processi aziendali, anche con l'aiuto di Tommaso, abbiamo deciso di realizzare lo schema che mostriamo nella Figura 1.1 che descrive, in linea generale, ma chiara, tutte le principali operazioni dell'impresa e come queste interagiscono tra di loro.

1.4 Descrizione in linguaggio naturale

Studiando attentamente sia le interviste effettuate sia lo schema dei processi aziendali, siamo riusciti a fare un elenco di tutte le funzionalità che il nostro software deve soddisfare.

L'obiettivo è quello di sviluppare un sistema per un'impresa artigiana edile che sia in grado di gestire tutti i settori evidenziati nelle fasi precedenti, che sono, nello specifico:

- settore lavori e clientela;
- settore dipendenti;
- settore sicurezza sul lavoro;
- settore magazzino;
- settore fornitori;
- settore contabilità.

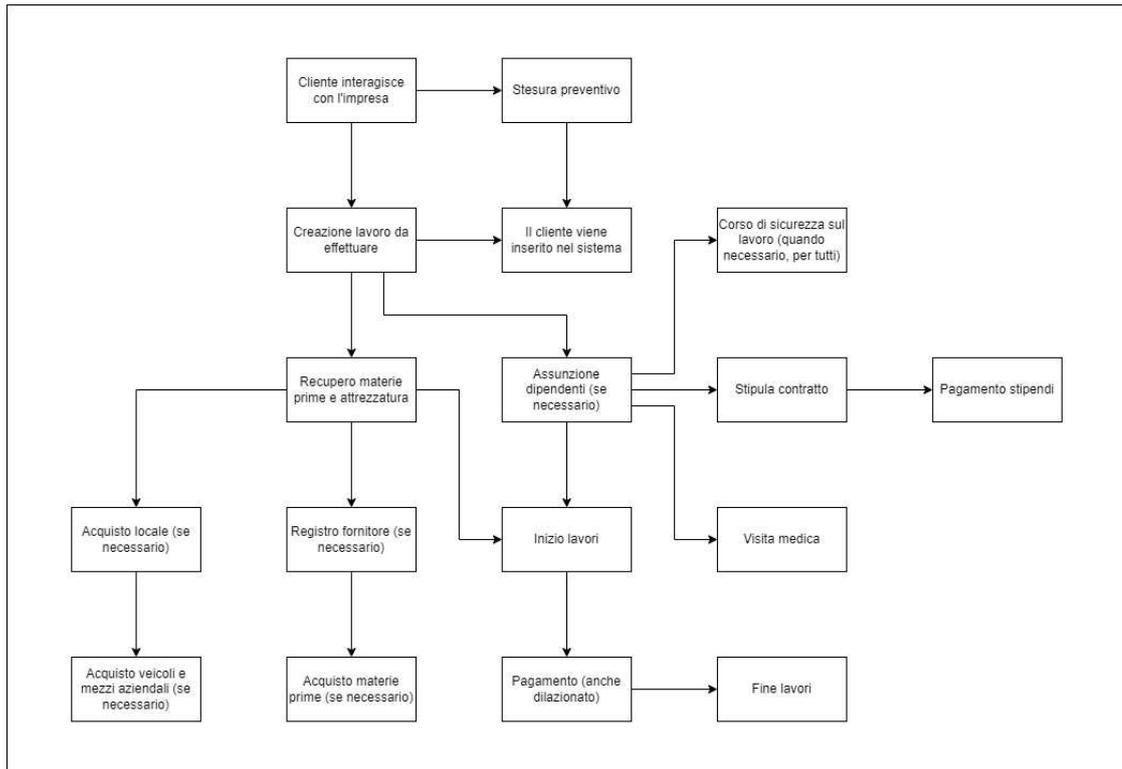


Figura 1.1: Diagramma dei processi interni

Riguardo il primo settore, il software deve permettere al personale di gestire le operazioni di inserimento, visualizzazione, modifica, eliminazione e ricerca per cliente di lavori intestati all'impresa e di preventivi stipulati. In merito alla clientela, così come suggeritoci dal diagramma dei processi aziendali, l'inserimento di un nuovo cliente potrà essere fatto soltanto a partire dalla creazione di un nuovo lavoro oppure di un nuovo preventivo; l'applicativo dovrà consentire anche la modifica dei dati di un singolo cliente, mentre, per quanto riguarda l'eliminazione, dovrà impedirlo nel caso in cui il cliente selezionato risulti impegnato in un lavoro oppure in un preventivo. L'utente, a partire dalla visualizzazione dei dettagli di un singolo preventivo, dovrà anche poter scaricare un PDF del preventivo stesso. Come ultima funzionalità il sistema dovrà consentire al personale di contattare la clientela attraverso l'e-mail, sia a partire dalla lista di tutti i clienti, che a partire dalla lista dei lavori/preventivi (in particolare, dovrà consentire di contattare il cliente del lavoro/preventivo selezionato).

Per quanto riguarda il settore relativo ai dipendenti, l'utente deve poter gestire sia la lista di tutto il personale che lavora per l'impresa sia i rispettivi contratti stipulati. L'inserimento di un nuovo dipendente deve essere svolto di pari passo a quello di un nuovo contratto; allo stesso modo l'eliminazione di un dipendente deve, a cascata, portare anche alla cancellazione del rispettivo contratto; deve essere consentito anche visualizzare, modificare e scaricare tramite PDF tali dati. Anche in questo caso il software deve permettere all'utente di contattare un qualsiasi dipendente a partire dall'e-mail. Ultimo requisito, il personale deve avere la possibilità di selezionare un dipendente e controllarne tutte le visite mediche effettuate; tali visite devono essere cancellate a seguito dell'eliminazione del dipendente che le ha svolte.

Proprio riguardo la sicurezza sul lavoro, il sistema deve consentire l'inserimento di una nuova visita medica, oltre che permetterne la visualizzazione, la modifica, l'eliminazione e la ricerca per paziente. La gestione dei corsi di sicurezza sul lavoro riguarda operazioni di inserimento, visualizzazione, modifica ed eliminazione.

La gestione del magazzino riguarda tre sottosectori: materie prime, locali e veicoli. Per

quanto riguarda le materie prime l'utente deve poter visualizzare la lista delle scorte presenti in magazzino e modificarne la quantità attraverso il consumo e l'acquisto; quest'ultimo deve essere gestito attraverso una lista della spesa e un listino prezzi dal quale, anche attraverso una ricerca per nome prodotto, l'utente deve poter selezionare la materia prima di cui necessita con la rispettiva quantità che desidera acquistare. La sezione locali deve essere gestita attraverso operazioni di inserimento, visualizzazione, modifica, eliminazione e filtraggio per tipo; l'utente deve, anche, poter selezionare un locale qualsiasi e, se presenti, visualizzarne i veicoli al suo interno. Infine, il sistema deve consentire l'inserimento, la visualizzazione, la modifica, l'eliminazione e il filtraggio per tipo o per locale dei veicoli.

I fornitori devono essere gestiti con operazioni di inserimento, visualizzazione, modifica, eliminazione e filtraggio per nome o tipologia. Inoltre, al momento dell'acquisto delle materie prime, l'utente deve poter selezionare il fornitore da cui effettua la spesa.

Come ultimo settore analizziamo la contabilità; il sistema deve permettere l'inserimento di un'entrata a partire dai lavori effettuati e dai locali e dai veicoli posseduti, i quali possono essere venduti o affittati/noleggiati; le uscite devono essere inserite a partire dall'acquisto di materie prime, di veicoli e di locali e dal pagamento dello stipendio per ogni dipendente; il bilancio deve essere creato automaticamente dal sistema a partire dall'anno selezionato e dalle entrate e le uscite avute in tale periodo. Il personale deve poter anche inserire manualmente entrate e uscite extra, visualizzarne i dettagli, modificarli ed, eventualmente, eliminarle. Deve essere anche consentita la visualizzazione e l'eliminazione di un bilancio. Per quanto riguarda il filtraggio, il sistema deve consentire la ricerca di entrate/uscite per data (iniziale e finale) e per provenienza/destinazione.

Dalla fase di intervista effettuata su vari livelli dell'azienda, ci siamo resi conto della necessità di aggiungere al sistema anche una parte relativa alle statistiche; nello specifico, l'utente deve poter consultare grafici inerenti agli stipendi dei dipendenti, agli importi dei lavori effettuati e ai bilanci annuali, deve poter visualizzare una panoramica della situazione economica (quindi totale entrate, totale uscite e saldo complessivo) a partire da una data iniziale e una finale e deve poter conoscere qual è la percentuale sia dei lavori correttamente pagati dal cliente (sul totale dei lavori presenti nel sistema) sia dei preventivi accettati (sul totale dei preventivi presenti nel sistema).

1.5 Glossario dei termini

Per evitare qualsiasi tipo di ambiguità, è conveniente specificare il significato esatto di ogni termine, che potrebbe essere interpretato in maniera sbagliata o addirittura non compreso o conosciuto. A tal scopo mostriamo nella Tabella 1.1 il glossario dei termini.

*Dipendente non è un sinonimo di paziente; tuttavia, nel nostro contesto, possiamo affermare che tutti i pazienti siano effettivamente dei dipendenti.

1.6 Diagramma dei sistemi

Nelle prime fasi del processo di specifica, parlando con gli stakeholder, abbiamo definito i confini del sistema. Dopo averli individuati, abbiamo realizzato un modello contestuale che mostra, come possiamo vedere in Figura 1.2, il nostro software e gli altri sistemi con il quale avvengono le interazioni.

Termine	Descrizione	Sinonimi
Acquisto	Processo attraverso il quale un individuo o un'organizzazione acquisisce beni o servizi in cambio di un pagamento.	-
Affitto	Locazione di un locale a tempo determinato e dietro pagamento.	-
Bilancio	Documento contabile che riassume la situazione economica, patrimoniale e finanziaria di un'azienda in un determinato periodo di tempo, solitamente un anno. Contiene informazioni sulle entrate, le uscite, i beni, i debiti e i crediti dell'azienda.	Analisi di bilancio
Causale	Breve descrizione che spiega il motivo o lo scopo del pagamento.	-
Cliente	Persona fisica che commissiona un lavoro oppure stipula un preventivo con la ditta.	Committente
Commercialista	Professionista esperto in un determinato settore che offre consulenze, consigli e supporto tecnico o strategico a individui, aziende o organizzazioni.	Consulente
Contabilità	Sistema di registrazione, classificazione e analisi delle transazioni finanziarie di un'azienda.	-
Contratto	Accordo formale che regola il rapporto di lavoro tra dipendente e datore di lavoro; definisce i diritti e i doveri di entrambi.	Accordo
Corso	Nell'ambito della sicurezza sul lavoro, è una formazione specifica rivolta ai lavoratori, ai datori di lavoro e ai responsabili della sicurezza, con l'obiettivo di fornire competenze e conoscenze necessarie per prevenire e gestire i rischi legati alle attività lavorative.	-
Dipendente	Persona fisica che opera all'interno dell'azienda in questione e viene pagata dalla stessa.	Lavoratore, operaio, personale
E-mail	Metodo di comunicazione digitale che consente di inviare e ricevere messaggi attraverso Internet.	-
Entrata	Somma di denaro ricevuta da un'azienda o un individuo, che aumenta il patrimonio.	-
Fornitore	Individuo o un'azienda che fornisce beni, materiali, prodotti o servizi ad un'altra azienda o organizzazione.	Venditore
Impresa	Attività economica organizzata per la produzione di beni e servizi.	Ditta, azienda
Lavoro	Attività svolta da una persona che richiede impegno, tempo e competenze, e che è svolta per raggiungere un obiettivo specifico o per ottenere una ricompensa, tipicamente sotto forma di salario o stipendio.	Attività, servizio
Lista spesa	Elenco scritto di articoli o prodotti che una persona intende acquistare.	-
Listino prezzi	Documento o tabella che elenca i prezzi dei beni o dei servizi offerti da un'azienda o un fornitore.	Tariffario
Locale	Unità di spazio affittabile o acquistabile per attività commerciali oppure adibita al soggiorno e alla vita quotidiana.	-
Magazzino	Area o edificio utilizzato per immagazzinare beni, materiali o prodotti.	Deposito
Materia prima	Insieme dei mezzi necessari per svolgere un'attività, eseguire un lavoro.	Materiale, prodotto
Noleggio	Come affitto, ma riferito a mezzi e veicoli.	-
PDF	Formato di file che consente di visualizzare, stampare e condividere documenti in modo uniforme su diversi dispositivi e piattaforme.	-
Paziente	Persona che riceve assistenza sanitaria o trattamenti medici da parte di professionisti della salute.	Dipendente*
Preventivo	Documento che fornisce una stima dettagliata dei costi associati a un progetto, un servizio o un bene.	-
Processo aziendale	Insieme di attività inter-correlate, svolte all'interno dell'azienda, che creano valore aggiunto destinato ad un soggetto interno o esterno all'azienda.	Processo interno
Saldo complessivo	Differenza tra le entrate e le uscite complessive.	-
Software gestionale	Tipo di software progettato per aiutare le aziende e le organizzazioni a gestire e automatizzare le loro operazioni quotidiane e i processi aziendali.	Sistema, applicativo
Uscita	Somma di denaro spesa da un'azienda o un individuo, che riduce il patrimonio.	-
Vendita	Processo attraverso il quale un bene o servizio viene trasferito da un venditore a un acquirente in cambio di un pagamento.	-
Veicolo	Mezzo di trasporto progettato per muoversi su strada, acqua, aria o terra e per trasportare persone, merci o entrambi da un luogo all'altro.	Mezzo
Visita medica	Incontro tra un paziente e un medico durante il quale viene valutata la salute del paziente.	-

Tabella 1.1: Glossario dei termini

1.7 Analisi dei requisiti

Per prima cosa, abbiamo ricavato i requisiti del software dall'intervista con gli stakeholder. Questi sono stati suddivisi in funzionali e non funzionali. I primi individuano le funzionalità o i servizi che il sistema deve fornire. Gli altri, invece, rappresentano i vincoli sulle funzioni o sui servizi offerti dal sistema.

Riportiamo una prima suddivisione in requisiti funzionali e non funzionali nella Figura 1.3.

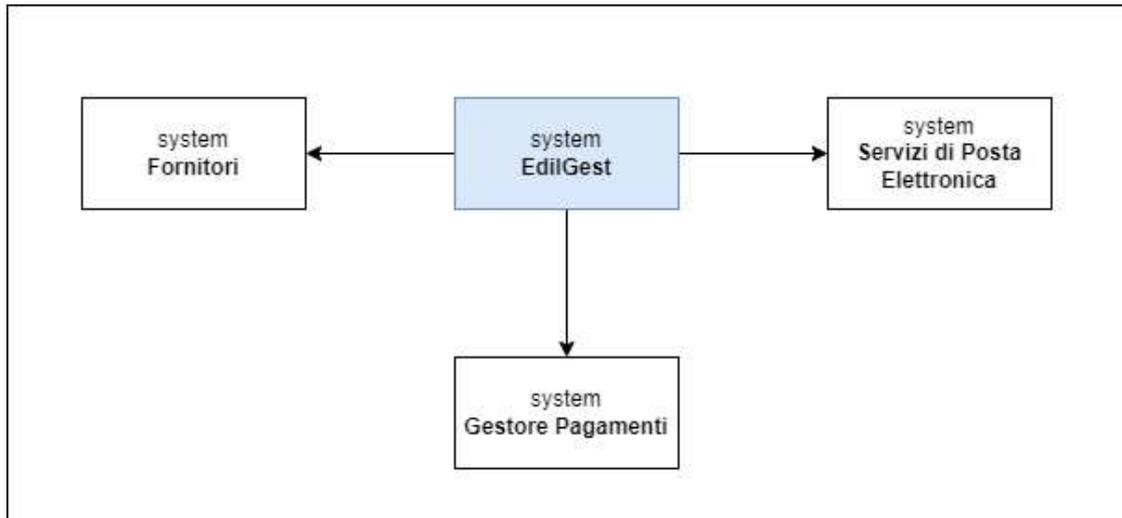


Figura 1.2: Diagramma dei sistemi



Figura 1.3: Suddivisione dei requisiti in funzionali e non funzionali

1.7.1 Requisiti funzionali

Successivamente siamo passati ad un'elencazione completa dei requisiti funzionali, suddivisi, come mostrato in Figura 1.4, nelle sette aree in cui è articolato il sistema: lavori e clientela, dipendenti, sicurezza sul lavoro, magazzino, fornitori, contabilità e statistiche.

In seguito vediamo elencati tutti i requisiti appena visti affiancati ognuno da una breve descrizione che ci aiuta a chiarire nello specifico cosa gli stakeholder vogliono che il sistema faccia:

1. *inserimento lavoro*: il sistema deve gestire l'inserimento di un nuovo lavoro;
2. *modifica lavoro*: il sistema deve gestire la modifica dei dati di un lavoro esistente;
3. *visualizzazione lista lavori*: il sistema deve gestire la visualizzazione di una lista contenente tutti i lavori esistenti;



Figura 1.4: Elenco dei requisiti funzionali

4. *visualizzazione dettagli lavoro*: il sistema deve gestire la visualizzazione dei dettagli di ogni singolo lavoro;
5. *eliminazione lavoro*: il sistema deve gestire l'eliminazione di un lavoro esistente;
6. *ricerca lavori per cliente*: il sistema deve permettere all'utente di selezionare un cliente e visualizzare la lista di tutti i lavori a lui associati;
7. *contatta cliente lavoro*: il sistema deve permettere all'utente di contattare il cliente di un qualsiasi lavoro selezionato;
8. *inserimento preventivo*: il sistema deve gestire l'inserimento di un nuovo preventivo;
9. *modifica preventivo*: il sistema deve gestire la modifica dei dati di un preventivo esistente;
10. *visualizzazione preventivi*: il sistema deve gestire la visualizzazione di una lista contenente tutti i preventivi stipulati;
11. *visualizzazione dettagli preventivo*: il sistema deve gestire la visualizzazione dei dettagli di ogni singolo preventivo;
12. *scarica PDF preventivo*: il sistema deve permettere la creazione di un PDF per ogni preventivo;
13. *eliminazione preventivo*: il sistema deve gestire l'eliminazione di un preventivo esistente;
14. *ricerca preventivi per cliente*: il sistema deve permettere all'utente di selezionare un cliente e visualizzare la lista di tutti i preventivi a lui associati;
15. *contatta cliente preventivo*: il sistema deve permettere all'utente di contattare il cliente di un qualsiasi preventivo selezionato;
16. *inserimento cliente*: il sistema deve gestire l'inserimento di un nuovo cliente;
17. *modifica cliente*: il sistema deve gestire la modifica dei dati di un cliente esistente;
18. *visualizzazione lista clienti*: il sistema deve gestire la visualizzazione di una lista contenente tutti i clienti;
19. *visualizzazione dettaglio cliente*: il sistema deve gestire la visualizzazione dei dettagli di ogni singolo cliente;
20. *eliminazione cliente*: il sistema deve gestire l'eliminazione di un cliente esistente;
21. *ricerca clienti per nome*: il sistema deve permettere all'utente di visualizzare tutti i clienti che hanno il nome dato in input;
22. *ricerca clienti per cognome*: il sistema deve permettere all'utente di visualizzare tutti i clienti che hanno il cognome dato in input;
23. *contatta cliente*: il sistema deve permettere all'utente di contattare un qualsiasi cliente;
24. *inserimento dipendente*: il sistema deve gestire l'inserimento di un nuovo dipendente;
25. *modifica dipendente*: il sistema deve gestire la modifica dei dati di un dipendente esistente;
26. *visualizzazione lista dipendenti*: il sistema deve gestire la visualizzazione di una lista di tutti i dipendenti;

27. *visualizzazione dettaglio dipendente*: il sistema deve gestire la visualizzazione dei dettagli di ogni singolo dipendente;
28. *visualizzazione lista visite mediche associate ad un dipendente*: il sistema deve permettere la visualizzazione di una lista di tutte le visite mediche effettuate da ciascun dipendente;
29. *eliminazione dipendente*: il sistema deve gestire l'eliminazione di un dipendente esistente;
30. *inserimento contratto dipendente*: il sistema deve gestire l'inserimento di un nuovo contratto per un dipendente;
31. *modifica contratto dipendente*: il sistema deve gestire la modifica dei dati di un contratto esistente;
32. *eliminazione contratto dipendente*: il sistema deve gestire l'eliminazione di un contratto esistente;
33. *visualizzazione dettaglio contratto dipendente*: il sistema deve gestire la visualizzazione dei dettagli di ogni singolo contratto dipendente;
34. *scarica PDF contratto dipendente*: il sistema deve permettere la creazione di un PDF per ogni contratto dipendente;
35. *contatta dipendente*: il sistema deve permettere all'utente di contattare un qualsiasi dipendente;
36. *inserimento corso di sicurezza*: il sistema deve gestire l'inserimento di un nuovo corso di sicurezza;
37. *modifica corso di sicurezza*: il sistema deve gestire la modifica dei dati di un corso di sicurezza esistente;
38. *visualizzazione lista corsi di sicurezza*: il sistema deve gestire la visualizzazione di una lista contenente tutti i corsi di sicurezza effettuati;
39. *visualizzazione dettagli corso di sicurezza*: il sistema deve gestire la visualizzazione dei dettagli di ogni singolo corso di sicurezza;
40. *eliminazione corso di sicurezza*: il sistema deve gestire l'eliminazione di un corso di sicurezza esistente;
41. *inserimento visita medica*: il sistema deve gestire l'inserimento di una nuova visita medica;
42. *modifica visita medica*: il sistema deve gestire la modifica dei dati di una visita medica esistente;
43. *visualizzazione lista visite mediche*: il sistema deve gestire la visualizzazione di una lista contenente tutte le visite mediche effettuate;
44. *visualizzazione dettagli visita medica*: il sistema deve gestire la visualizzazione dei dettagli di ogni singola visita medica;
45. *eliminazione visita medica*: il sistema deve gestire l'eliminazione di una visita medica esistente;
46. *ricerca visita medica per dipendente*: il sistema deve permettere all'utente di selezionare un dipendente e visualizzare la lista di tutte le visite mediche a lui associate;

47. *contatta paziente visita medica*: il sistema deve permettere all'utente di contattare il paziente di una qualsiasi visita medica selezionata;
48. *acquisto materie prime*: il sistema deve gestire l'acquisto di materie prime;
49. *inserimento materie prime in magazzino*: il sistema deve gestire l'inserimento di materie prime in magazzino;
50. *ricerca materia prima in listino prezzi*: il sistema deve gestire la ricerca per nome dei prodotti presenti nel listino prezzi;
51. *consumo materie prime*: il sistema deve gestire l'eliminazione (quindi il consumo) di materie prime dal magazzino;
52. *visualizzazione scorte in magazzino*: il sistema deve gestire la visualizzazione delle scorte presenti in magazzino;
53. *inserimento veicolo*: il sistema deve gestire l'inserimento di un nuovo veicolo;
54. *modifica veicolo*: il sistema deve gestire la modifica dei dati di un veicolo esistente;
55. *visualizzazione lista veicoli*: il sistema deve gestire la visualizzazione di una lista contenente tutti i veicoli posseduti;
56. *visualizzazione dettagli veicolo*: il sistema deve gestire la visualizzazione dei dettagli di ogni singolo veicolo;
57. *eliminazione veicolo*: il sistema deve gestire l'eliminazione di un veicolo esistente;
58. *ricerca veicoli per tipologia*: il sistema deve permettere all'utente di visualizzare tutti i veicoli della tipologia data in input;
59. *ricerca veicoli per locale*: il sistema deve permettere all'utente di visualizzare una lista di tutti i veicoli associati al locale dato in input;
60. *inserimento locale*: il sistema deve gestire l'inserimento di un nuovo locale;
61. *modifica locale*: il sistema deve gestire la modifica dei dati di un locale esistente;
62. *visualizzazione lista locali*: il sistema deve gestire la visualizzazione di una lista contenente tutti i locali posseduti;
63. *visualizzazione dettagli locale*: il sistema deve gestire la visualizzazione dei dettagli di ogni singolo locale;
64. *visualizzazione lista veicoli associati ad un locale*: il sistema dovrà permettere la visualizzazione di una lista di tutti i veicoli contenuti in ogni locale;
65. *eliminazione locale*: il sistema deve gestire l'eliminazione di un locale esistente;
66. *ricerca locali per tipo*: il sistema deve permettere all'utente di visualizzare tutti i locali del tipo dato in input;
67. *inserimento fornitore*: il sistema deve gestire l'inserimento di un nuovo fornitore;
68. *modifica fornitore*: il sistema deve gestire la modifica dei dati un fornitore esistente;
69. *visualizzazione lista fornitori*: il sistema deve gestire la visualizzazione di una lista contenente tutti i fornitori;

70. *visualizzazione dettagli fornitore*: il sistema deve gestire la visualizzazione dei dettagli di ogni singolo fornitore;
71. *eliminazione fornitore*: il sistema deve gestire l'eliminazione di un fornitore esistente;
72. *contatta fornitore*: il sistema deve permettere all'utente di contattare un qualsiasi fornitore;
73. *ricerca fornitori per tipologia*: il sistema deve permettere all'utente di visualizzare tutti i fornitori della tipologia data in input;
74. *ricerca fornitore per nome*: il sistema deve permettere all'utente di ricercare un fornitore a partire dal nome dato in input;
75. *inserimento bilancio*: il sistema deve gestire l'inserimento di una nuova analisi di bilancio;
76. *visualizzazione lista bilanci*: il sistema deve gestire la visualizzazione di una lista contenente tutti i bilanci passati dell'azienda;
77. *visualizzazione dettagli bilancio*: il sistema dovrà gestire la visualizzazione dei dettagli di ogni singolo bilancio;
78. *eliminazione bilancio*: il sistema deve gestire l'eliminazione di un bilancio esistente;
79. *inserimento uscita*: il sistema deve gestire l'inserimento di una qualsiasi nuova uscita;
80. *modifica uscita*: il sistema deve gestire la modifica dei dati di un'uscita esistente;
81. *visualizzazione lista uscite*: il sistema deve gestire la visualizzazione di una lista contenente tutte le uscite economiche dell'impresa;
82. *visualizzazione dettagli uscita*: il sistema deve gestire la visualizzazione dei dettagli di ogni singola uscita;
83. *eliminazione uscita*: il sistema deve gestire l'eliminazione di un'uscita presente in memoria;
84. *inserimento uscita specifica*: il sistema deve permettere all'utente di inserire una nuova uscita a partire dall'acquisto di materie prime, locali e veicoli e dal pagamento degli stipendi ai dipendenti;
85. *ricerca uscite per destinazione*: il sistema deve permettere all'utente di filtrare le uscite per destinazione;
86. *ricerca uscite per data*: il sistema deve permettere all'utente di filtrare le uscite per data (iniziale e finale);
87. *inserimento entrata*: il sistema deve gestire l'inserimento di una qualsiasi nuova entrata;
88. *modifica entrata*: il sistema deve gestire la modifica dei dati di un'entrata esistente;
89. *visualizzazione lista entrate*: il sistema deve gestire la visualizzazione di una lista contenente tutte le entrate economiche dell'impresa;
90. *visualizzazione dettagli entrata*: il sistema deve gestire la visualizzazione dei dettagli di ogni singola entrata;

91. *eliminazione entrata*: il sistema deve gestire l'eliminazione di un'entrata presente in memoria;
92. *inserimento entrata specifica*: il sistema deve permettere all'utente di inserire una nuova entrata a partire da un lavoro pagato, da un locale venduto o affittato e da un veicolo venduto o noleggiato;
93. *ricerca entrate per provenienza*: il sistema deve permettere all'utente di filtrare le entrate per provenienza;
94. *ricerca entrate per data*: il sistema deve permettere all'utente di filtrare le entrate per data (iniziale e finale);
95. *visualizzazione statistiche sui dipendenti*: il sistema deve gestire la visualizzazione di un grafico che mostra gli stipendi dei dipendenti;
96. *visualizzazione statistiche su costi/ricavi*: il sistema deve gestire la visualizzazione di una pagina in cui sono mostrati entrate, uscite e saldo totale a partire da un arco temporale selezionato;
97. *visualizzazione statistiche sui bilanci*: il sistema deve gestire la visualizzazione di un grafico che mostra l'andamento anno per anno del saldo totale di tutti i bilanci;
98. *visualizzazione statistiche sui lavori*: il sistema deve gestire la visualizzazione di un grafico che mostra gli importi dei lavori commissionati all'impresa e del dato percentuale dei lavori pagati interamente dal cliente su quelli totali presenti nel sistema;
99. *visualizzazione statistiche sui preventivi*: il sistema deve gestire la visualizzazione del dato percentuale dei preventivi accettati su quelli totali stipulati.

1.7.2 Requisiti non funzionali

I requisiti non funzionali sono esattamente quelli elencati nella Figura 1.3 nella rispettiva sezione. Di seguito, analogamente a quanto fatto con i requisiti funzionali, proponiamo un elenco di tutti i requisiti non funzionali affiancati ad una breve descrizione per ciascuno di essi.

1. *Linguaggio*: il sistema deve essere implementato in Python;
2. *login*: il sistema deve garantire la sicurezza attraverso un login (nome utente e password);
3. *interfaccia grafica*: il sistema deve essere dotato di interfaccia grafica;
4. *usabilità*: il sistema deve offrire un'ottima UI.

1.8 Diagrammi dei casi d'uso

Dopo una prima definizione e descrizione dei requisiti, siamo passati ad elencare i casi d'uso, ovvero gli scenari che si potranno verificare durante l'utilizzo del sistema.

Anche in questo caso abbiamo ritenuto opportuno analizzare separatamente le sette sezioni in cui è articolato il sistema. Ogni settore verrà presentato attraverso un diagramma UML dei casi d'uso accompagnato da una descrizione per ogni scenario che entra in gioco.

1.8.1 Gestione dei lavori e dei clienti

Iniziamo con il primo settore, ovvero quello relativo ai lavori e alla clientela, di cui presentiamo il diagramma dei casi d'uso nella Figura 1.5

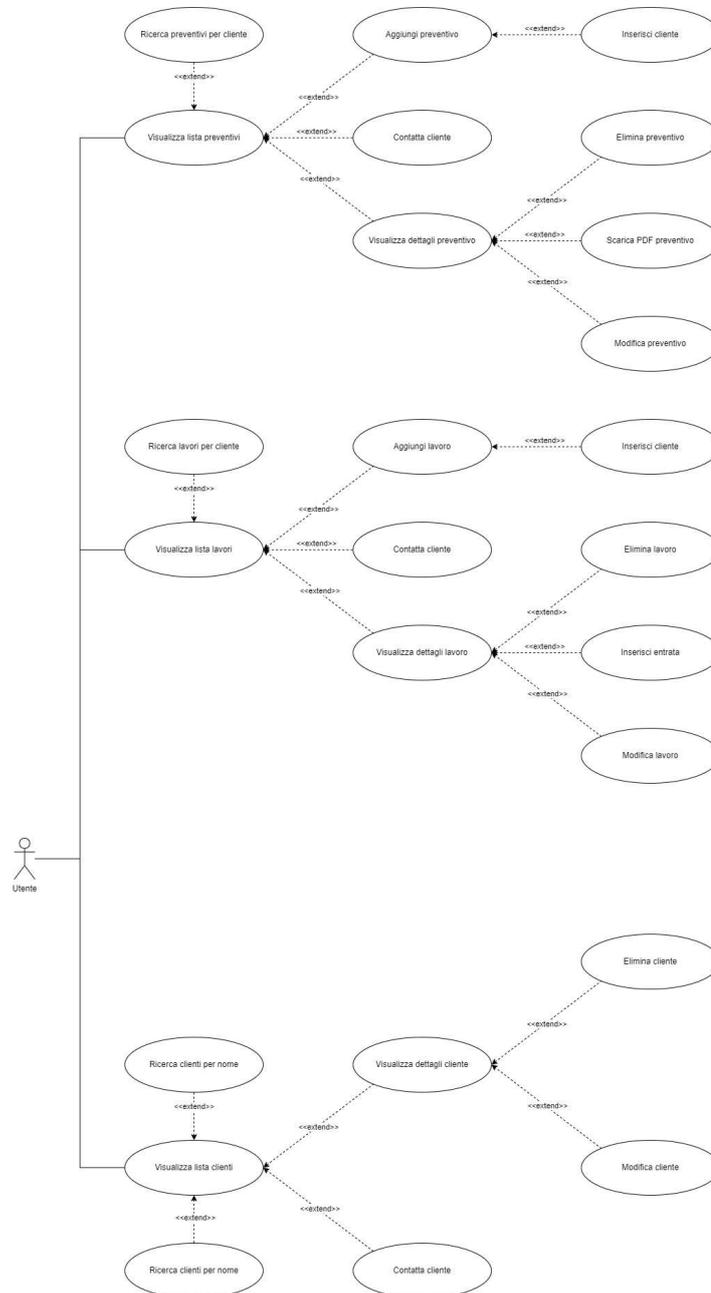


Figura 1.5: Diagramma dei casi d'uso relativo alla gestione dei lavori e dei clienti

Visualizza lista preventivi

Il caso d'uso "Visualizza lista preventivi" si verifica quando l'utente vuole visualizzare sullo schermo una lista contenente tutti i preventivi presenti nel sistema. L'utente, a partire dalla home page, clicca sul pulsante "Lavori e Preventivi" e seleziona l'opzione "Preventivi"; successivamente, il sistema visualizza la lista dei preventivi stipulati dalla ditta.

Ricerca preventivi per cliente

Il caso d'uso "Ricerca preventivi per cliente" si verifica quando l'utente vuole effettuare un filtraggio tra tutti i preventivi per cliente. Dopo aver visualizzato la lista completa, l'utente clicca sul pulsante "Filtri", seleziona l'opzione "Filtra per cliente", seleziona il cliente tra quelli presenti a sistema e visualizza a schermo una lista di tutti i preventivi che hanno come cliente quello selezionato.

Aggiungi preventivo

Il caso d'uso "Aggiungi preventivo" si verifica quando l'utente vuole inserire a sistema un nuovo preventivo. Dopo essere entrato nella sezione "Preventivi" e aver visualizzato la lista di tutti i preventivi, l'utente clicca sul pulsante "Aggiungi", successivamente compila i campi della schermata di inserimento e, se corretti, dopo aver cliccato sul pulsante "Conferma", completa correttamente l'operazione. In caso di dati incompleti o non corretti, il sistema ripropone l'inserimento dopo aver segnalato l'errore commesso.

Contatta cliente

Il caso d'uso "Contatta cliente" si verifica quando l'utente vuole contattare il cliente di un certo preventivo. Dopo aver visualizzato la lista di tutti i preventivi, l'utente ne seleziona uno e clicca sul pulsante "Contatta cliente"; successivamente il sistema recupera l'e-mail del cliente del preventivo selezionato e apre una nuova mail con il campo "Destinatario" precompilato con l'e-mail recuperata.

Visualizza dettagli preventivo

Il caso d'uso "Visualizza dettagli preventivo" si verifica quando l'utente vuole visualizzare a schermo tutti i dati di un preventivo specifico. Dopo aver visualizzato la lista di tutti i preventivi, l'utente ne seleziona uno e clicca sul pulsante "Modifica"; successivamente visualizza una nuova schermata contenente tutte le informazioni riguardo il preventivo selezionato.

Elimina preventivo

Il caso d'uso "Elimina preventivo" si verifica quando l'utente vuole eliminare un preventivo presente nel sistema. Dopo aver visualizzato il dettaglio del preventivo da eliminare, l'utente clicca sul pulsante "Elimina" e, dopo aver confermato, termina correttamente l'operazione. Il preventivo eliminato non è più presente nel sistema.

Scarica PDF preventivo

Il caso d'uso "Scarica PDF preventivo" si verifica quando l'utente vuole scaricare sul dispositivo un file PDF contenente tutti i dati di un certo preventivo. Dopo aver visualizzato il dettaglio del preventivo da scaricare, l'utente clicca sul pulsante "Scarica PDF"; il sistema genera un PDF contenente tutti i dati del preventivo selezionato.

Modifica preventivo

Il caso d'uso "Modifica preventivo" si verifica quando l'utente vuole modificare i dati di un preventivo presente nel sistema. Dopo aver visualizzato il dettaglio del preventivo da modificare, l'utente ne modifica i dati e, se corretti, dopo aver cliccato sul pulsante

"Conferma", termina correttamente l'operazione. In caso di dati incompleti o non corretti, il sistema ripropone la modifica dopo aver segnalato l'errore commesso.

Visualizza lista lavori

Il caso d'uso "Visualizza lista lavori" si verifica quando l'utente vuole visualizzare a schermo una lista contenente tutti i lavori presenti nel sistema. L'utente, a partire dalla home page, clicca sul pulsante "Lavori e Preventivi" e seleziona l'opzione "Lavori"; successivamente visualizza correttamente la lista dei lavori commissionati alla ditta.

Ricerca lavori per cliente

Il caso d'uso "Ricerca lavori per cliente" si verifica quando l'utente vuole effettuare un filtraggio tra tutti i lavori per cliente. Dopo aver visualizzato la lista completa, l'utente clicca sul pulsante "Filtri", seleziona l'opzione "Filtra per cliente", seleziona il cliente tra quelli presenti a sistema e visualizza a schermo una lista di tutti i lavori che hanno come cliente quello selezionato.

Aggiungi lavoro

Il caso d'uso "Aggiungi lavoro" si verifica quando l'utente vuole inserire a sistema un nuovo lavoro. Dopo essere entrato nella sezione "Lavori" e aver visualizzato la lista di tutti i lavori, l'utente clicca sul pulsante "Aggiungi", successivamente compila i campi della schermata di inserimento e, se corretti, dopo aver cliccato sul pulsante "Conferma", completa correttamente l'operazione. In caso di dati incompleti o non corretti, il sistema ripropone l'inserimento dopo aver segnalato l'errore commesso.

Contatta cliente

Il caso d'uso "Contatta cliente" si verifica quando l'utente vuole contattare il cliente di un certo lavoro. Dopo aver visualizzato la lista di tutti i lavori, l'utente ne seleziona uno e clicca sul pulsante "Contatta cliente"; successivamente il sistema recupera l'e-mail del cliente del lavoro selezionato e apre una nuova mail con il campo "Destinatario" precompilato con l'e-mail recuperata.

Visualizza dettagli lavoro

Il caso d'uso "Visualizza dettagli lavoro" si verifica quando l'utente vuole visualizzare a schermo tutti i dati di un lavoro specifico. Dopo aver visualizzato la lista di tutti i lavori, l'utente ne seleziona uno e clicca sul pulsante "Modifica"; successivamente visualizza una nuova schermata contenente tutte le informazioni riguardo il lavoro selezionato.

Elimina lavoro

Il caso d'uso "Elimina lavoro" si verifica quando l'utente vuole eliminare un lavoro presente nel sistema. Dopo aver visualizzato il dettaglio del lavoro da eliminare, l'utente clicca sul pulsante "Elimina" e, dopo aver confermato, termina correttamente l'operazione. Il lavoro eliminato non è più presente nel sistema.

Modifica lavoro

Il caso d'uso "Modifica lavoro" si verifica quando l'utente vuole modificare i dati di un lavoro presente nel sistema. Dopo aver visualizzato il dettaglio del lavoro da modificare, l'utente ne modifica i dati e, se corretti, dopo aver cliccato sul pulsante "Conferma", termina correttamente l'operazione. In caso di dati incompleti o non corretti, il sistema ripropone la modifica dopo aver segnalato l'errore commesso.

Inserisci cliente

Il caso d'uso "Inserisci cliente" si verifica al momento dell'inserimento di un nuovo preventivo o di un nuovo lavoro. Durante la fase di compilazione dei dati relativi al lavoro/-preventivo il sistema propone all'utente di compilare i campi relativi alle informazioni del cliente; il sistema, sempre dopo aver controllato la correttezza e la completezza di tutti i dati, compresi quelli del cliente, effettua l'inserimento prima del lavoro/preventivo e poi controlla se il cliente è già presente: in caso di risposta affermativa non succede nulla, mentre in caso di risposta negativa avviene anche l'inserimento del nuovo cliente.

Visualizza lista clienti

Il caso d'uso "Visualizza lista clienti" si verifica quando l'utente vuole visualizzare a schermo una lista contenente tutti i clienti presenti nel sistema. L'utente, a partire dalla home page, clicca sul pulsante "Clienti"; successivamente visualizza correttamente la lista di tutti i clienti dell'azienda.

Ricerca clienti per nome

Il caso d'uso "Ricerca clienti per nome" si verifica quando l'utente vuole effettuare un filtraggio tra tutti i clienti per nome. Dopo aver visualizzato la lista completa, l'utente clicca sul pulsante "Filtri", seleziona l'opzione "Filtra per nome", inserisce il nome che desidera e visualizza a schermo una lista di tutti i clienti che hanno come nome quello dato in input.

Ricerca clienti per cognome

Il caso d'uso "Ricerca clienti per cognome" si verifica quando l'utente vuole effettuare un filtraggio tra tutti i clienti per cognome. Dopo aver visualizzato la lista completa, l'utente clicca sul pulsante "Filtri", seleziona l'opzione "Filtra per cognome", inserisce il cognome che desidera e visualizza a schermo una lista di tutti i clienti che hanno come cognome quello dato in input.

Visualizza dettagli cliente

Il caso d'uso "Visualizza dettagli cliente" si verifica quando l'utente vuole visualizzare a schermo tutti i dati di un cliente specifico. Dopo aver visualizzato la lista di tutti i clienti, l'utente ne seleziona uno e clicca sul pulsante "Modifica"; successivamente visualizza una nuova schermata contenente tutte le informazioni riguardo il cliente selezionato.

Elimina cliente

Il caso d'uso "Elimina cliente" si verifica quando l'utente vuole eliminare un cliente presente nel sistema. Dopo aver visualizzato il dettaglio del cliente da eliminare, l'utente clicca sul pulsante "Elimina"; se il cliente risulta impegnato in un preventivo oppure in un

lavoro l'eliminazione non va a buon fine, altrimenti, dopo aver dato conferma, l'operazione termina correttamente. Il cliente eliminato non è più presente nel sistema.

Modifica cliente

Il caso d'uso "Modifica cliente" si verifica quando l'utente vuole modificare i dati di un cliente presente nel sistema. Dopo aver visualizzato il dettaglio del cliente da modificare, l'utente ne modifica i dati e, se corretti, dopo aver cliccato sul pulsante "Conferma", termina correttamente l'operazione. In caso di dati incompleti o non corretti, il sistema ripropone la modifica dopo aver segnalato l'errore commesso.

Contatta cliente

Il caso d'uso "Contatta cliente" si verifica quando l'utente vuole contattare un certo cliente a partire dalla lista di tutti i clienti. Dopo aver visualizzato la lista di tutti i clienti, l'utente ne seleziona uno e clicca sul pulsante "Contatta"; successivamente il sistema recupera l'e-mail del cliente selezionato e apre una nuova mail con il campo "Destinatario" precompilato con l'e-mail recuperata.

1.8.2 Gestione contratti e dipendenti

Andiamo avanti con il secondo settore, quello, cioè, riferito alla gestione dei dipendenti e dei contratti, di cui mostriamo il rispettivo diagramma dei casi d'uso in Figura 1.6.

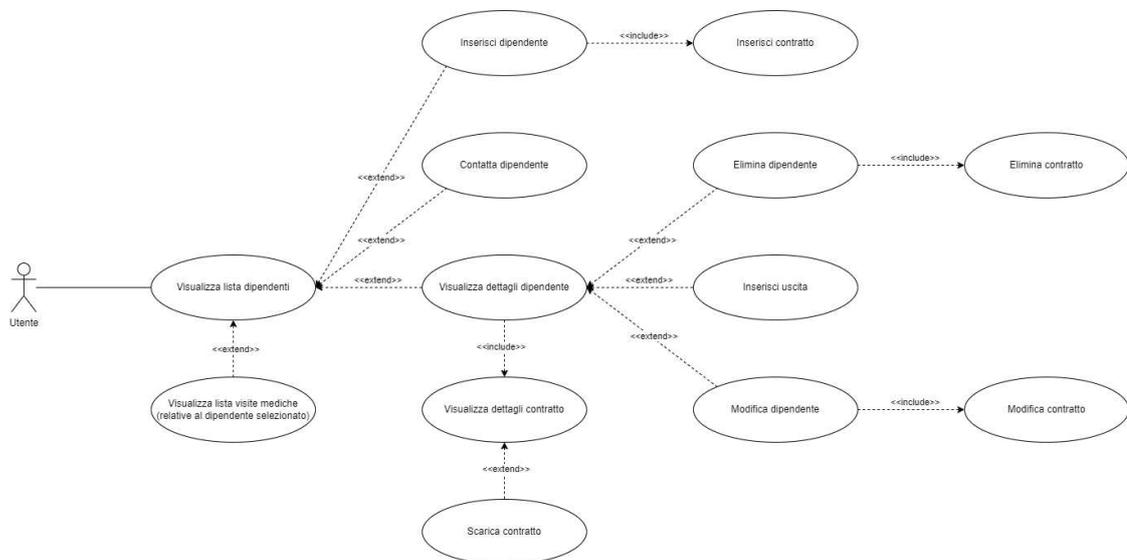


Figura 1.6: Diagramma dei casi d'uso relativo alla gestione dei contratti e dei dipendenti

Aggiungi dipendente

Il caso d'uso "Aggiungi dipendente" si verifica quando l'utente vuole inserire nel sistema un nuovo dipendente. Dopo essere entrato nella sezione "Dipendenti" e aver visualizzato la lista di tutti i dipendenti, l'utente clicca sul pulsante "Aggiungi"; successivamente compila i campi della schermata di inserimento e, se questi sono corretti, dopo aver cliccato sul pulsante "Conferma", completa correttamente l'operazione. In caso di dati incompleti o non corretti, il sistema ripropone l'inserimento dopo aver segnalato l'errore commesso.

Aggiungi contratto

Il caso d'uso "Aggiungi contratto" si verifica al momento dell'inserimento di un nuovo dipendente. Durante la fase di compilazione dei dati relativi al dipendente, il sistema propone all'utente di compilare i campi relativi alle informazioni del rispettivo contratto; il sistema, sempre dopo aver controllato la correttezza e la completezza di tutti i dati, compresi quelli del contratto, effettua l'inserimento prima del dipendente e poi del rispettivo contratto.

Contatta dipendente

Il caso d'uso "Contatta dipendente" si verifica quando l'utente vuole contattare un certo dipendente. Dopo aver visualizzato la lista di tutti i dipendenti, l'utente ne seleziona uno e clicca sul pulsante "Contatta dipendente"; successivamente il sistema recupera l'e-mail del dipendente selezionato e apre una nuova mail con il campo "Destinatario" precompilato con l'e-mail recuperata.

Visualizza dettagli dipendente

Il caso d'uso "Visualizza dettagli dipendente" si verifica quando l'utente vuole visualizzare sullo schermo tutti i dati di un dipendente specifico. Dopo aver visualizzato la lista di tutti i dipendenti, l'utente ne seleziona uno e clicca sul pulsante "Modifica"; successivamente visualizza una nuova schermata contenente tutte le informazioni riguardanti sia il dipendente selezionato sia il rispettivo contratto.

Visualizza dettagli contratto

Il caso d'uso "Visualizza dettagli contratto" si verifica parallelamente al caso d'uso "Visualizza dettagli dipendente". Dopo aver visualizzato la lista di tutti i dipendenti, l'utente ne seleziona uno e clicca sul pulsante "Modifica"; successivamente visualizza nella nuova schermata, oltre alle informazioni relative al dipendente selezionato, anche i dati del rispettivo contratto.

Scarica PDF contratto

Il caso d'uso "Scarica PDF contratto" si verifica quando l'utente vuole scaricare sul dispositivo un file PDF contenente tutti i dati di un certo contratto e del relativo dipendente. Dopo aver visualizzato il dettaglio del contratto da scaricare, l'utente clicca sul pulsante "Scarica PDF"; il sistema genera un PDF contenente tutti i dati sia del dipendente sia del rispettivo contratto desiderato.

Elimina dipendente

Il caso d'uso "Elimina dipendente" si verifica quando l'utente vuole eliminare un dipendente presente nel sistema. Dopo aver visualizzato il dettaglio del dipendente da eliminare, l'utente clicca sul pulsante "Elimina" e, dopo aver confermato, termina correttamente l'operazione. Il dipendente eliminato non è più presente nel sistema.

Elimina contratto

Il caso d'uso "Elimina contratto" si verifica parallelamente al caso d'uso "Elimina dipendente". Dopo aver visualizzato il dettaglio del dipendente da eliminare, l'utente clicca sul

pulsante "Elimina" e, dopo aver confermato, termina correttamente l'operazione. Oltre al dipendente, anche il rispettivo contratto viene eliminato e non esiste più nel sistema.

Modifica dipendente

Il caso d'uso "Modifica dipendente" si verifica quando l'utente vuole modificare i dati di un dipendente presente nel sistema. Dopo aver visualizzato il dettaglio del dipendente da modificare, l'utente ne modifica i dati e, se questi sono corretti, dopo aver cliccato sul pulsante "Conferma", termina correttamente l'operazione. In caso di dati incompleti o non corretti, il sistema ripropone la modifica dopo aver segnalato l'errore commesso.

Modifica contratto

Il caso d'uso "Modifica contratto" si verifica parallelamente al caso d'uso "Modifica dipendente". Dopo aver visualizzato il dettaglio del dipendente da modificare e del rispettivo contratto, l'utente ne modifica i dati (sia del dipendente sia del contratto) e, se questi sono corretti, dopo aver cliccato sul pulsante "Conferma", termina correttamente l'operazione. In caso di dati incompleti o non corretti, il sistema ripropone la modifica dopo aver segnalato l'errore commesso.

1.8.3 Gestione della sicurezza

Proseguiamo passando al terzo settore, cioè quello che si occupa della gestione della sicurezza sul lavoro, di cui facciamo vedere in Figura 1.7 il diagramma dei casi d'uso.

Visualizza lista corsi di sicurezza

Il caso d'uso "Visualizza lista corsi di sicurezza" si verifica quando l'utente vuole visualizzare sullo schermo una lista contenente tutti i corsi di sicurezza presenti nel sistema. L'utente, a partire dalla home page, clicca sul pulsante "Sicurezza" e seleziona l'opzione "Corsi di sicurezza"; successivamente il sistema visualizza la lista dei corsi di sicurezza seguiti dal personale dell'impresa.

Aggiungi corso di sicurezza

Il caso d'uso "Aggiungi corso di sicurezza" si verifica quando l'utente vuole inserire nel sistema un nuovo corso di sicurezza. Dopo essere entrato nella sezione "Corsi di sicurezza" e aver visualizzato la lista di tutti i corsi di sicurezza, l'utente clicca sul pulsante "Aggiungi"; successivamente compila i campi della schermata di inserimento e, se questi sono corretti, dopo aver cliccato sul pulsante "Conferma", completa correttamente l'operazione. In caso di dati incompleti o non corretti, il sistema ripropone l'inserimento dopo aver segnalato l'errore commesso.

Visualizza dettagli corso di sicurezza

Il caso d'uso "Visualizza dettagli corso di sicurezza" si verifica quando l'utente vuole visualizzare sullo schermo tutti i dati di un corso di sicurezza specifico. Dopo aver visualizzato la lista di tutti i corsi di sicurezza, l'utente ne seleziona uno e clicca sul pulsante "Modifica"; successivamente visualizza una nuova schermata contenente tutte le informazioni riguardo il corso di sicurezza selezionato.

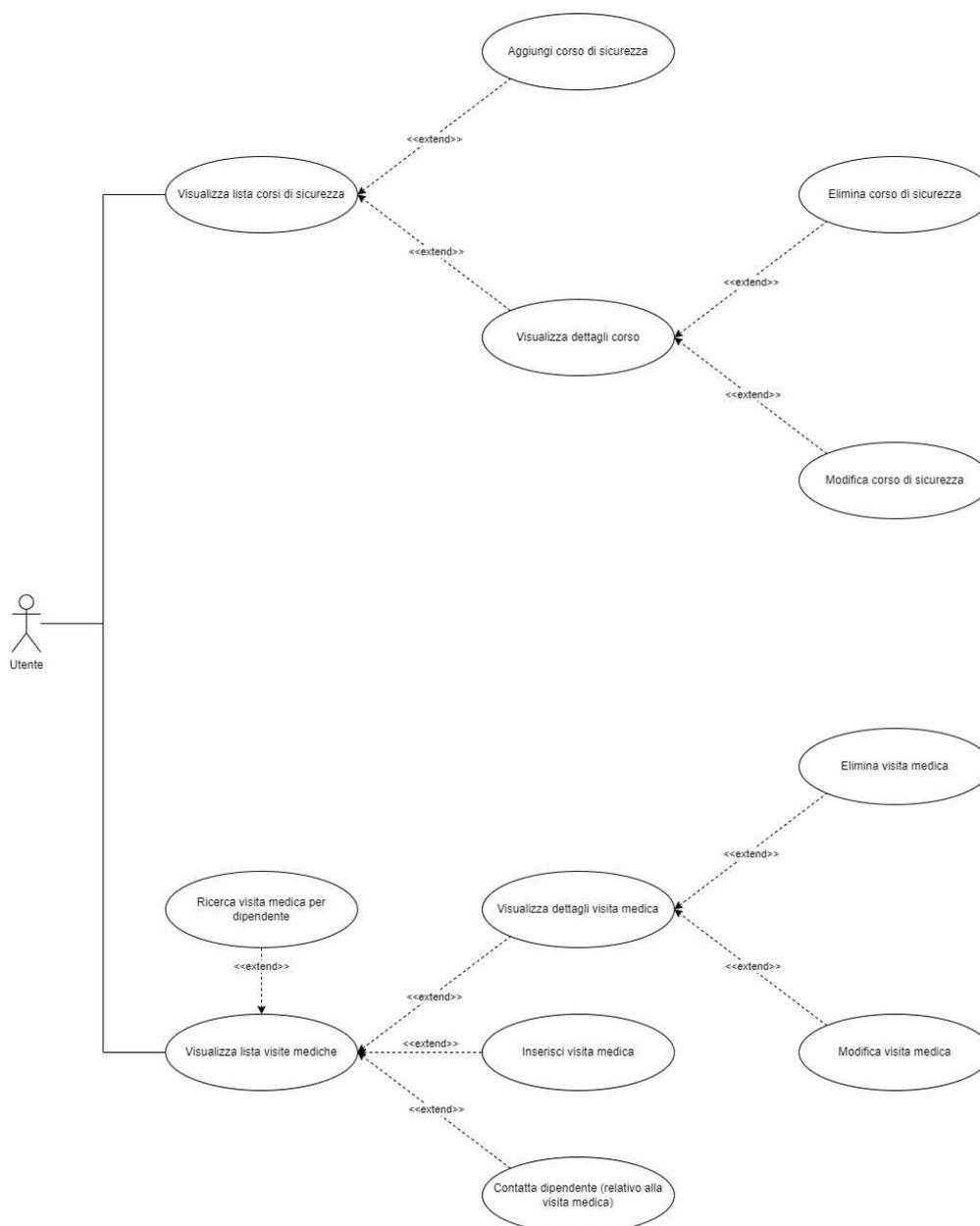


Figura 1.7: Diagramma dei casi d'uso relativo alla gestione della sicurezza

Elimina corso di sicurezza

Il caso d'uso "Elimina corso di sicurezza" si verifica quando l'utente vuole eliminare un corso di sicurezza presente nel sistema. Dopo aver visualizzato il dettaglio del corso di sicurezza da eliminare, l'utente clicca sul pulsante "Elimina" e, dopo aver confermato, termina correttamente l'operazione. Il corso di sicurezza eliminato non è più presente nel sistema.

Modifica corso di sicurezza

Il caso d'uso "Modifica corso di sicurezza" si verifica quando l'utente vuole modificare i dati di un corso di sicurezza presente nel sistema. Dopo aver visualizzato il dettaglio del corso di sicurezza da modificare, l'utente ne modifica i dati e, se questi sono corretti, dopo aver cliccato sul pulsante "Conferma", termina correttamente l'operazione. In caso di

dati incompleti o non corretti, il sistema ripropone la modifica dopo aver segnalato l'errore commesso.

Visualizza lista visite mediche

Il caso d'uso "Visualizza lista visite mediche" si verifica quando l'utente vuole visualizzare sullo schermo una lista contenente tutte le visite mediche presenti nel sistema. L'utente, a partire dalla home page, clicca sul pulsante "Sicurezza" e seleziona l'opzione "Visite mediche"; successivamente, visualizza correttamente la lista di tutte le visite mediche effettuate.

Ricerca visite mediche per dipendente

Il caso d'uso "Ricerca visite mediche per dipendente" si verifica quando l'utente vuole effettuare un filtraggio tra tutte le visite mediche per dipendente. Dopo essere entrato nella sezione "Visite mediche" e aver visualizzato la lista completa, l'utente clicca sul pulsante "Filtri", seleziona l'opzione "Filtra per dipendente", seleziona il dipendente tra quelli presenti nel sistema e visualizza sullo schermo una lista di tutte le visite mediche che hanno come paziente il dipendente selezionato.

Aggiungi visita medica

Il caso d'uso "Aggiungi visita medica" si verifica quando l'utente vuole inserire nel sistema una nuova visita medica. Dopo essere entrato nella sezione "Visite mediche" e aver visualizzato la lista di tutte le visite mediche, l'utente clicca sul pulsante "Aggiungi", successivamente compila i campi della schermata di inserimento e, se questi sono corretti, dopo aver cliccato sul pulsante "Conferma", completa correttamente l'operazione. In caso di dati incompleti o non corretti, il sistema ripropone l'inserimento dopo aver segnalato l'errore commesso.

Contatta dipendente

Il caso d'uso "Contatta dipendente" si verifica quando l'utente vuole contattare il dipendente che ha effettuato una certa visita medica. Dopo aver visualizzato la lista di tutte le visite mediche, l'utente ne seleziona una e clicca sul pulsante "Contatta dipendente"; successivamente il sistema recupera l'e-mail del dipendente che ha effettuato la visita medica selezionata e apre una nuova mail con il campo "Destinatario" precompilato con l'e-mail recuperata.

Visualizza dettagli visita medica

Il caso d'uso "Visualizza dettagli visita medica" si verifica quando l'utente vuole visualizzare sullo schermo tutti i dati di una visita medica specifica. Dopo aver visualizzato la lista di tutte le visite mediche, l'utente ne seleziona una e clicca sul pulsante "Modifica"; successivamente visualizza una nuova schermata contenente tutte le informazioni riguardanti la visita medica selezionata.

Elimina visita medica

Il caso d'uso "Elimina visita medica" si verifica quando l'utente vuole eliminare una visita medica presente nel sistema. Dopo aver visualizzato il dettaglio della visita medica da eliminare, l'utente clicca sul pulsante "Elimina" e, dopo aver confermato, termina correttamente l'operazione. La visita medica eliminata non è più presente nel sistema.

Modifica visita medica

Il caso d'uso "Modifica visita medica" si verifica quando l'utente vuole modificare i dati di una visita medica presente nel sistema. Dopo aver visualizzato il dettaglio della visita medica da modificare, l'utente ne modifica i dati e, se questi sono corretti, dopo aver cliccato sul pulsante "Conferma", termina correttamente l'operazione. In caso di dati incompleti o non corretti, il sistema ripropone la modifica dopo aver segnalato l'errore commesso.

1.8.4 Gestione del magazzino

Passiamo, ora, al quarto settore, ovvero quello che si occupa della gestione del magazzino e mettiamo in evidenza, in Figura 1.8, il rispettivo diagramma dei casi d'uso.

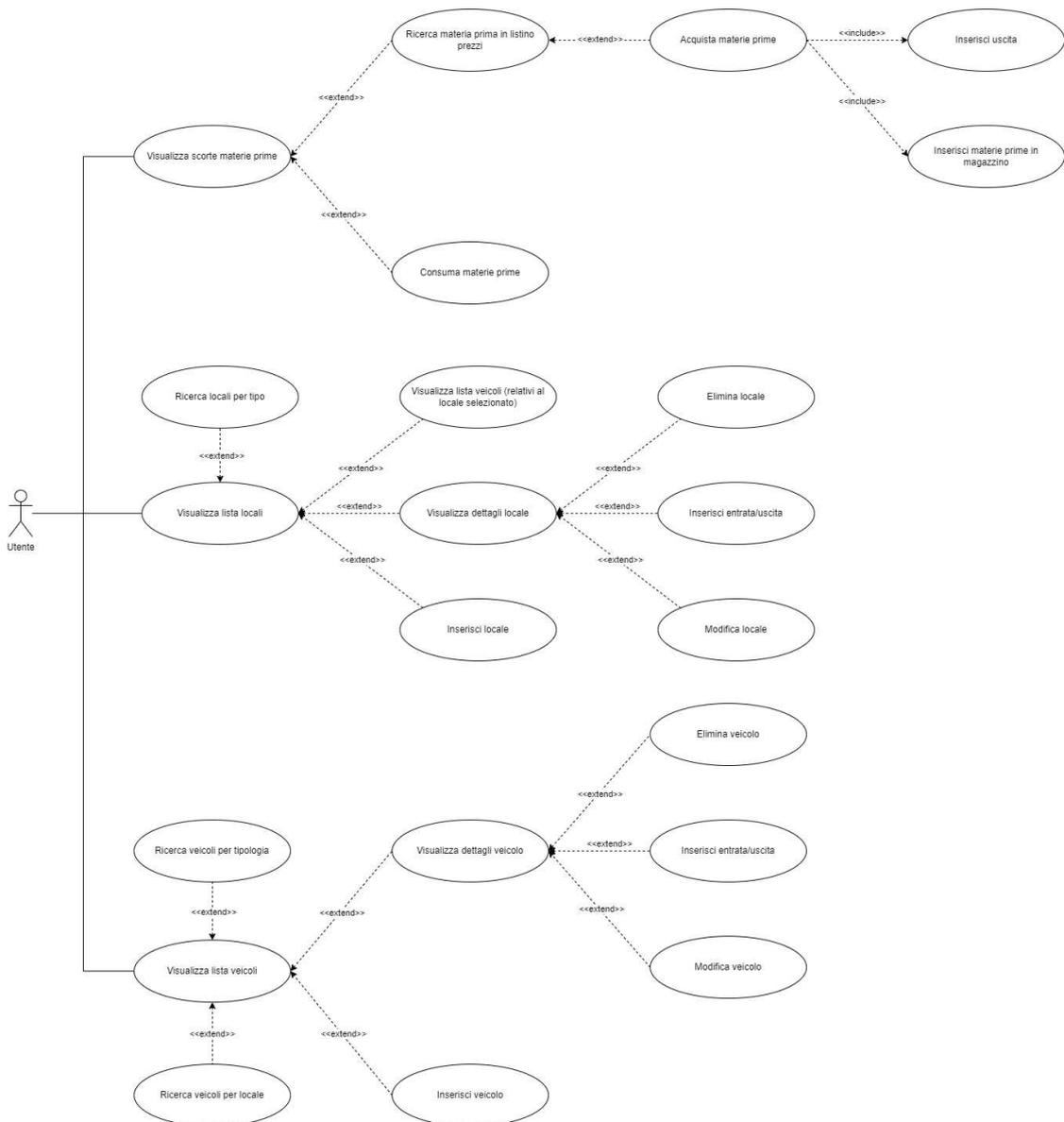


Figura 1.8: Diagramma dei casi d'uso relativi alla gestione del magazzino

Visualizza scorte materie prime

Il caso d'uso "Visualizza scorte materie prime" si verifica quando l'utente vuole visualizzare sullo schermo una lista contenente tutte le materie prime presenti nel sistema con le rispettive quantità. L'utente, a partire dalla home page, clicca sul pulsante "Magazzino"; successivamente visualizza correttamente la lista delle materie prime presenti in magazzino con le rispettive quantità.

Consuma materie prime

Il caso d'uso "Consuma materie prime" si verifica quando l'utente vuole ridurre la quantità di una materia prima presente in magazzino. L'utente, dopo essere entrato nella sezione "Magazzino" e aver visualizzato correttamente la lista di tutte le materie prime presenti, ne seleziona una e, se la sua quantità è maggiore di 0, indica il valore che desidera consumare. Dopo queste operazioni, la quantità del prodotto selezionato risulterà diminuita del valore indicato.

Ricerca materia prima in listino prezzi

Il caso d'uso "Ricerca materia prima in listino prezzi" si verifica quando l'utente vuole ricercare per nome nel tariffario un prodotto da acquistare. L'utente, dopo essere entrato nella sezione "Magazzino" e aver cliccato sul pulsante "Acquista materie prime", si ritrova in una schermata con un listino prezzi e una lista della spesa; a questo punto clicca sul pulsante "Filtri", seleziona l'opzione "Cerca prodotto per nome" e inserisce il nome del prodotto che desidera acquistare. Al termine di queste operazioni l'utente visualizza nella sezione dedicata al listino prezzi soltanto il prodotto desiderato.

Acquista materie prime

Il caso d'uso "Acquista materie prime" si verifica quando l'utente vuole acquistare nuove scorte di materiale da un fornitore. L'utente, dopo essere entrato nella pagina con il tariffario e la lista della spesa, seleziona un prodotto, ne indica la quantità e lo inserisce nella lista delle materie prime da acquistare. Dopo aver riempito la lista come desidera, l'utente clicca sul pulsante "Conferma acquisto" e completa l'operazione.

Inserisci materie prime in magazzino

Il caso d'uso "Inserisci materie prime in magazzino" si verifica parallelamente al caso d'uso "Acquista materie prime". Dopo aver completato l'operazione di acquisto, il sistema aggiorna in automatico il magazzino secondo i prodotti che sono stati acquistati dall'utente.

Inserisci uscita

Il caso d'uso "Inserisci uscita" si verifica parallelamente al caso d'uso "Acquista materie prime". Dopo aver completato l'operazione di acquisto, il sistema preleva il costo totale e inserisce nel sistema una nuova uscita con importo pari a quello appena prelevato.

Visualizza lista locali

Il caso d'uso "Visualizza lista locali" si verifica quando l'utente vuole visualizzare sullo schermo una lista contenente tutti i locali presenti nel sistema. L'utente, a partire dalla home

page, clicca sul pulsante "Locali"; successivamente il sistema visualizza la lista dei locali appartenenti alla ditta.

Ricerca locale per tipo

Il caso d'uso "Ricerca locale per tipo" si verifica quando l'utente vuole effettuare un filtraggio per tipo di tutti i locali. Dopo aver visualizzato la lista completa, l'utente clicca sul pulsante "Filtri", seleziona l'opzione "Filtra per tipo", seleziona il tipo tra quelli a disposizione (Garage, Appartamento, Ufficio, Magazzino) e visualizza sullo schermo una lista di tutti i locali del tipo selezionato.

Aggiungi locale

Il caso d'uso "Aggiungi locale" si verifica quando l'utente vuole inserire nel sistema un nuovo locale. Dopo essere entrato nella sezione "Locali" e aver visualizzato la lista di tutti i locali, l'utente clicca sul pulsante "Aggiungi", successivamente compila i campi della schermata di inserimento e, se questi sono corretti, dopo aver cliccato sul pulsante "Conferma", completa correttamente l'operazione. In caso di dati incompleti o non corretti, il sistema ripropone l'inserimento dopo aver segnalato l'errore commesso.

Visualizza dettagli locale

Il caso d'uso "Visualizza dettagli locale" si verifica quando l'utente vuole visualizzare sullo schermo tutti i dati di un locale specifico. Dopo aver visualizzato la lista di tutti i locali, l'utente ne seleziona uno e clicca sul pulsante "Modifica"; successivamente visualizza una nuova schermata contenente tutte le informazioni riguardo il locale selezionato.

Elimina locale

Il caso d'uso "Elimina locale" si verifica quando l'utente vuole eliminare un locale presente nel sistema. Dopo aver visualizzato il dettaglio del locale da eliminare, l'utente clicca sul pulsante "Elimina"; se il locale risulta occupato da almeno un veicolo l'eliminazione non va a buon fine; altrimenti, dopo aver dato conferma, l'operazione termina correttamente. Il locale eliminato non è più presente nel sistema.

Modifica locale

Il caso d'uso "Modifica locale" si verifica quando l'utente vuole modificare i dati di un locale presente nel sistema. Dopo aver visualizzato il dettaglio del locale da modificare, l'utente ne modifica i dati e, se questi sono corretti, dopo aver cliccato sul pulsante "Conferma", termina correttamente l'operazione. In caso di dati incompleti o non corretti, il sistema ripropone la modifica dopo aver segnalato l'errore commesso.

Visualizza lista veicoli (relativi al locale selezionato)

Il caso d'uso "Visualizza veicoli (relative al locale selezionato)" si verifica quando l'utente vuole visualizzare sullo schermo una lista contenente tutti i veicoli presenti in un certo locale. L'utente, dopo aver visualizzato la lista di tutti i locali, ne seleziona uno, clicca sul pulsante "Veicoli" e visualizza correttamente la lista dei veicoli presenti nel locale selezionato.

Visualizza lista veicoli

Il caso d'uso "Visualizza lista veicoli" si verifica quando l'utente vuole visualizzare sullo schermo una lista contenente tutti i veicoli presenti nel sistema. L'utente, a partire dalla home page, clicca sul pulsante "Veicoli"; successivamente visualizza correttamente la lista dei veicoli posseduti dalla ditta.

Ricerca veicolo per tipologia

Il caso d'uso "Ricerca veicolo per tipologia" si verifica quando l'utente vuole effettuare un filtraggio tra tutti i veicoli per tipologia. Dopo aver visualizzato la lista completa, l'utente clicca sul pulsante "Filtri", seleziona l'opzione "Filtra per tipologia", seleziona la tipologia tra quelle a disposizione (Autovettura, Autocarro, Motociclo, Furgone, Autobus, Industriale) e il sistema visualizza sullo schermo una lista di tutti i veicoli della tipologia selezionata.

Ricerca veicolo per locale

Il caso d'uso "Ricerca veicolo per locale" si verifica quando l'utente vuole effettuare un filtraggio tra tutti i veicoli per locale. Dopo aver visualizzato la lista completa, l'utente clicca sul pulsante "Filtri", seleziona l'opzione "Filtra per locale", seleziona il locale tra quelli presenti nel sistema e quest'ultimo visualizza sullo schermo una lista di tutti i veicoli presenti nel locale selezionato.

Aggiungi veicolo

Il caso d'uso "Aggiungi veicolo" si verifica quando l'utente vuole inserire nel sistema un nuovo veicolo. Dopo essere entrato nella sezione "Veicoli" e aver visualizzato la lista di tutti i veicoli, l'utente clicca sul pulsante "Aggiungi"; se non sono presenti locali dove poter collocare il nuovo veicolo, il sistema impedisce di effettuare l'operazione di aggiunta; in caso contrario, l'utente può completare l'operazione di inserimento compilando correttamente tutti i campi della schermata di inserimento. Dopo aver cliccato sul pulsante "Conferma", se tutti i dati sono stati inseriti correttamente, l'operazione di inserimento del nuovo veicolo va a buon fine. In caso di dati incompleti o non corretti, il sistema ripropone l'inserimento dopo aver segnalato l'errore commesso.

Visualizza dettagli veicolo

Il caso d'uso "Visualizza dettagli veicolo" si verifica quando l'utente vuole visualizzare sullo schermo tutti i dati di un veicolo specifico. Dopo aver visualizzato la lista di tutti i veicoli, l'utente ne seleziona uno e clicca sul pulsante "Modifica"; successivamente visualizza una nuova schermata contenente tutte le informazioni riguardo il veicolo selezionato.

Elimina veicolo

Il caso d'uso "Elimina veicolo" si verifica quando l'utente vuole eliminare un veicolo presente nel sistema. Dopo aver visualizzato il dettaglio del veicolo da eliminare, l'utente clicca sul pulsante "Elimina" e, dopo aver confermato, termina correttamente l'operazione. Il veicolo eliminato non è più presente nel sistema.

Modifica veicolo

Il caso d'uso "Modifica veicolo" si verifica quando l'utente vuole modificare i dati di un veicolo presente nel sistema. Dopo aver visualizzato il dettaglio del veicolo da modificare, l'utente ne modifica i dati e, se questi sono corretti, dopo aver cliccato sul pulsante "Conferma", termina correttamente l'operazione. In caso di dati incompleti o non corretti, il sistema ripropone la modifica dopo aver segnalato l'errore commesso.

1.8.5 Gestione dei fornitori

Analizziamo ora il quinto settore, quello che si riferisce alla gestione dei fornitori; anche in questo caso mostriamo il rispettivo diagramma dei casi d'uso in Figura 1.9.

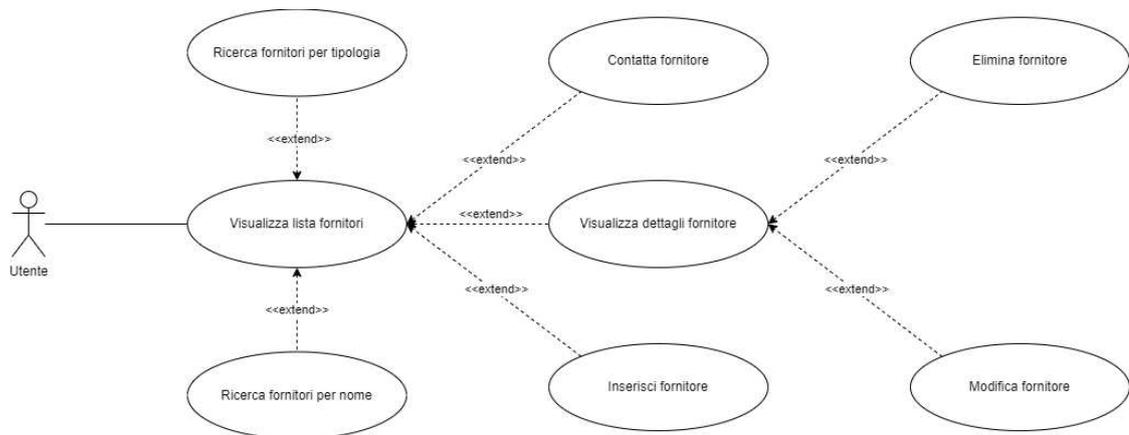


Figura 1.9: Diagramma dei casi d'uso relativo alla gestione dei fornitori

Visualizza lista fornitori

Il caso d'uso "Visualizza lista fornitori" si verifica quando l'utente vuole visualizzare sullo schermo una lista contenente tutti i fornitori presenti nel sistema. L'utente, a partire dalla home page, clicca sul pulsante "Fornitori"; successivamente visualizza correttamente la lista di tutti i fornitori dell'impresa.

Ricerca fornitore per tipologia

Il caso d'uso "Ricerca fornitore per tipologia" si verifica quando l'utente vuole effettuare un filtraggio tra tutti i fornitori per tipologia. Dopo aver visualizzato la lista completa, l'utente clicca sul pulsante "Filtri", seleziona l'opzione "Filtra per tipologia", seleziona la tipologia tra quelle a disposizione e visualizza sullo schermo una lista di tutti i fornitori della tipologia selezionata.

Ricerca fornitore per nome

Il caso d'uso "Ricerca fornitore per nome" si verifica quando l'utente vuole ricercare per nome un fornitore specifico. Dopo aver visualizzato la lista completa, l'utente clicca sul pulsante "Filtri", seleziona l'opzione "Filtra per nome", inserisce il nome del fornitore desiderato e visualizza sullo schermo il fornitore con il nome dato in input.

Contatta fornitore

Il caso d'uso "Contatta fornitore" si verifica quando l'utente vuole contattare un certo fornitore. Dopo aver visualizzato la lista di tutti i fornitori, l'utente ne seleziona uno e clicca sul pulsante "Contatta fornitore"; successivamente il sistema recupera l'e-mail del fornitore selezionato e apre una nuova mail con il campo "Destinatario" precompilato con l'e-mail recuperata.

Aggiungi fornitore

Il caso d'uso "Aggiungi fornitore" si verifica quando l'utente vuole inserire nel sistema un nuovo fornitore. Dopo essere entrato nella sezione "Fornitori" e aver visualizzato la lista di tutti i fornitori, l'utente clicca sul pulsante "Aggiungi"; successivamente compila i campi della schermata di inserimento e, se questi sono corretti, dopo aver cliccato sul pulsante "Conferma", completa correttamente l'operazione. In caso di dati incompleti o non corretti, il sistema ripropone l'inserimento dopo aver segnalato l'errore commesso.

Visualizza dettagli fornitore

Il caso d'uso "Visualizza dettagli fornitore" si verifica quando l'utente vuole visualizzare sullo schermo tutti i dati di un fornitore specifico. Dopo aver visualizzato la lista di tutti i fornitori, l'utente ne seleziona uno e clicca sul pulsante "Modifica"; successivamente visualizza una nuova schermata contenente tutte le informazioni riguardo il fornitore selezionato.

Elimina fornitore

Il caso d'uso "Elimina fornitore" si verifica quando l'utente vuole eliminare un fornitore presente nel sistema. Dopo aver visualizzato il dettaglio del fornitore da eliminare, l'utente clicca sul pulsante "Elimina" e, dopo aver confermato, termina correttamente l'operazione. Il fornitore eliminato non è più presente nel sistema.

Modifica fornitore

Il caso d'uso "Modifica fornitore" si verifica quando l'utente vuole modificare i dati di un fornitore presente nel sistema. Dopo aver visualizzato il dettaglio del fornitore da modificare, l'utente ne modifica i dati e, se questi sono corretti, dopo aver cliccato sul pulsante "Conferma", il sistema termina correttamente l'operazione. In caso di dati incompleti o non corretti, il sistema ripropone la modifica dopo aver segnalato l'errore commesso.

1.8.6 Gestione della contabilità

Passiamo al sesto settore, quello che si occupa della gestione della contabilità, di cui facciamo vedere il diagramma dei casi d'uso in Figura 1.10.

Visualizza lista analisi di bilancio

Il caso d'uso "Visualizza lista analisi di bilancio" si verifica quando l'utente vuole visualizzare sullo schermo una lista contenente tutti i bilanci presenti nel sistema. L'utente, a partire dalla home page, clicca sul pulsante "Contabilità" e seleziona l'opzione "Bilanci"; successivamente, il sistema visualizza la lista di tutti i bilanci effettuati dall'impresa.

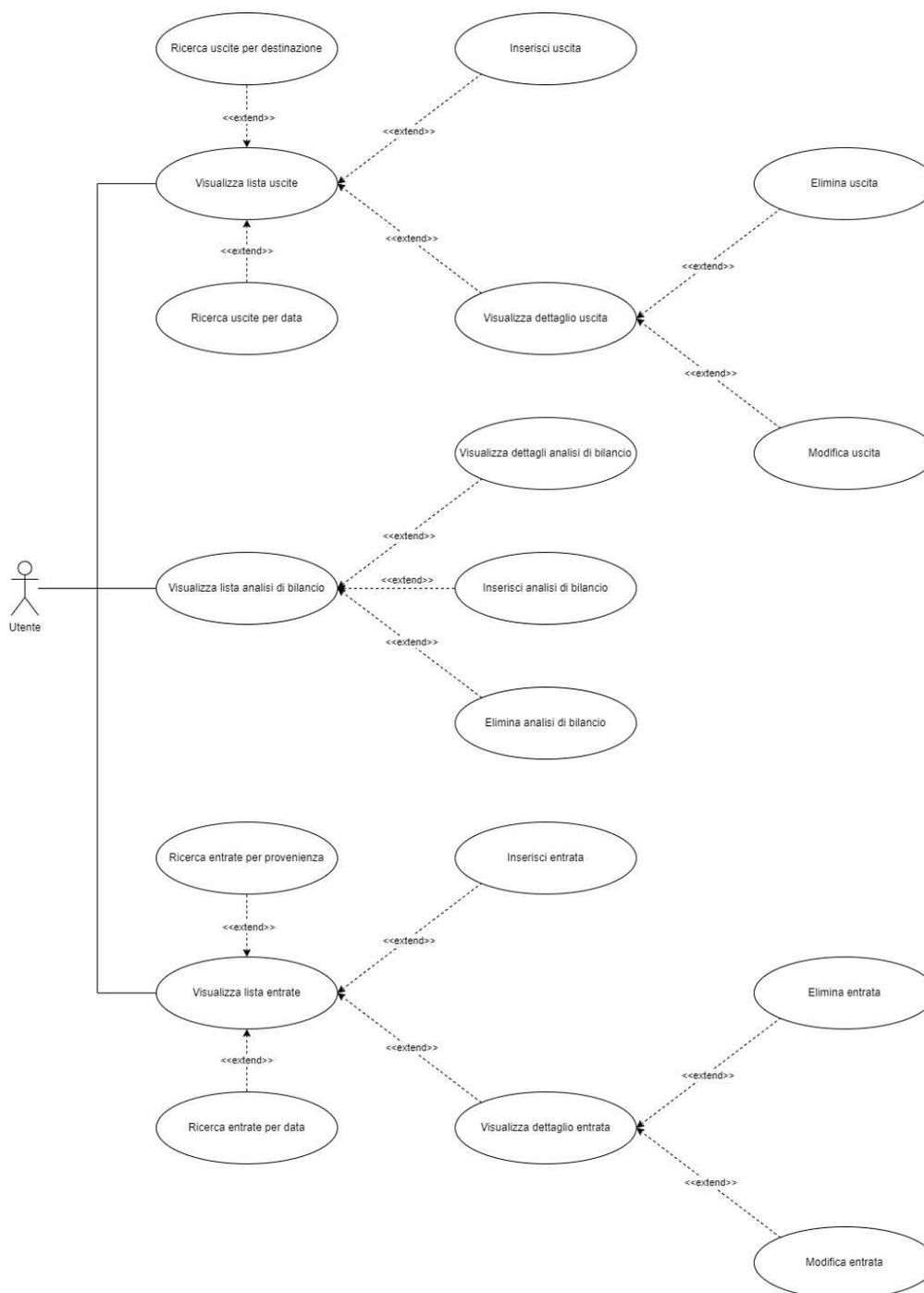


Figura 1.10: Diagramma dei casi d'uso relativo alla gestione della contabilità

Aggiungi analisi di bilancio

Il caso d'uso "Aggiungi analisi di bilancio" si verifica quando l'utente vuole inserire nel sistema un nuovo bilancio. Dopo essere entrato nella sezione "Bilanci" e aver visualizzato la lista di tutti i bilanci, l'utente clicca sul pulsante "Aggiungi", successivamente seleziona l'anno di interesse e, se non è presente nessun bilancio per l'anno indicato, dopo che il sistema ha calcolato automaticamente i valori di interesse a partire dalle entrate e dalle uscite dell'anno in questione, esso completa correttamente l'operazione. Nel caso in cui l'anno selezionato abbia già un bilancio presente nel sistema, l'operazione di inserimento non va a buon fine.

Visualizza dettagli analisi di bilancio

Il caso d'uso "Visualizza dettagli analisi di bilancio" si verifica quando l'utente vuole visualizzare sullo schermo tutti i dati di un bilancio specifico. Dopo aver visualizzato la lista di tutti i bilanci, l'utente ne seleziona uno e clicca sul pulsante "Visualizza dettagli"; successivamente, il sistema visualizza una nuova schermata contenente tutte le informazioni riguardo il bilancio selezionato.

Elimina analisi di bilancio

Il caso d'uso "Elimina analisi di bilancio" si verifica quando l'utente vuole eliminare un bilancio presente nel sistema. Dopo aver visualizzato la lista di tutti i bilanci, l'utente ne seleziona uno, clicca sul pulsante "Elimina" e, dopo aver confermato, termina correttamente l'operazione. Il bilancio eliminato non è più presente nel sistema.

Visualizza lista entrate

Il caso d'uso "Visualizza lista entrate" si verifica quando l'utente vuole visualizzare sullo schermo una lista contenente tutte le entrate presenti nel sistema. L'utente, a partire dalla home page, clicca sul pulsante "Contabilità" e seleziona l'opzione "Entrate"; successivamente visualizza correttamente la lista di tutte le entrate che ha avuto l'impresa.

Ricerca entrate per data

Il caso d'uso "Ricerca entrate per data" si verifica quando l'utente vuole effettuare un filtraggio tra tutte le entrate per data. Dopo aver visualizzato la lista completa, l'utente clicca sul pulsante "Filtri", seleziona l'opzione "Filtra per data", indica la data iniziale e quella finale e visualizza sullo schermo una lista di tutte le entrate avute nel lasso di tempo indicato.

Ricerca entrate per provenienza

Il caso d'uso "Ricerca entrata per provenienza" si verifica quando l'utente vuole effettuare un filtraggio tra tutte le entrate per provenienza. Dopo aver visualizzato la lista completa, l'utente clicca sul pulsante "Filtri", seleziona l'opzione "Filtra per provenienza", seleziona la provenienza tra quelle a disposizione e il sistema visualizza sullo schermo una lista di tutte le entrate relative alla provenienza selezionata.

Aggiungi entrata

Il caso d'uso "Aggiungi entrata" si verifica quando l'utente vuole inserire nel sistema una nuova entrata. Dopo essere entrato nella sezione "Entrate" e aver visualizzato la lista di tutte le entrate, l'utente clicca sul pulsante "Aggiungi"; successivamente compila i campi della schermata di inserimento e, se questi sono corretti, dopo aver cliccato sul pulsante "Conferma", completa correttamente l'operazione. In caso di dati incompleti o non corretti, il sistema ripropone l'inserimento dopo aver segnalato l'errore commesso.

Visualizza dettagli entrata

Il caso d'uso "Visualizza dettagli entrata" si verifica quando l'utente vuole visualizzare sullo schermo tutti i dati di un'entrata specifica. Dopo aver visualizzato la lista di tutte le entrate, l'utente ne seleziona una e clicca sul pulsante "Modifica"; successivamente il

sistema visualizza una nuova schermata contenente tutte le informazioni riguardo l'entrata selezionata.

Elimina entrata

Il caso d'uso "Elimina entrata" si verifica quando l'utente vuole eliminare un'entrata presente nel sistema. Dopo aver visualizzato il dettaglio dell'entrata da eliminare, l'utente clicca sul pulsante "Elimina" e, dopo che egli ha confermato, il sistema termina l'operazione. L'entrata eliminata non è più presente nel sistema.

Modifica entrata

Il caso d'uso "Modifica entrata" si verifica quando l'utente vuole modificare i dati di un'entrata presente nel sistema. Dopo aver visualizzato il dettaglio dell'entrata da modificare, l'utente ne modifica i dati e, se questi sono corretti, dopo aver cliccato sul pulsante "Conferma", il sistema termina correttamente l'operazione. In caso di dati incompleti o non corretti, il sistema ripropone la modifica dopo aver segnalato l'errore commesso.

Visualizza lista uscite

Il caso d'uso "Visualizza lista uscite" si verifica quando l'utente vuole visualizzare sullo schermo una lista contenente tutte le uscite presenti nel sistema. L'utente, a partire dalla home page, clicca sul pulsante "Contabilità" e seleziona l'opzione "Uscite"; successivamente visualizza correttamente la lista di tutte le uscite che ha avuto l'impresa.

Ricerca uscite per data

Il caso d'uso "Ricerca uscite per data" si verifica quando l'utente vuole effettuare un filtraggio tra tutte le uscite per data. Dopo aver visualizzato la lista completa, l'utente clicca sul pulsante "Filtri", seleziona l'opzione "Filtra per data", indica la data iniziale e quella finale e il sistema visualizza sullo schermo una lista di tutte le uscite effettuate nel lasso di tempo indicato.

Ricerca uscite per destinazione

Il caso d'uso "Ricerca uscita per destinazione" si verifica quando l'utente vuole effettuare un filtraggio tra tutte le uscite per destinazione. Dopo aver visualizzato la lista completa, l'utente clicca sul pulsante "Filtri", seleziona l'opzione "Filtra per destinazione", seleziona la destinazione tra quelle a disposizione e il sistema visualizza sullo schermo una lista di tutte le uscite relative alla destinazione selezionata.

Aggiungi uscita

Il caso d'uso "Aggiungi uscita" si verifica quando l'utente vuole inserire nel sistema una nuova uscita. Dopo essere entrato nella sezione "Uscite" e aver visualizzato la lista di tutte le uscite, l'utente clicca sul pulsante "Aggiungi", successivamente compila i campi della schermata di inserimento e, se questi sono corretti, dopo aver cliccato sul pulsante "Conferma", completa correttamente l'operazione. In caso di dati incompleti o non corretti, il sistema ripropone l'inserimento dopo aver segnalato l'errore commesso.

Visualizza dettagli uscita

Il caso d'uso "Visualizza dettagli uscita" si verifica quando l'utente vuole visualizzare sullo schermo tutti i dati di un'uscita specifica. Dopo aver visualizzato la lista di tutte le uscite, l'utente ne seleziona una e clicca sul pulsante "Modifica"; successivamente il sistema visualizza una nuova schermata contenente tutte le informazioni riguardo l'uscita selezionata.

Elimina uscita

Il caso d'uso "Elimina uscita" si verifica quando l'utente vuole eliminare un'uscita presente nel sistema. Dopo aver visualizzato il dettaglio dell'uscita da eliminare, l'utente clicca sul pulsante "Elimina" e, dopo aver confermato, il sistema termina correttamente l'operazione. L'uscita eliminata non è più presente nel sistema.

Modifica uscita

Il caso d'uso "Modifica uscita" si verifica quando l'utente vuole modificare i dati di un'uscita presente nel sistema. Dopo aver visualizzato il dettaglio dell'uscita da modificare, l'utente ne modifica i dati e, se questi sono corretti, dopo aver cliccato sul pulsante "Conferma", il sistema termina correttamente l'operazione. In caso di dati incompleti o non corretti, il sistema ripropone la modifica dopo aver segnalato l'errore commesso.

Inserisci entrata

Il caso d'uso "Inserisci entrata" può verificarsi anche quando l'utente vuole inserire una nuova entrata nel sistema a partire da altre parti del software (caso d'uso esteso che si trova negli altri settori). L'utente può inserire questo tipo di entrate a partire dai lavori, dai veicoli oppure dai locali. La descrizione viene fatta per i lavori (per locali e veicoli è la stessa cosa; l'unica differenza è che va indicato se l'entrata proviene da una vendita oppure da un noleggio/affitto; nel caso in cui non ci sia un importo, come nei lavori, il sistema chiede all'utente di inserirlo manualmente). Dopo aver visualizzato il dettaglio del lavoro da cui si vuole ricavare l'entrata, l'utente clicca sul pulsante "Inserisci pagamento in entrata"; il sistema, a partire dall'importo del lavoro, inserisce in memoria una nuova entrata.

Inserisci uscita

Il caso d'uso "Inserisci uscita" può verificarsi anche quando l'utente vuole inserire una nuova uscita nel sistema a partire da altre parti del software (caso d'uso esteso che si trova negli altri settori). L'utente può inserire questo tipo di uscite a partire dai dipendenti, dai veicoli oppure dai locali. La descrizione viene fatta per i dipendenti (per locali e veicoli è la stessa cosa; l'unica differenza è che va indicato se l'uscita proviene da un acquisto oppure da un noleggio/affitto; nel caso in cui non ci sia un importo, come nei dipendenti, il sistema chiede all'utente di inserirlo manualmente). Dopo aver visualizzato il dettaglio del dipendente a cui si vuole intestare l'uscita, l'utente clicca sul pulsante "Inserisci pagamento in uscita"; il sistema, a partire dallo stipendio del contratto del dipendente, inserisce in memoria una nuova uscita.

1.8.7 Calcolo statistiche

Terminiamo con il settimo e ultimo settore ovvero quello relativo al calcolo delle statistiche di cui, allo stesso modo dei settori precedenti, illustriamo in Figura 1.11 il diagramma dei casi d'uso relativo.

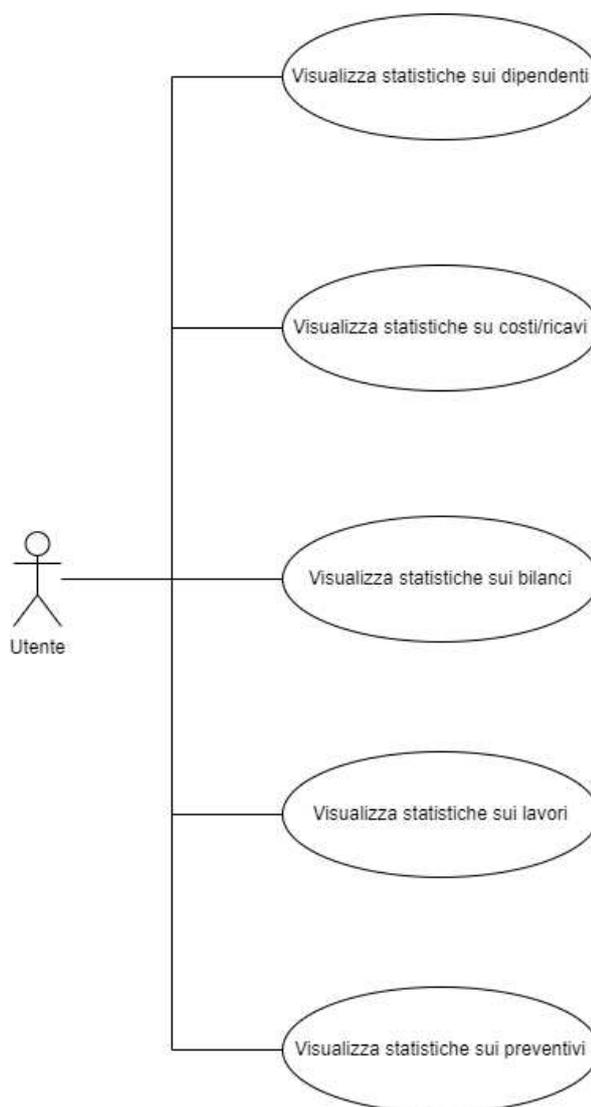


Figura 1.11: Diagramma dei casi d'uso relativo al calcolo delle statistiche

Visualizza statistiche sui dipendenti

Il caso d'uso "Visualizza statistiche sui dipendenti" si verifica quando l'utente vuole visualizzare sullo schermo le statistiche riguardanti gli stipendi dei dipendenti. L'utente, dalla home page, dopo essere entrato nella sezione "Statistiche", clicca sul pulsante "Dipendenti" e il sistema visualizza un grafico con le statistiche sugli stipendi dei dipendenti.

Visualizza statistiche sui costi/ricavi

Il caso d'uso "Visualizza statistiche sui costi/ricavi" si verifica quando l'utente vuole visualizzare sullo schermo le statistiche riguardanti i costi e i ricavi in un determinato lasso di tempo. Dopo essere entrato nella sezione "Statistiche" e aver cliccato sul pulsante "Costi/Ricavi", l'utente inserisce la data iniziale e la data finale; il sistema recupera i costi e i ricavi nel lasso di tempo specificato e li mostra sullo schermo.

Visualizza statistiche sui bilanci

Il caso d'uso "Visualizza statistiche sui bilanci" si verifica quando l'utente vuole visualizzare sullo schermo le statistiche riguardanti i saldi complessivi dei bilanci effettuati dall'impresa. L'utente, dalla home page, dopo essere entrato nella sezione "Statistiche", clicca sul pulsante "Bilanci" e il sistema visualizza un grafico con i saldi complessivi per ogni bilancio presente nel sistema.

Visualizza statistiche sui lavori

Il caso d'uso "Visualizza statistiche sui lavori" si verifica quando l'utente vuole visualizzare sullo schermo le statistiche riguardanti gli importi dei lavori commissionati all'impresa. L'utente, dalla home page, dopo essere entrato nella sezione "Statistiche", clicca sul pulsante "Lavori" e visualizza correttamente un grafico con i saldi complessivi per ogni bilancio presente nel sistema. Il sistema mostra sullo schermo anche una percentuale dei lavori interamente pagati dal cliente (sul totale dei lavori commissionati).

Visualizza statistiche sui preventivi

Il caso d'uso "Visualizza statistiche sui preventivi" si verifica quando l'utente vuole visualizzare sullo schermo la statistica riguardante la percentuale dei preventivi accettati su tutti quelli stipulati e presenti nel sistema. L'utente, dalla home page, dopo essere entrato nella sezione "Statistiche", clicca sul pulsante "Preventivi" e visualizza correttamente la percentuale desiderata.

Progettazione del database

In questo secondo capitolo ci occuperemo della progettazione del database. Dopo aver brevemente dato qualche richiamo su cosa sia una base di dati, a cosa serve un DBMS e perché utilizziamo questa strategia piuttosto che altre, inizieremo subito con il definire le entità fondamentali del sistema, ovvero gli oggetti con cui interagirà il nostro software. In seguito introdurremo le relazioni tra le entità ovvero la parte fondamentale di un database di tipo Entità/Relazione, per poi completare lo schema del diagramma E/R. Infine, dopo aver chiarito i vincoli sui dati del sistema, l'ultimo passaggio sarà quello di traduzione verso il modello relazionale in cui introdurremo le tabelle con i rispettivi identificatori e attributi che formeranno poi, nella fase di implementazione, attraverso codifica SQL, la nostra base di dati.

2.1 Scelta database

2.1.1 Perché utilizzare un database

La descrizione del mondo reale o, meglio, di porzioni del mondo reale può essere compiuta spesso tramite la “schedatura” di informazioni organizzate in modo strutturato ed omogeneo; ad esempio, la descrizione dei dipendenti di un’azienda prevede, per ogni dipendente, una scheda personale contenente i dati anagrafici e quelli necessari all’azienda per gestire il dipendente; la descrizione del territorio usata dalla Protezione Civile prevede delle mappe, a ciascun punto delle quali vengono associate informazioni di carattere geologico o informazioni spazio-temporali. Le informazioni con tipologia analoga a quelle qui esemplificate possono essere efficientemente organizzate e memorizzate in modo da consentirne un trattamento informatizzato. L’efficienza dell’organizzazione comporta semplicità di gestione e aggiornamento delle informazioni, ma anche velocità e semplicità di reperimento e trattamento. L’archivio dei dipendenti è un problema risolto ormai con strumenti consolidati sin dalla fine degli anni '70; più recenti sono, invece, le tecnologie per l’organizzazione dei sistemi georeferenziati (GIS, Geographical Information System) sulle quali la ricerca è ancora vivace.

La memorizzazione e la gestione dei dati è, quindi, un’applicazione fondamentale dell’informatica. Con i calcolatori è possibile inserire, modificare, cancellare, organizzare e recuperare dati; diciamo che un dato è persistente se è mantenuto integralmente quando il calcolatore viene spento oppure si verifica un errore. Normalmente ci affidiamo alla memorizzazione e alla gestione dei dati quando, per esempio, effettuiamo operazioni di gestione degli ordini, del personale, della produzione e degli approvvigionamenti.

Esistono sistemi software specifici, detti sistemi per la gestione di basi di dati (o DBMS, DataBase Management System), che forniscono linguaggi di alto livello dedicati alla gestione dei dati. Tali sistemi offrono un insieme di proprietà aggiuntive rispetto ai linguaggi di programmazione tradizionali, tra cui ricordiamo maggiore efficienza, robustezza ai guasti e indipendenza del linguaggio dalla specifica organizzazione dei dati in memoria, benché usino anch'essi, al loro interno, le tecniche di organizzazione degli archivi.

Il compito di memorizzare l'informazione è "di norma" svolto dalla mente umana; come avvenga questo processo nel cervello è un fatto alquanto straordinario e in buona misura ancora misterioso. In confronto ad esso, un calcolatore è soggetto a enormi limitazioni; affinché sia possibile estrarre delle informazioni, i dati devono essere raccolti in base a precise regole e organizzati in strutture piuttosto rigide, dette basi di dati (o, in inglese, database).

Una base di dati è una collezione di informazioni che vengono organizzate e gestite da un sistema software specifico, il DBMS, componente essenziale per la realizzazione di un sistema informativo.

Le basi di dati offrono un software unificato e organizzato per la gestione dei dati. Il loro effetto può essere meglio apprezzato confrontandole con l'accesso ai dati convenzionale da parte di un linguaggio di programmazione che opera senza di esse. Quando i programmi applicativi usano i file per come vengono forniti dal sistema operativo, i principali problemi sono causati dalla condivisione dei file stessi. Quando gli stessi archivi vengono utilizzati da più programmi, si possono verificare molti problemi, tra i quali segnaliamo:

- inconsistenza e ridondanza dei dati;
- riservatezza dei dati;
- integrità dei dati;
- concorrenza.

I DBMS possono ovviare, se ben utilizzati, a queste difficoltà, dal momento che, grazie ad essi:

- i dati hanno una sola rappresentazione, anche se vengono usati da molti utenti, e questo riduce il pericolo di ridondanza e inconsistenza;
- l'accesso alla base di dati viene disciplinato dal DBMS, al cui interno è possibile definire le modalità di utilizzo di ciascun dato da parte di ciascun utente, proteggendo, così, la riservatezza dei dati;
- alcuni vincoli di integrità, che definiscono la correttezza della base di dati, possono essere definiti all'interno del DBMS e venire automaticamente verificati dal sistema, impedendo, così, l'esecuzione di programmi che li violino;
- l'accesso concorrente ai dati è controllato dal DBMS tramite operazioni che consentono la mutua esclusione dei programmi, senza peraltro rallentarne eccessivamente l'esecuzione.

2.1.2 Perché scegliere un modello relazionale

Un modello dei dati è un insieme di concetti utilizzati per organizzare i dati di interesse e per descriverne la struttura in modo che essa risulti comprensibile a un elaboratore. Il modello relazionale dei dati, tutt'ora il più diffuso e didatticamente più interessante, permette di definire tipi per mezzo del costruttore "relazione", che consente di organizzare i dati in

insiemi di record a struttura fissa. Una relazione viene spesso rappresentata per mezzo di una tabella, le cui righe rappresentano specifici record e le cui colonne corrispondono ai campi dei record. Il modello relazionale, definito formalmente agli inizi degli anni '70 e affermatosi nel decennio successivo, è, come detto, il più diffuso e viene utilizzato in questo progetto come modello di riferimento.

Oltre al modello relazionale sono stati definiti altri tipi di modelli che elenchiamo di seguito:

- *modello gerarchico*: basato sull'uso di strutture ad albero;
- *modello reticolare*: basato sull'uso di grafi;
- *modello ad oggetti*: estende alle basi di dati il paradigma di programmazione ad oggetti;
- *modello XML*: rivisitazione del modello gerarchico, in cui, però, i dati vengono presentati assieme alla loro descrizione e non devono sottostare rigidamente ad un'unica struttura logica;
- *modelli semistrutturati e flessibili*: sviluppati nel contesto dei cosiddetti sistemi NoSQL, che cercano di superare, in specifici contesti applicativi, alcune delle limitazioni dei sistemi relazionali.

Il modello relazionale si basa su due concetti, ovvero relazione e tabella, di natura diversa ma facilmente riconducibili l'uno all'altro. La nozione di relazione proviene dalla matematica, in particolare dalla teoria degli insiemi, mentre il concetto di tabella è semplice e intuitivo. La presenza contemporanea di questi due concetti, l'uno formale e l'altro intuitivo, è responsabile del grande successo ottenuto dal modello relazionale. Infatti, le tabelle risultano naturali e comprensibili anche per gli utenti finali (che spesso le utilizzano in tanti contesti per scopi diversi, senza riferimento alle basi di dati). D'altra parte, la disponibilità di una formalizzazione semplice e precisa ha permesso anche uno sviluppo teorico a supporto del modello con risultati di interesse concreto.

2.2 Entità fondamentali

Come approccio iniziale, dopo aver attentamente studiato le informazioni reperite dall'analisi dei requisiti, abbiamo deciso di individuare le entità fondamentali della nostra base di dati. Di seguito le elenchiamo, accompagnate ognuna da una breve descrizione:

- *lavoro*: qualsiasi attività o servizio che effettua l'impresa, sotto la commissione di un cliente;
- *preventivo*: documento redatto dall'impresa con il cliente che descrive una qualsiasi proposta di lavoro;
- *cliente*: qualsiasi individuo che, attraverso lavori e/o preventivi, si interfaccia con l'impresa;
- *dipendente*: qualsiasi individuo che, attraverso un contratto stipulato, si interfaccia con l'impresa;
- *contratto*: documento redatto dall'impresa con il dipendente che regola il patto di lavoro tra le due parti;

- *visita medica*: qualsiasi incontro tra un dipendente e un medico durante il quale si valuta la salute del paziente;
- *materia prima*: tutto ciò di materiale che può essere utilizzato dall'impresa;
- *locale*: qualsiasi unità di spazio posseduto dall'impresa;
- *veicolo*: qualsiasi mezzo posseduto dall'impresa e posto in un particolare locale;
- *entrata*: qualsiasi somma di denaro che ha una certa provenienza;
- *uscita*: qualsiasi somma di denaro che ha una certa destinazione.

2.3 Diagramma E/R

Il modello Entità-Relazione (nel seguito utilizzeremo spesso per questo termine l'abbreviazione E/R) è un modello concettuale di dati e, come tale, fornisce una serie di strutture, dette costrutti, atte a descrivere la realtà di interesse in una maniera facile da comprendere e che prescinde dai criteri di organizzazione dei dati nei calcolatori. Nello schema E/R mostrato in Figura 2.1 possiamo visualizzare alcuni dei costrutti che questo tipo di modello mette a disposizione:

- *entità*: rappresentano classi di oggetti che hanno proprietà comuni ed esistenza "autonoma" ai fini dell'applicazione di interesse: "Lavori" e "Clienti" sono esempi di entità individuate nel nostro sistema;
- *relazioni*: rappresentano legami logici tra due o più entità: "Partecipazione" è un esempio di relazione tra le entità "Lavori" e "Clienti";
- *attributi semplici*: descrivono le proprietà elementari di entità o relazioni: "Nome", "Cognome" e "Codice fiscale" sono attributi dell'entità "Clienti";
- *attributi composti*: gruppo di attributi di una medesima entità o relazione che presentano affinità nel loro significato o uso: "Dati di interesse" raggruppa gli attributi "Totale entrate", "Totale uscite" e "Saldo complessivo" dell'entità "Bilanci";
- *cardinalità delle relazioni*: vengono specificate per ciascuna partecipazione di entità a una relazione e descrivono il numero minimo e massimo di occorrenze di relazione a cui una occorrenza di entità può partecipare; nella relazione "Partecipazione" tra "Clienti" e "Lavori" tutti i lavori devono essere associati a uno e un solo cliente, mentre un cliente potrebbe essere impegnato in uno o più lavori, ma anche non essere impegnato in nessun lavoro (perché, magari, è stato inserito nel sistema a partire dall'aggiunta di un nuovo preventivo);
- *identificatore interno*: specificato per ciascuna entità, descrive il concetto dello schema che permette di identificare univocamente le occorrenze dell'entità a cui si riferisce; gli identificatori delle entità "Lavori" e "Clienti" sono rispettivamente "Denominazione" e "Codice fiscale".

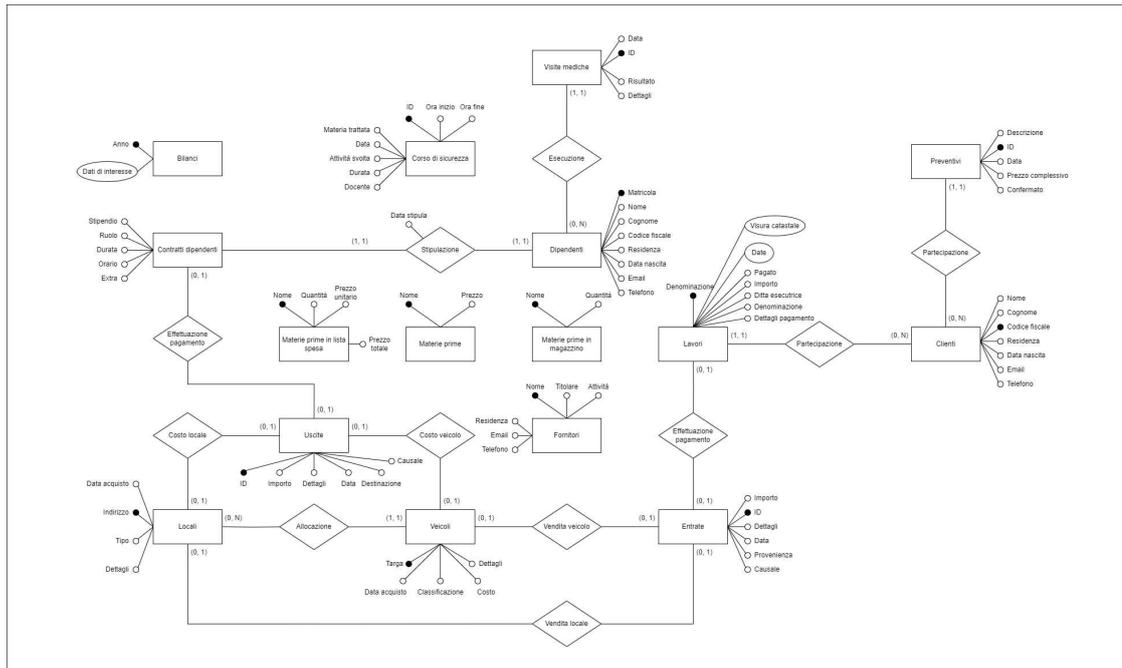


Figura 2.1: Esempio di diagramma E/R

2.4 Vincoli sui dati

I vincoli sui dati sono dei vincoli intra-relazionali di integrità, ovvero il loro soddisfacimento è definito rispetto a singole relazioni della base di dati. Nello specifico, si tratta di vincoli che esprimono condizioni sui valori di ciascun dato, indipendentemente dagli altri dati. Elenchiamo, di seguito, i seguenti vincoli, i quali derivano da regole aziendali che ci sono state segnalate dall'impresa:

- **attività:** vincolo relativo all'entità "Fornitori"; può avere valori "Fornitori di Materiali da Costruzione", "Rivenditori di Ferramenta", "Fornitori di Materiali Isolanti e Impermeabilizzanti", "Noleggiatori di Attrezzature", "Fornitori di Attrezzature per la Sicurezza", "Produttori e Rivenditori di Pavimentazioni", "Fornitori di Impianti e Sistemi Tecnologici", "Rivenditori di Infissi e Serramenti", "Fornitori di Vernici e Finiture", "Fornitori di Arredo Urbano e Verde", "Fornitori di Servizi di Progettazione e Consulenza", "Imprese di Trasporto e Logistica", "Rivenditori di Attrezzature di Sollevamento e Movimentazione", "Laboratori di Analisi e Controllo Qualità", "Distributori di Prodotti Chimici per Edilizia", "Rivenditori di Veicoli";
- **attività svolta:** vincolo relativo all'entità "Corsi di sicurezza"; può avere valori "Formazione", "Intervento", "Formazione e Intervento";
- **classificazione:** vincolo relativo all'entità "Veicoli"; può essere "Autovettura", "Autocarro", "Motociclo", "Furgone", "Autobus", "Industriale" (dal codice della strada);
- **codice fiscale:** vincolo relativo alle entità "Clienti" e "Dipendenti"; deve rispettare le regole previste dallo Stato Italiano per la costruzione del codice fiscale;
- **confermato:** vincolo relativo all'entità "Preventivi" per i clienti; può essere "Sì" o "No";
- **contratto:** vincolo relativo all'entità "Dipendenti"; è un intero maggiore o uguale di 1;

- *data*: ogni volta che troviamo questo attributo, bisogna considerare una stringa che rappresenta una data scritta correttamente in formato gg-mm-aaaa;
- *destinazione*: vincolo relativo all'entità "Uscite"; può avere valori "Stipendi", "Locali", "Veicoli", "Materiali e Forniture", "Spese Amministrative", "Permessi e Licenze", "Servizi Professionali", "Marketing e Pubblicità", "Spese Legali", "Spese Bancarie", "Altro";
- *dati di interesse*: vincolo relativo all'entità "Bilanci"; totale entrate e totale uscite devono essere necessariamente maggiori di 0, saldo complessivo può essere anche negativo (supponendo possibili conti in rosso);
- *email*: ovunque si trovi deve essere espressa nel formato <nome>@<dominio>;
- *id*: ovunque si trovi è un intero maggiore o uguale di 1;
- *matricola*: vincolo relativo all'entità "Dipendenti"; è un intero maggiore o uguale di 1;
- *pagato*: vincolo relativo all'entità "Lavori" per i clienti; può essere "Sì" o "No";
- *provenienza*: vincolo relativo all'entità "Entrate"; può avere valori "Lavori", "Locali", "Veicoli", "Servizi di Consulenza", "Sovvenzioni e Incentivi", "Recupero di Crediti", "Vendita di Materiali Residui", "Finanziamenti", "Altro";
- *quantità*: vincolo relativo alle entità "Materie prime in lista spesa" e "Materie prime in magazzino"; deve essere un numero maggiore o uguale a 0;
- *risultato*: vincolo relativo alla relazione "Visite mediche"; può avere come valori "Idoneo" o "Non Idoneo", a seconda del risultato del suddetto controllo medico;
- *stipendio*: vincolo relativo all'entità "Contratti dipendenti"; deve essere maggiore di zero e deve essere espresso con due cifre decimali dopo la virgola;
- *tipo*: vincolo relativo all'entità "Locali"; può essere "Garage", "Appartamento", "Magazzino", "Ufficio".

2.5 Traduzione verso il modello relazionale

2.5.1 Traduzione

Una fase fondamentale della progettazione di una base di dati è la traduzione che avviene tra due modelli di dati diversi: a partire da uno schema E/R, si costruisce uno schema logico equivalente, in grado, cioè, di rappresentare le stesse informazioni. Abbiamo affrontato questa fase tenendo in considerazione le seguenti regole generali, su cui si basa la metodologia di traduzione.

- *Associazioni uno-a-molti*. Partendo dalla relazione "Partecipazione" tra le entità "Lavori" e "Clienti", possiamo pensare di tradurre tutto con tre tabelle diverse; utilizzando questo approccio, notiamo che l'eventuale chiave della relazione "Partecipazione" è costituita solo dall'identificatore dell'entità "Lavori" perché le cardinalità dell'associazione ci dicono che ogni lavoro è associato a uno e un solo cliente. A questo punto le relazioni "Lavori" e "Partecipazione" hanno la stessa chiave ed è allora possibile fonderle in un'unica relazione di questo tipo: Lavori(denominazione, ..., cliente).

- Associazioni uno-a-uno. In questo contesto, la traduzione può essere fatta rappresentando l'associazione in una qualunque delle relazioni che rappresentano le due entità. Si è scelto di accorpare la chiave dell'entità "Dipendente" dentro l'entità "Contratti dipendenti" che, in questo modo, diventerà, dopo la traduzione, una tabella del tipo: Contratti dipendenti(dipendente, ...).

Applicando questi concetti a tutto lo schema E/R di Figura 2.1, siamo riusciti a completare la fase di traduzione, della quale mostriamo il risultato nella Tabella 2.1.

Entità/Relazione	Traduzione
Bilanci	Bilanci(<u>anno</u> , entrate, uscite, saldo complessivo)
Clienti	Clienti(<u>codice fiscale</u> , nome, cognome, residenza, data nascita, email, numero di telefono)
Contratti dipendenti	Contratti dipendenti(<u>dipendente</u> , stipendio, ruolo, data stipula, durata, orario, extra)
Corsi di sicurezza	Corsi di sicurezza(<u>id</u> , materia trattata, data, ora inizio, ora fine, attività svolta, durata, docente)
Dipendenti	Dipendenti(<u>matricola</u> , nome, cognome, codice fiscale, residenza, data nascita, email, numero di telefono)
Entrate	Entrate(<u>id</u> , data, importo, provenienza, dettagli, causale)
Fornitori	Fornitori(<u>nome</u> , titolare, attività, residenza, email, numero di telefono)
Lavori	Lavori(<u>denominazione</u> , importo, inizio previsto, inizio effettivo, fine prevista, fine effettiva, foglio, particella, sub, classe, categoria, consistenza, pagato, dettagli pagamento, cliente)
Locali	Locali(<u>indirizzo</u> , tipo, data acquisto, dettagli)
Materie prime	Materie prime(<u>nome</u> , prezzo)
Materie prime in lista spesa	Materie prime in lista spesa(<u>nome</u> , quantità, prezzo unitario, prezzo complessivo)
Materie prime in magazzino	Materie prime in magazzino(<u>nome</u> , quantità)
Preventivi	Preventivi(<u>id</u> , descrizione, prezzo complessivo, cliente, confermato, data)
Uscite	Uscite(<u>id</u> , data, importo, destinazione, dettagli, causale)
Veicoli	Veicoli(<u>targa</u> , classificazione, dettagli, locale, data acquisto, costo)
Visite mediche	Visite mediche(<u>id</u> , data, risultato, dettagli, dipendente)

Tabella 2.1: Traduzione verso il modello relazionale

2.5.2 Vincoli di riferimento

I vincoli di riferimento, detti anche vincoli di integrità referenziale o, in inglese, foreign key) sono i vincoli inter-relazionali (coinvolgono, cioè, più relazioni) di maggiore interesse. La definizione di questo tipo di vincolo è abbastanza complicata; nel nostro contesto, per comprendere che tipo di vincolo la base di dati è chiamata a rispettare, ci basta analizzare uno degli esempi presentati nella Tabella 2.2, presentata come supporto allo schema logico precedentemente ottenuto.

Nella fase precedente abbiamo tradotto le entità "Lavori", "Clienti" e "Partecipazione" nelle tabelle Lavori(denominazione, ..., cliente) e Clienti(codice fiscale, ...). Il vincolo di integrità referenziale dice che il valore dell'attributo "Cliente" in ogni istanza di "Lavoro" deve essere presente anche tra i valori dell'attributo "Codice fiscale" per ogni istanza di "Cliente"; se questo non dovesse accadere, il vincolo di integrità verrebbe violato.

Traduzione	Vincoli di riferimento
Contratti dipendenti(<u>dipendente</u> , stipendio, ruolo, data stipula, durata, orario, extra)	Dipendente → Dipendenti.matricola
Lavori(<u>denominazione</u> , importo, ditta esecutrice, inizio previsto, inizio effettivo, fine prevista, fine effettiva, foglio, particella, sub, classe, categoria, consistenza, pagato, dettagli pagamento, cliente)	Cliente → Clienti.codice_fiscale
Preventivi(<u>id</u> , descrizione, prezzo complessivo, cliente, confermato)	Cliente → Clienti.codice_fiscale
Veicoli(<u>targa</u> , classificazione, dettagli, locale)	Locale → Locali.indirizzo
Visite mediche(<u>id</u> , data, risultato, dettagli, dipendente)	Dipendente → Dipendenti.matricola

Tabella 2.2: Vincoli di riferimento

Progettazione della componente applicativa

Nel terzo capitolo passeremo in rassegna la parte relativa alla progettazione della componente applicativa. In un primo momento ci occuperemo della modellazione da un punto di vista strutturale, quindi organizzazione del sistema e struttura dei dati elaborati; utilizzeremo, in questa sezione, il diagramma dell'architettura e il diagramma delle classi. Nella seconda parte analizzeremo il sistema da una prospettiva comportamentale, cioè, vedremo come si comporta dinamicamente in risposta a degli eventi; i mezzi di cui faremo uso in questa parte sono i diagrammi di attività e i diagrammi di sequenza.

3.1 Diagramma dell'architettura

In questa prima fase della progettazione, ci siamo occupati dell'organizzazione architettonica del sistema e della progettazione della sua struttura. Si tratta del collegamento critico tra progettazione e ingegneria dei requisiti, in quanto identifica i principali componenti strutturali del sistema e le loro relazioni. L'output ottenuto da questo processo è il modello architettonico mostrato in Figura 3.1; esso descrive come il sistema è organizzato in funzione dei componenti. L'architettura di un sistema software può basarsi su un particolare pattern o stile architettonico (questi termini hanno assunto lo stesso significato). Un pattern architettonico è una descrizione dell'organizzazione del sistema, per esempio un'organizzazione client-server o un'architettura a strati. I pattern architettonici esprimono l'essenza dell'architettura che è stata utilizzata in diversi sistemi software. Quando si prendono decisioni sull'architettura di un sistema, è importante conoscere i pattern architettonici più comuni, le loro applicazioni, i loro punti di forza e le loro debolezze. Nel nostro sistema abbiamo utilizzato il pattern MVC (Model-View-Controller) in cui la presentazione e l'interazione sono separate dai dati del sistema; quest'ultimo è strutturato in tre componenti logiche che interagiscono tra di loro:

- *model*: gestisce i dati del sistema e le operazioni associate;
- *view*: definisce e gestisce il modo in cui i dati sono presentati all'utente;
- *controller*: gestisce le interazioni degli utenti passandole alle altre due componenti.

Il vantaggio principale di questo pattern è che consente ai dati di cambiare indipendentemente dalla loro rappresentazione, e viceversa; esso supporta la presentazione degli stessi dati in modi differenti, con le modifiche fatte in una rappresentazione che vengono estese in tutte le altre.

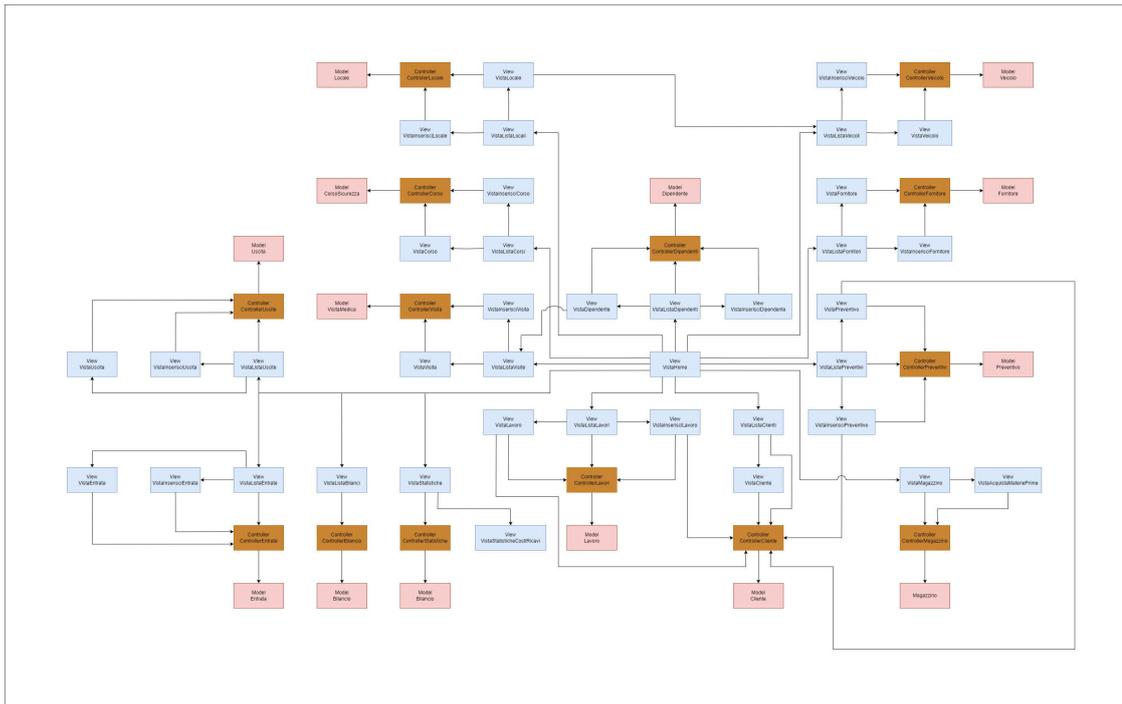


Figura 3.1: Mappa dell'architettura del sistema

Nella Figura 3.2 mostriamo le componenti logiche relative alle sezioni "Dipendenti", "Visite mediche", "Corsi di sicurezza", "Veicoli", "Locali" e "Fornitori".

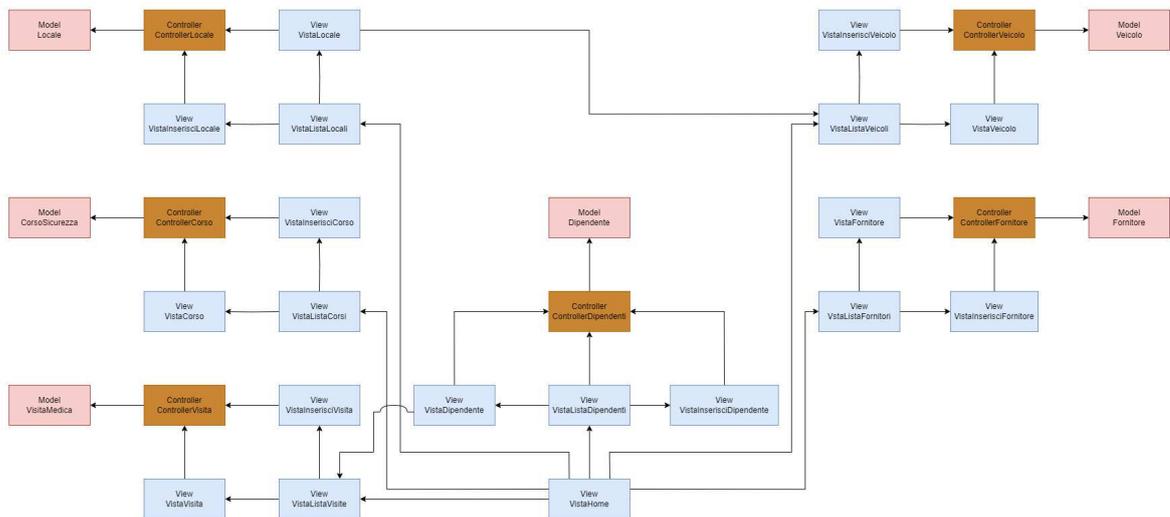


Figura 3.2: Mappa dell'architettura del sistema (parte 1)

Nella Figura 3.3 mostriamo le componenti logiche relative alle sezioni "Lavori", "Preventivi", "Clienti" e "Materie prime".

Nella Figura 3.4 mostriamo le componenti logiche delle sezioni "Entrate", "Uscite", "Bilanci" e "Statistiche".

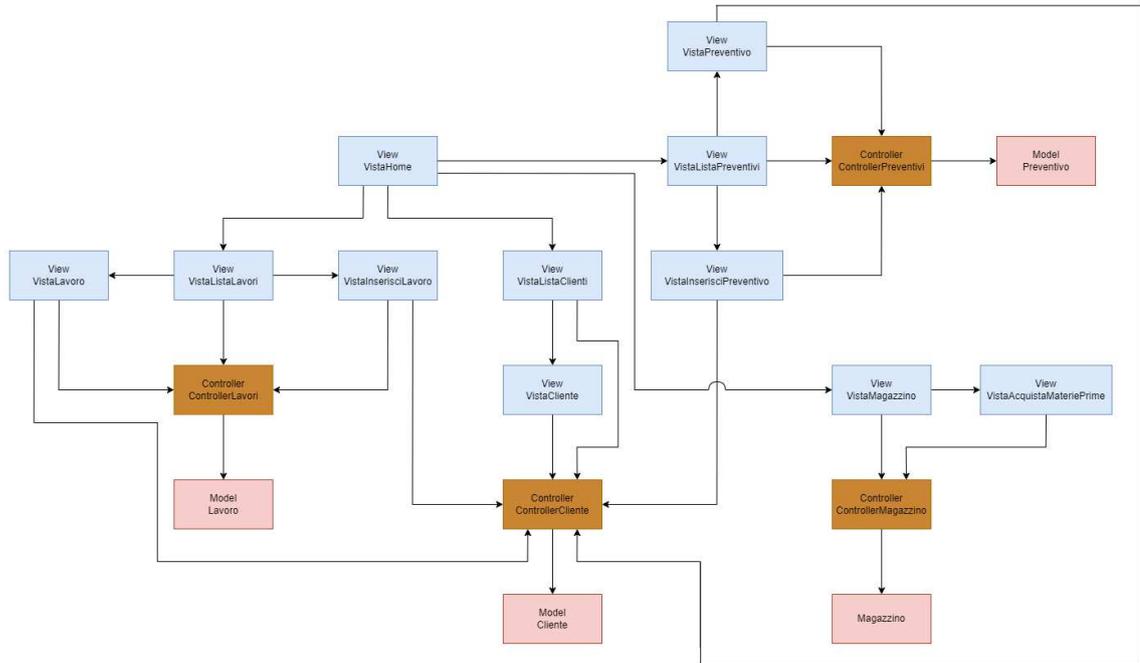


Figura 3.3: Mappa dell'architettura del sistema (parte 2)

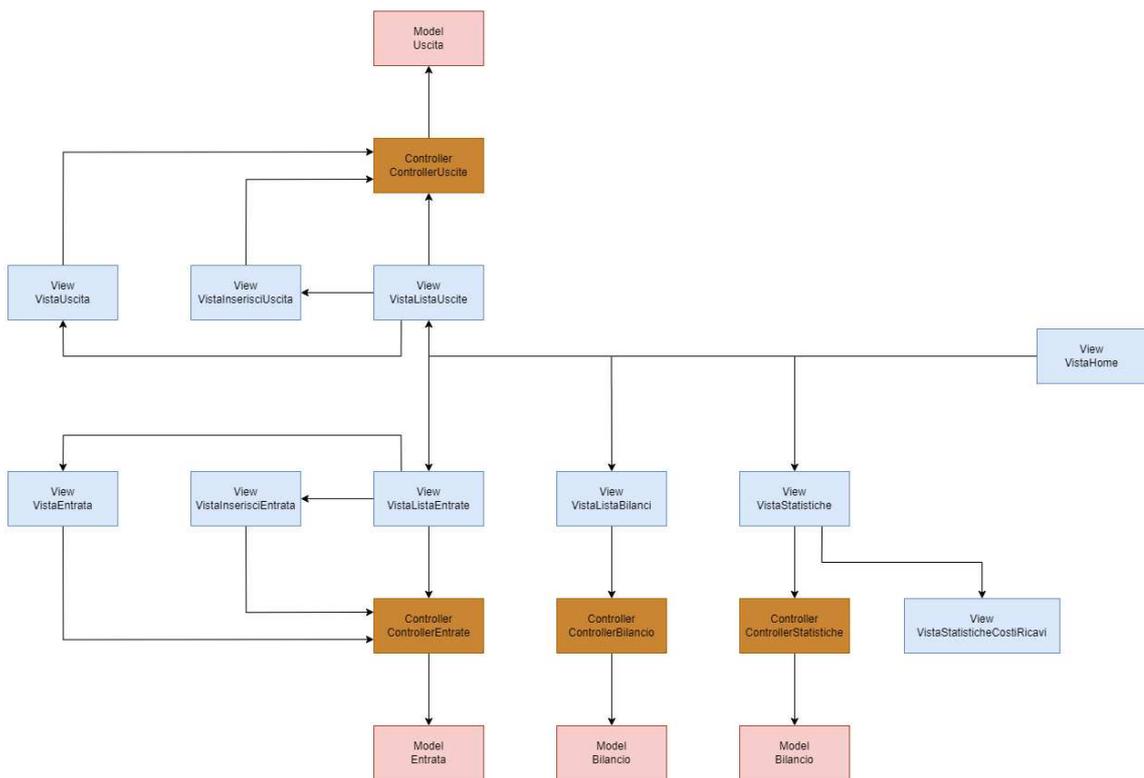


Figura 3.4: Mappa dell'architettura del sistema (parte 3)

3.2 Diagramma delle classi

Nella seconda fase, ci siamo occupati di ricavare un modello strutturale, ovvero che descrive la struttura statica del sistema tramite le classi di oggetti e le loro relazioni. Nello specifico, si tratta del diagramma delle classi, mostrato in Figura 3.5. Questo tipo di approccio è la naturale conseguenza della scelta di effettuare una progettazione orientata agli oggetti, un tipo di progettazione sviluppatasi negli anni '80. Nel corso degli anni sono state elaborate diverse tecniche di identificazione delle classi di oggetti per i sistemi orientati agli oggetti. In pratica, occorre servirsi di varie conoscenze per scoprire le classi di oggetti con i rispettivi attributi e le rispettive operazioni; molte di queste informazioni, nel nostro contesto, sono state derivate dall'analisi dei requisiti e dalla divisione in settori di applicazione dell'intero sistema.

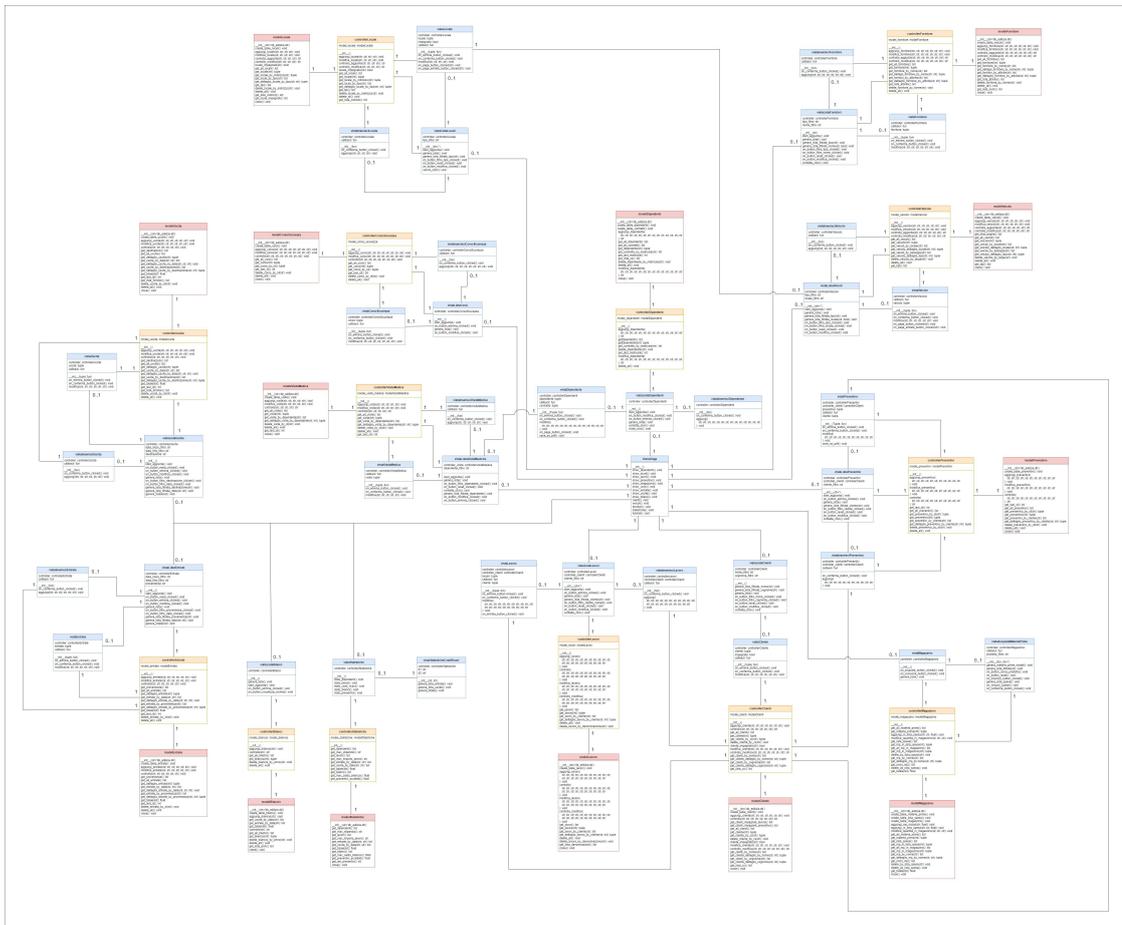


Figura 3.5: Diagramma delle classi del sistema

Nella Figura 3.6 mostriamo le classi relative alle sezioni "Veicoli", "Locali" e "Fornitori".

Nella Figura 3.7 mostriamo le classi relative alle sezioni "Dipendenti", "Visite mediche", "Corsi di sicurezza".

Nella Figura 3.8 mostriamo le classi delle sezioni "Entrate", "Uscite", "Bilanci" e "Statistiche".

Nella Figura 3.9 mostriamo le classi delle sezioni "Lavori", "Preventivi" e "Clienti".

Nella Figura 3.10 mostriamo le classi della sezione "Materie prime".

Mostriamo di seguito una divisione in settori del diagramma delle classi; la suddivisione viene svolta sulla base del lavoro effettuato durante la fase di analisi dei requisiti.

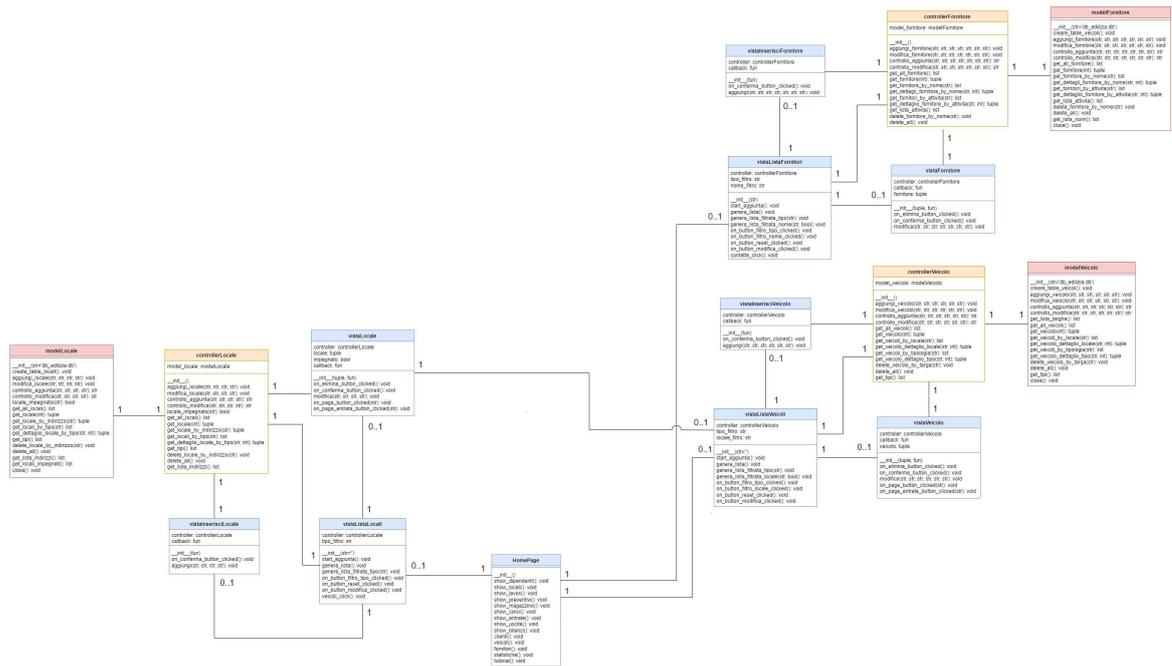


Figura 3.6: Diagramma delle classi del sistema (parte 1)

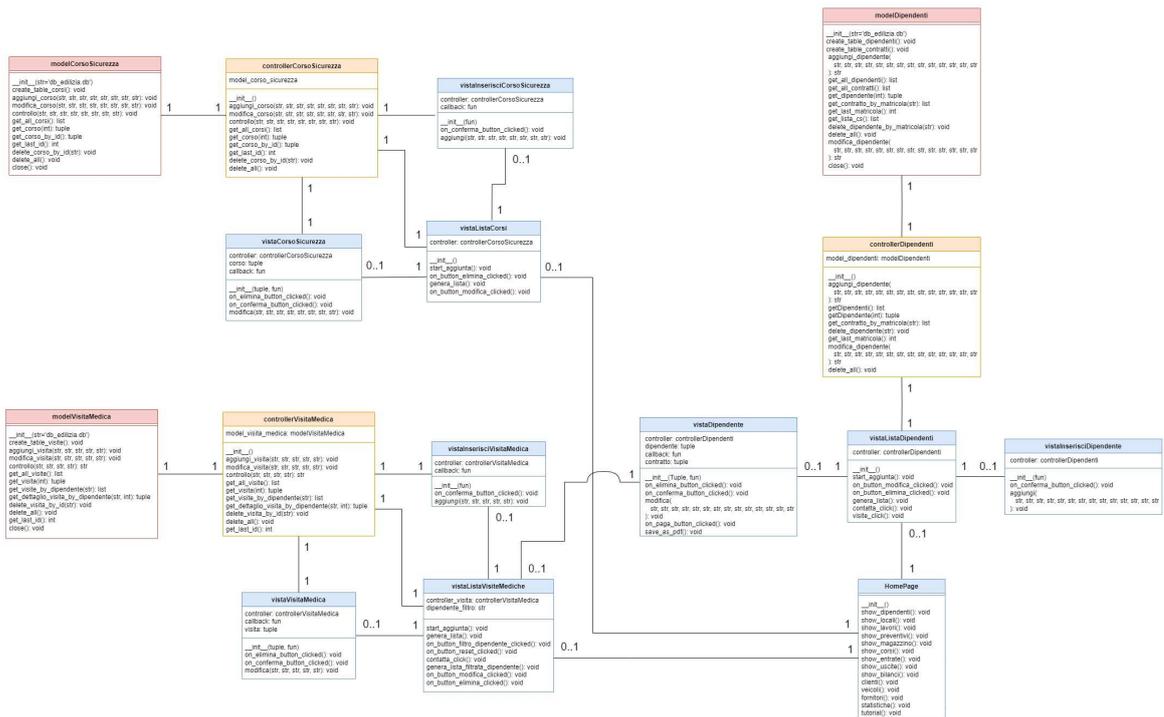


Figura 3.7: Diagramma delle classi del sistema (parte 2)

3.2.1 Sezione "Lavori e Clienti"

In Figura 3.11 sono illustrate le classi relative alla sezione "Lavori e Clienti", nello specifico alle sottosezioni "Lavori" e "Clienti". Nella Figura 3.12 mostriamo anche la parte relativa alle classi che gestiscono la sottosezione "Preventivi", che appartiene alla sezione "Lavori e Clienti".

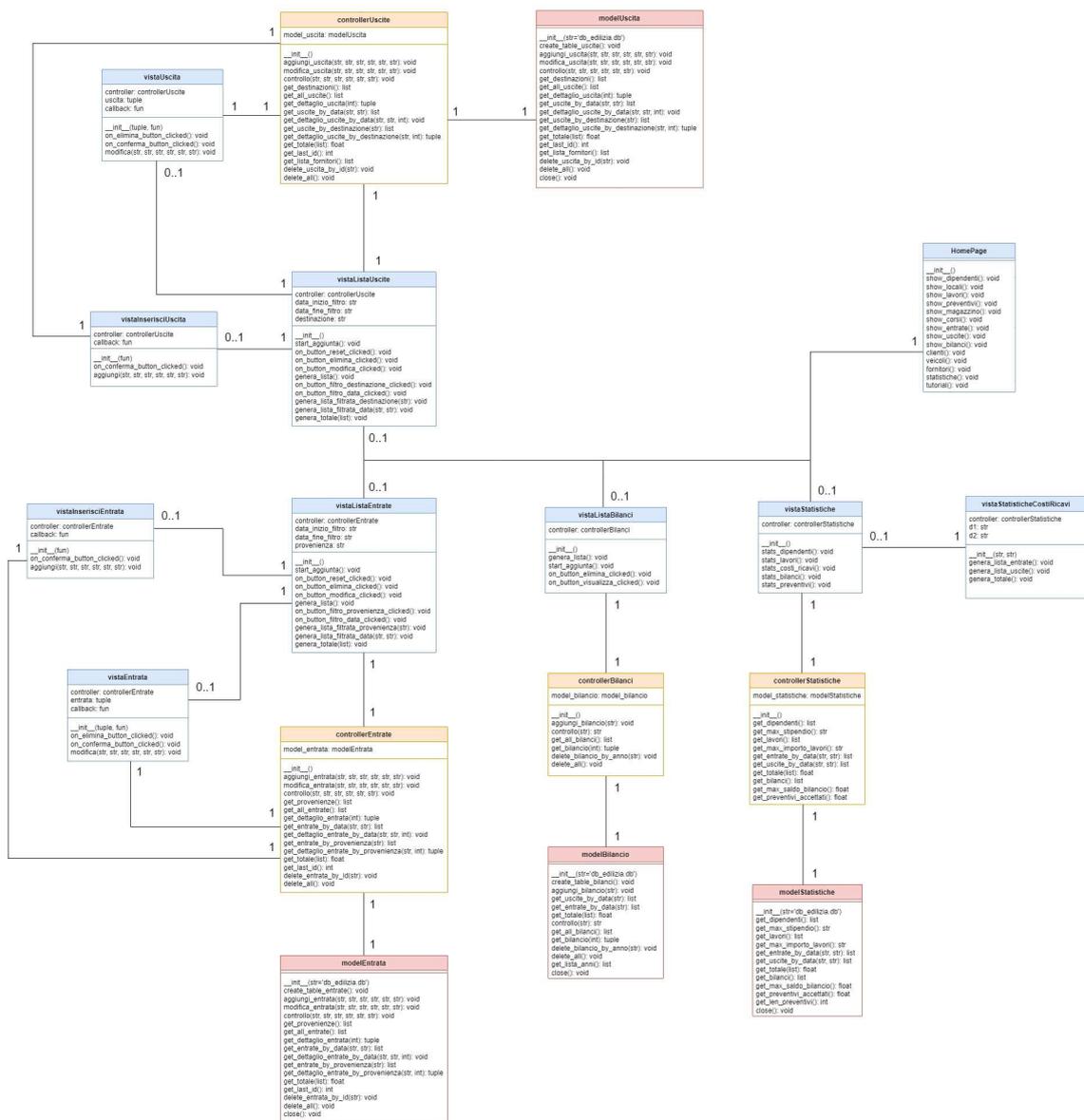


Figura 3.8: Diagramma delle classi del sistema (parte 3)

3.2.2 Sezione "Dipendenti"

In Figura 3.13 sono illustrate le classi relative alla sezione "Dipendenti".

3.2.3 Sezione "Sicurezza sul lavoro"

In Figura 3.14 sono illustrate le classi relative alla sezione "Sicurezza sul lavoro", che comprende le sottosezioni "Corsi di Sicurezza" e "Visite Mediche".

3.2.4 Sezione "Magazzino"

In Figura 3.15 sono illustrate le classi relative alla sezione "Magazzino", nello specifico alla sottosezione "Materie Prime". Mostriamo, anche, nella Figura 3.16 le parti relative alle

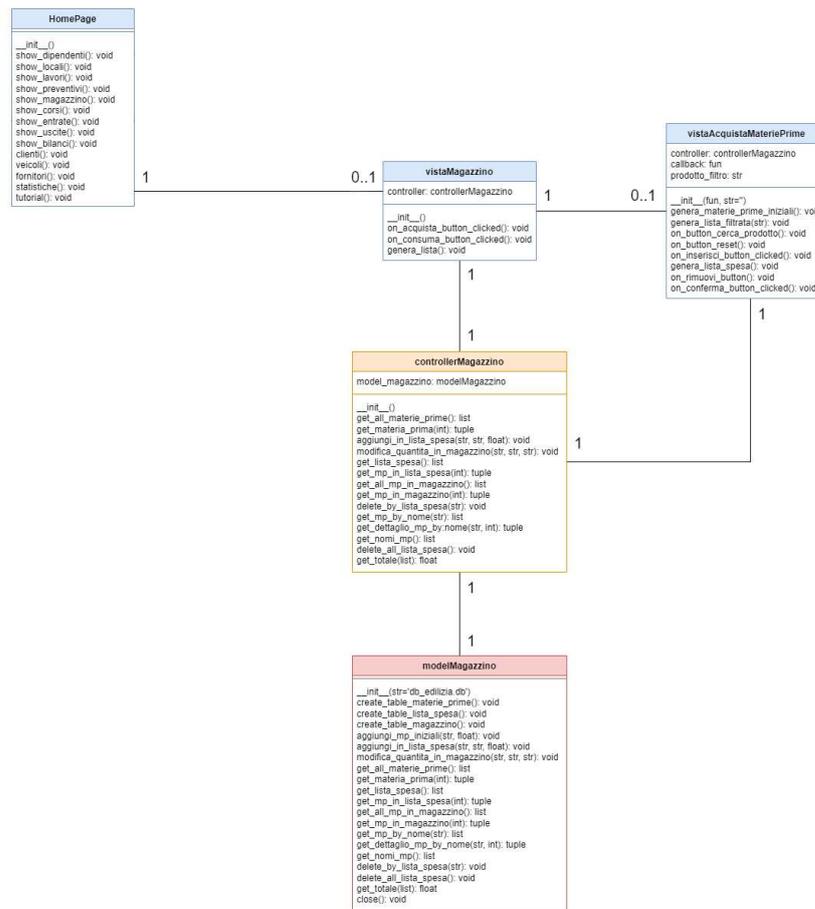


Figura 3.10: Diagramma delle classi del sistema (parte 5)

Anche in questa sezione, riprendendo quanto fatto nei capitoli precedenti, utilizzeremo una suddivisione secondo i campi di applicazione del sistema.

3.3.1 Sezione "Lavori e Clienti"

Nella Figura 3.21 sono illustrati i diagrammi delle attività dei casi d'uso riguardanti la sezione "Lavori e Clienti".

3.3.2 Sezione "Dipendenti"

Nella Figura 3.22 sono illustrati i diagrammi delle attività dei casi d'uso riguardanti la sezione "Dipendenti".

3.3.3 Sezione "Sicurezza sul lavoro"

Nelle Figure 3.23 e 3.24 sono illustrati, rispettivamente, i diagrammi delle attività dei casi d'uso riguardanti le sottosezioni "Corsi di Sicurezza" e "Visite Mediche".

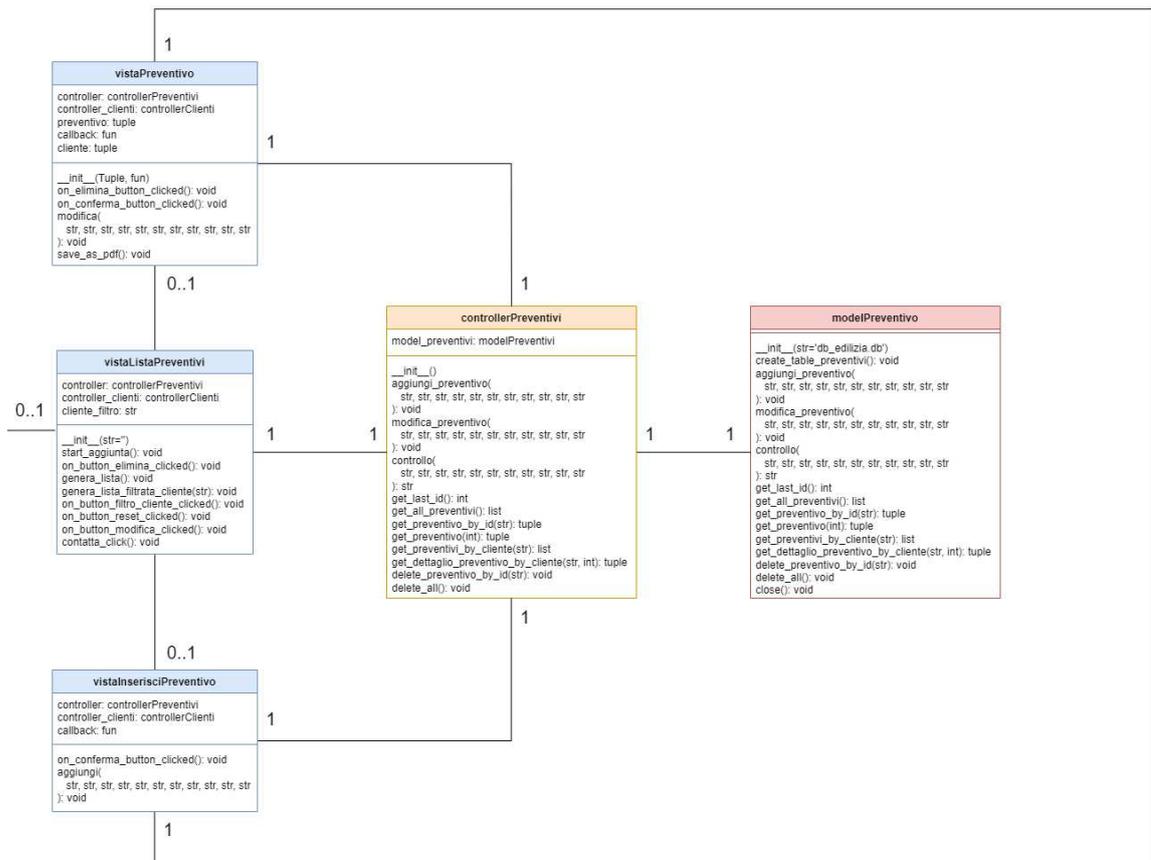


Figura 3.12: Diagramma delle classi relativo alla sottosezione "Preventivi"

3.3.7 Sezione "Statistiche"

Nella Figura 3.30 sono illustrati i diagrammi delle attività dei casi d'uso riguardanti la sezione "Statistiche".

3.3.8 Filtraggio

In quest'ultima sezione analizziamo il flusso di attività - comune - relativo a tutti i casi d'uso che riguardano il filtraggio. Il diagramma in Figura 3.31 è una generalizzazione (in quanto l'operazione di filtraggio è sostanzialmente uguale dappertutto) e non riguarda nessuna sezione in particolare; tuttavia ci aiuta a comprendere bene ciò che accade anche nello specifico.

3.4 Diagrammi di sequenza

Terminiamo la parte relativa alla progettazione dal punto di vista comportamentale con i diagrammi di sequenza, utilizzati per modellare le interazioni tra gli attori - in questo caso uno solo, l'utente - e gli oggetti del sistema e le interazioni tra gli stessi oggetti. Suddivideremo i diagrammi di sequenza in cinque sezioni: nelle prime quattro mostreremo la sequenza delle interazioni che si svolgono nelle operazioni CRUD (Create, Read, Update, e Delete); nell'ultima, ci occuperemo di casi d'uso che riguardano altri tipi di operazioni.

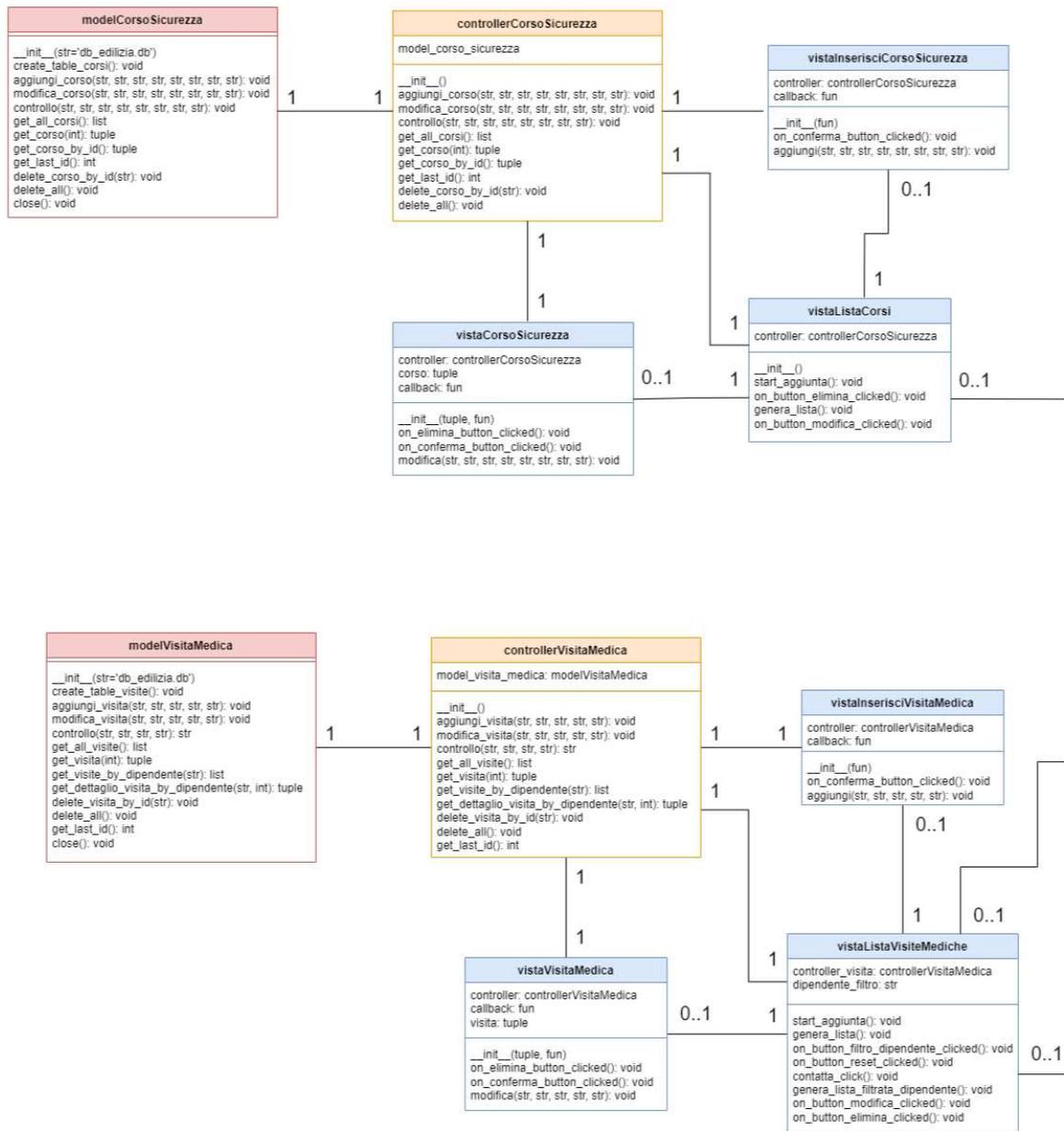


Figura 3.14: Diagramma delle classi relativo alla sezione "Sicurezza"

vistaListaFornitori, attraverso il click su un pulsante e la conseguente chiamata della funzione `start_aggiunta()`; successivamente, dopo l’inserimento dei dati da parte dell’utente, il sistema effettua un controllo e, se l’esito è positivo, completa l’operazione con successo.

Questo diagramma è stato illustrato per il caso d’uso "Inserisci fornitore"; tuttavia può essere applicato, con le opportune modifiche ai nomi degli oggetti in gioco e delle funzioni, anche ad altre operazioni di inserimento semplici.

Inserisci lavoro

Nella Figura 3.34 è illustrato il diagramma di sequenza relativo al caso d’uso "Inserisci lavoro". In questo caso ci sono delle modifiche rispetto ai diagrammi precedenti; infatti, dopo

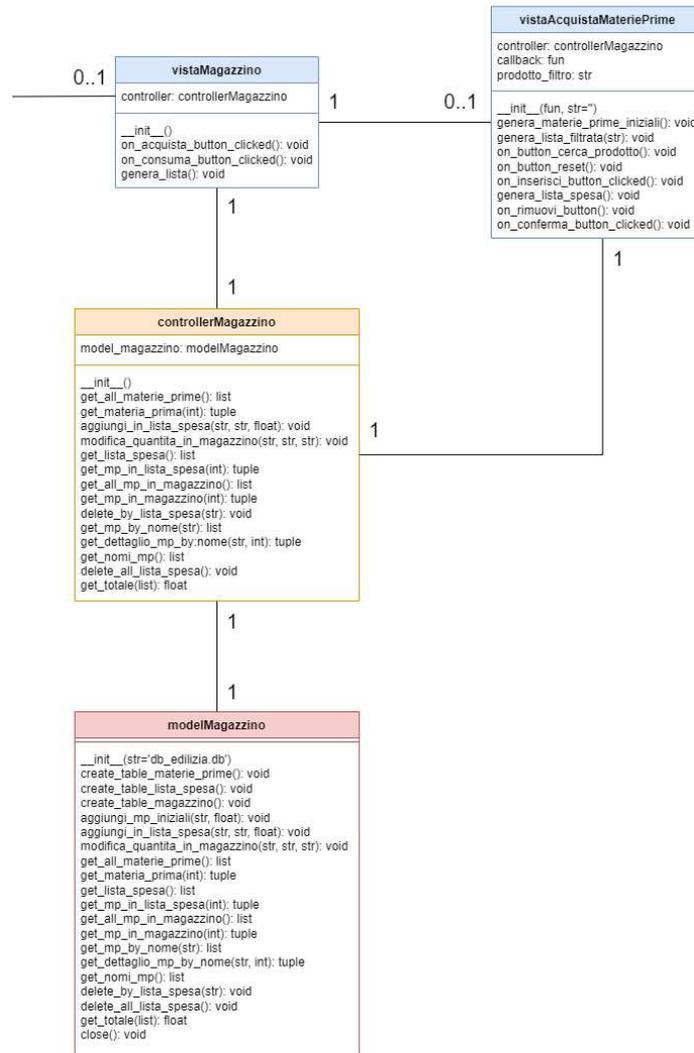


Figura 3.15: Diagramma delle classi relativo alla sottosezione "Materie Prime"

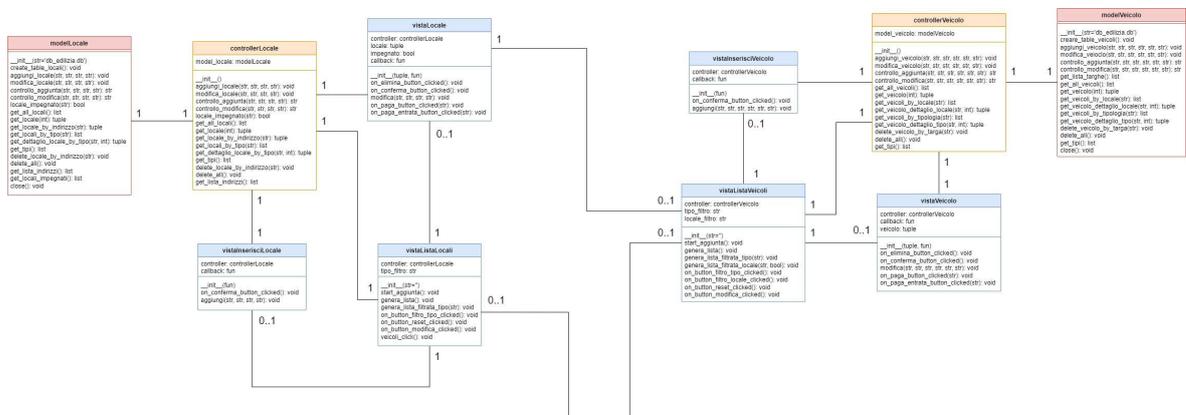


Figura 3.16: Diagramma delle classi relativo alle sottosezioni "Locali" e "Veicoli"

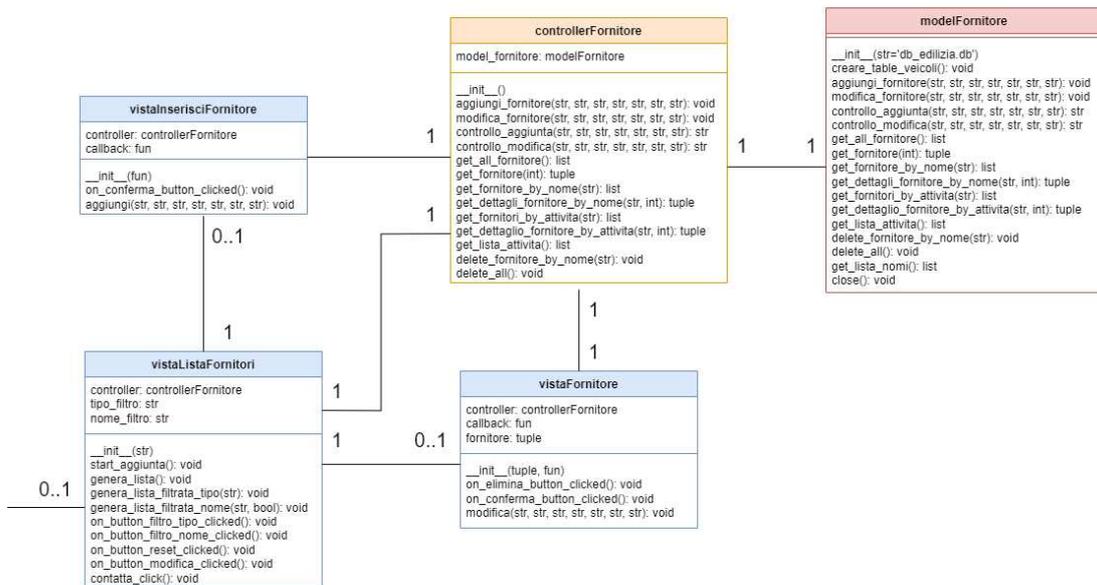


Figura 3.17: Diagramma delle classi relativo alla sezione "Fornitori"

un primo controllo sui dati del lavoro, il sistema, attraverso le classi relative alla sezione "Clienti", effettua un secondo controllo anche sui dati del rispettivo cliente (ricordiamo che l'inserimento del cliente deve essere fatto in contemporanea all'inserimento del lavoro, ed è necessario controllare che il cliente non esista già); a questo punto, se sia i dati del cliente, che quelli del lavoro sono corretti, il sistema procede prima con l'operazione di inserimento del lavoro e poi, se il cliente non esiste, anche con l'operazione di inserimento di quest'ultimo.

Possiamo applicare lo stesso diagramma, con le opportune modifiche, al caso d'uso "Inserisci preventivo".

Questo diagramma si può applicare anche al caso d'uso "Inserisci dipendente"; infatti, anche in questo caso, avvengono, in realtà, due operazioni di inserimento in contemporanea, ovvero l'inserimento dei dati anagrafici del dipendente e l'inserimento dei dati relativi al rispettivo contratto; di conseguenza, va effettuato un controllo sulla correttezza in entrambi i casi. L'unica differenza è questa: se sia il controllo sui dati anagrafici, che quello sui dati del contratto vanno a buon fine, le due operazioni vengono entrambe completate correttamente, a differenza del caso precedente, dove l'inserimento del cliente era vincolato al fatto che questo non doveva essere presente nel sistema; in questo caso, tale controllo non serve perché, attraverso un meccanismo di autoincremento della chiave interna, il dipendente viene inserito con la certezza di non violare nessun vincolo.

Per essere ancora più chiari, questa leggera differenza proviene dal fatto che un cliente potrebbe essere impegnato in più lavori (relazione di tipo uno-a-molti), mentre un dipendente ha uno e un solo contratto (e viceversa, quindi relazione uno-a-uno). Di conseguenza, nel momento in cui l'utente inserisce i dati del cliente, se questo è già presente, non deve essere sollevata nessuna eccezione (in quanto, evidentemente, questo cliente era già stato inserito per un lavoro precedente) e questo viene reso possibile dal fatto che l'operazione non venga svolta.

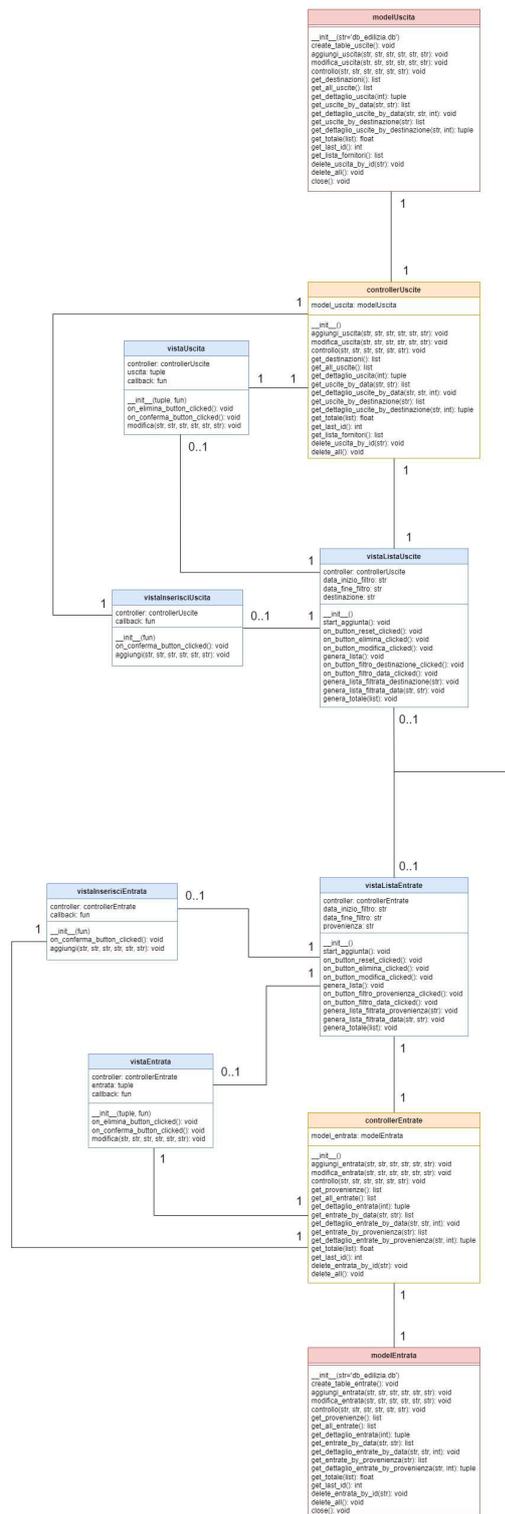


Figura 3.18: Diagramma delle classi relativo alle sottosezioni "Entrate" e "Uscite"

Inserisci veicolo

Nella Figura 3.35 è illustrato il diagramma di sequenza relativo al caso d'uso "Inserisci veicolo". Si tratta di un diagramma molto simile a quello mostrato in Figura 3.33, con,

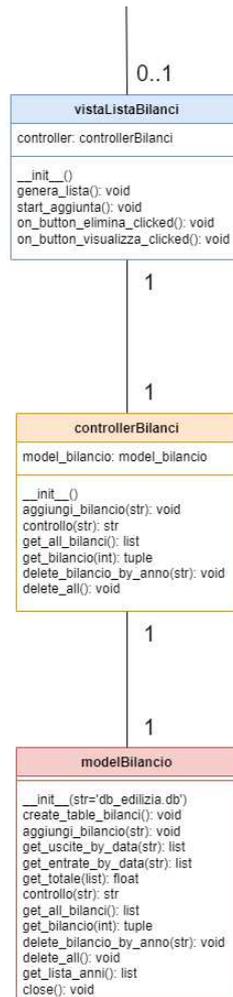


Figura 3.19: Diagramma delle classi relativo alla sottosezione "Bilanci"

però, una piccola differenza; prima di procedere con l'inserimento dei dati e del successivo controllo, viene fatta una verifica sulla presenza di locali; se questi sono presenti si può procedere alla fase successiva, altrimenti il sistema restituisce un messaggio di errore.

3.4.2 Read

Visualizza dettaglio

Nella Figura 3.36 è illustrato il diagramma di sequenza relativo al caso d'uso "Visualizza dettaglio cliente". La prima interazione è tra l'utente e la classe `vistaListaClienti`, attraverso il click di un pulsante che invoca la funzione `on_button_modifica_click()`. A questo punto, attraverso le classi relative alla sezione "Clienti", il sistema estrae i dati del cliente selezionato e li mostra sullo schermo attraverso la creazione di una nuova istanza (e, quindi, l'invocazione del suo costruttore) della classe `vistaCliente`.

Questo diagramma può essere applicato a qualsiasi altro caso d'uso che permette la visualizzazione di dettagli in altri settori del sistema.

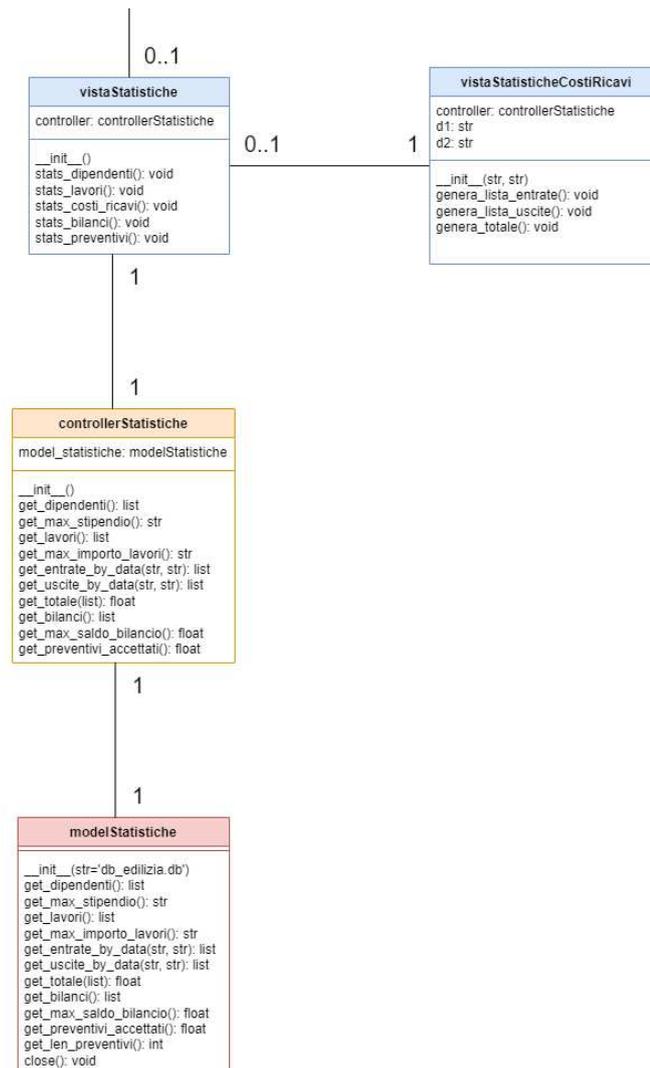


Figura 3.20: Diagramma delle classi relativo alla sezione "Statistiche"

Visualizza lista filtrata

Nella Figura 3.37 è illustrato il diagramma di sequenza relativo al caso d'uso "Filtra clienti per nome". L'utente, sempre attraverso il click di un pulsante, invoca la funzione `genera_lista_filtrata()`; successivamente il sistema, a partire dal nome selezionato, effettua il filtraggio, servendosi delle classi relative alla gestione dei clienti, e restituisce il risultato; viene, infine, restituita una lista all'utente con i clienti desiderati.

Questo diagramma può essere applicato anche a tutti gli altri casi d'uso che riguardano il filtraggio di una lista di oggetti qualsiasi, secondo un parametro qualsiasi.

Visualizza lista

Nella Figura 3.38 è illustrato il diagramma di sequenza relativo al caso d'uso "Visualizza lista clienti". L'utente invoca prima la funzione `clienti()` di `vistaHome`; successivamente, dopo il click su un pulsante, il sistema effettua la ricerca, attraverso metodi all'interno delle classi che gestiscono i clienti, e restituisce la lista desiderata.

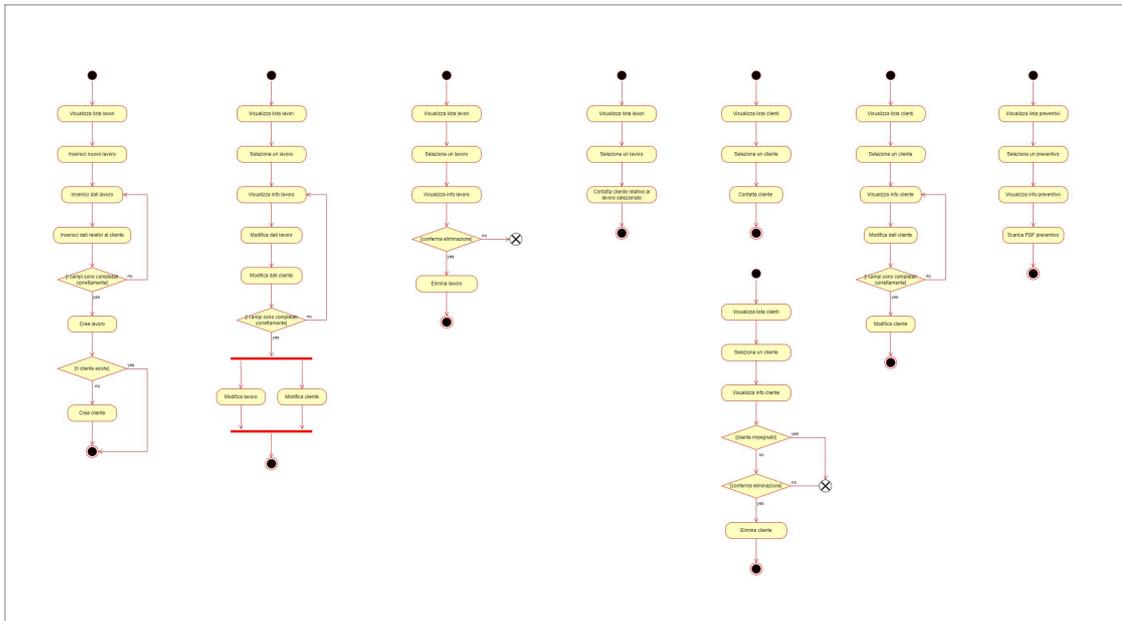


Figura 3.21: Diagrammi di attività relativi ai casi d’uso della sezione "Lavori e Clienti"

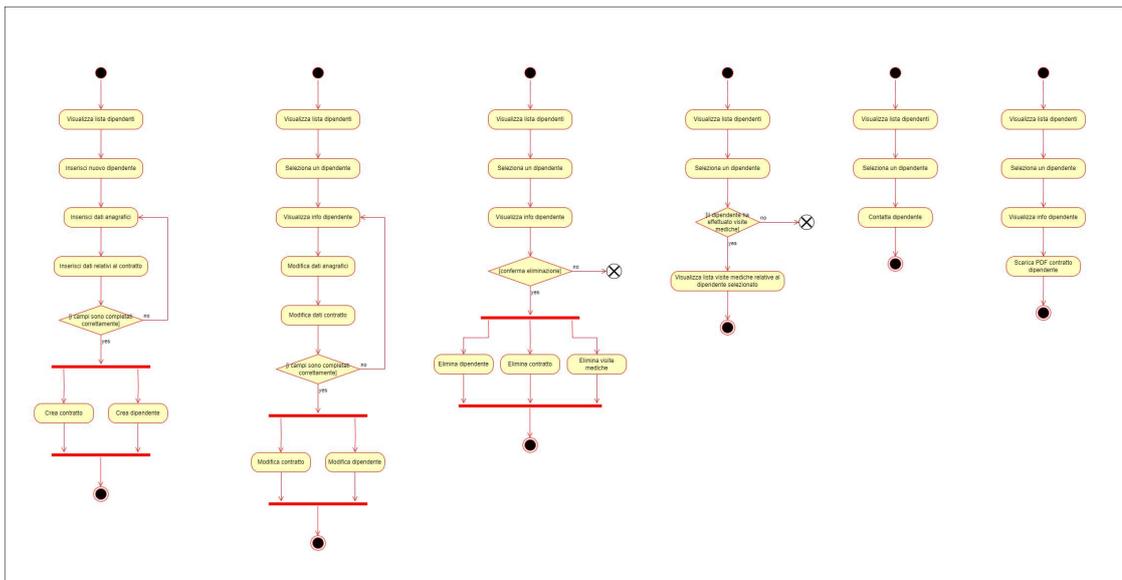


Figura 3.22: Diagrammi di attività relativi ai casi d’uso della sezione "Dipendenti"

Anche questo diagramma può essere applicato, con le opportune modifiche, a tutti i casi d’uso che riguardano la visualizzazione di una lista di oggetti in particolare. Quello che cambia è, ovviamente, la funzione di `vistaHome` che viene invocata, che cambia a seconda della categoria di oggetti che si vogliono visualizzare.

Visualizza veicoli a partire da un locale

Nella Figura 3.39 è illustrato il diagramma di sequenza relativo al caso d’uso "Visualizza lista veicoli a partire da un locale". La prima interazione è tra l’utente e la classe `vistaListaLocali` attraverso la funzione `veicoli_click()`, che corrisponde al click su un pulsante; in seguito, il sistema recupera il locale selezionato dall’utente e lo passa in input

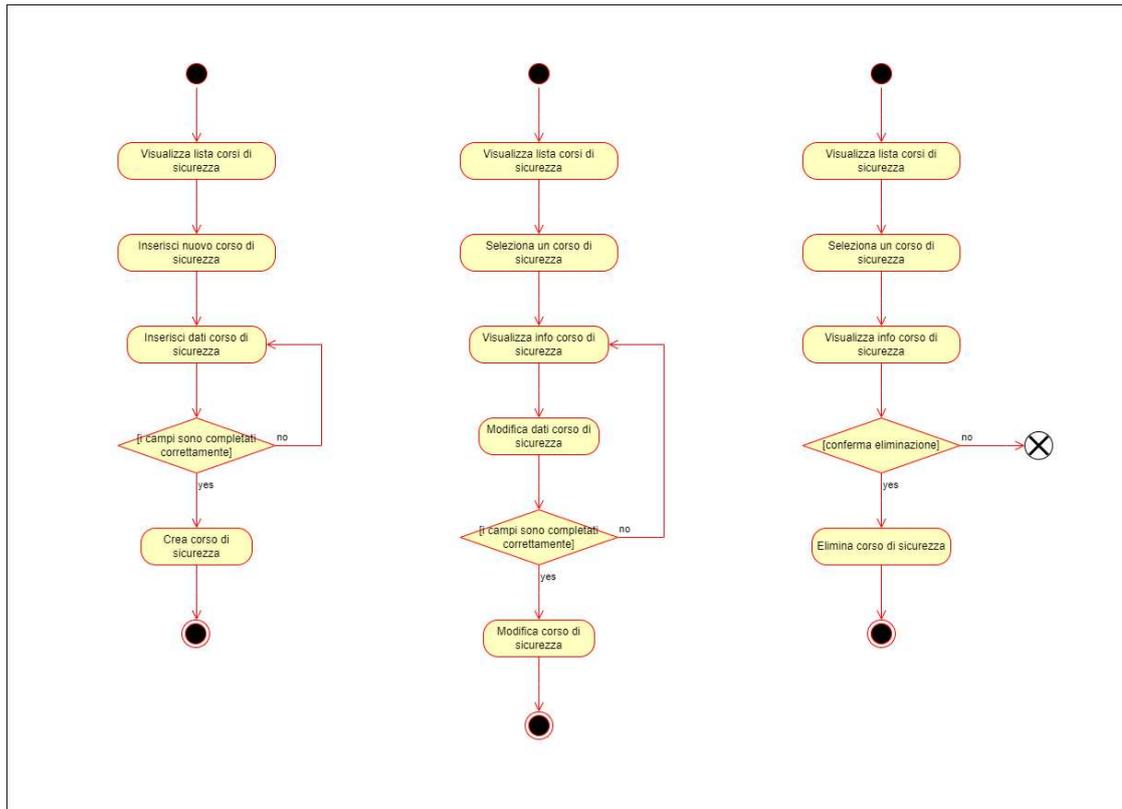


Figura 3.23: Diagrammi di attività relativi ai casi d'uso della sottosezione "Corsi di Sicurezza"

al costruttore della classe `vistaListaVeicoli`, istanziata per far sì che venga mostrata sullo schermo la lista dei veicoli contenuti nel locale selezionato.

Questo diagramma può essere applicato, con le opportune modifiche, anche al caso d'uso "Visualizza lista visite mediche a partire da un dipendente".

3.4.3 Update

Modifica lavoro

Nella Figura 3.40 è illustrato il diagramma di sequenza relativo al caso d'uso "Modifica lavoro". Dopo una fase iniziale in cui vengono visualizzati i dettagli del lavoro da modificare, l'utente, dopo aver cambiato i dati, invoca la funzione `modifica()` di `vistaLavoro` che, a sua volta, fa partire la classica serie di controlli necessari, prima di poter concludere correttamente l'operazione. Anche in questo caso, come nel caso d'uso "Inserisci lavoro", ci sono controlli anche sul cliente (che può essere modificato durante questa procedura).

Questo diagramma può essere applicato anche ai casi d'uso "Modifica preventivo" e "Modifica dipendente"; in questo secondo caso valgono ancora le considerazioni fatte nella sezione che riguarda l'inserimento.

Modifica fornitore

Nella Figura 3.41 è illustrato il diagramma di sequenza relativo al caso d'uso "Modifica fornitore". Si tratta di una versione semplificata del diagramma in Figura 3.40: in questo caso, dopo un'analogia parte in cui il sistema, a seguito del click dell'utente e della conseguente invocazione della funzione `onbutton_modifica_click()` di `vistaListaFornitori`, mostra sullo schermo la lista di tutti i fornitori e dopo che l'utente ne seleziona uno e ne

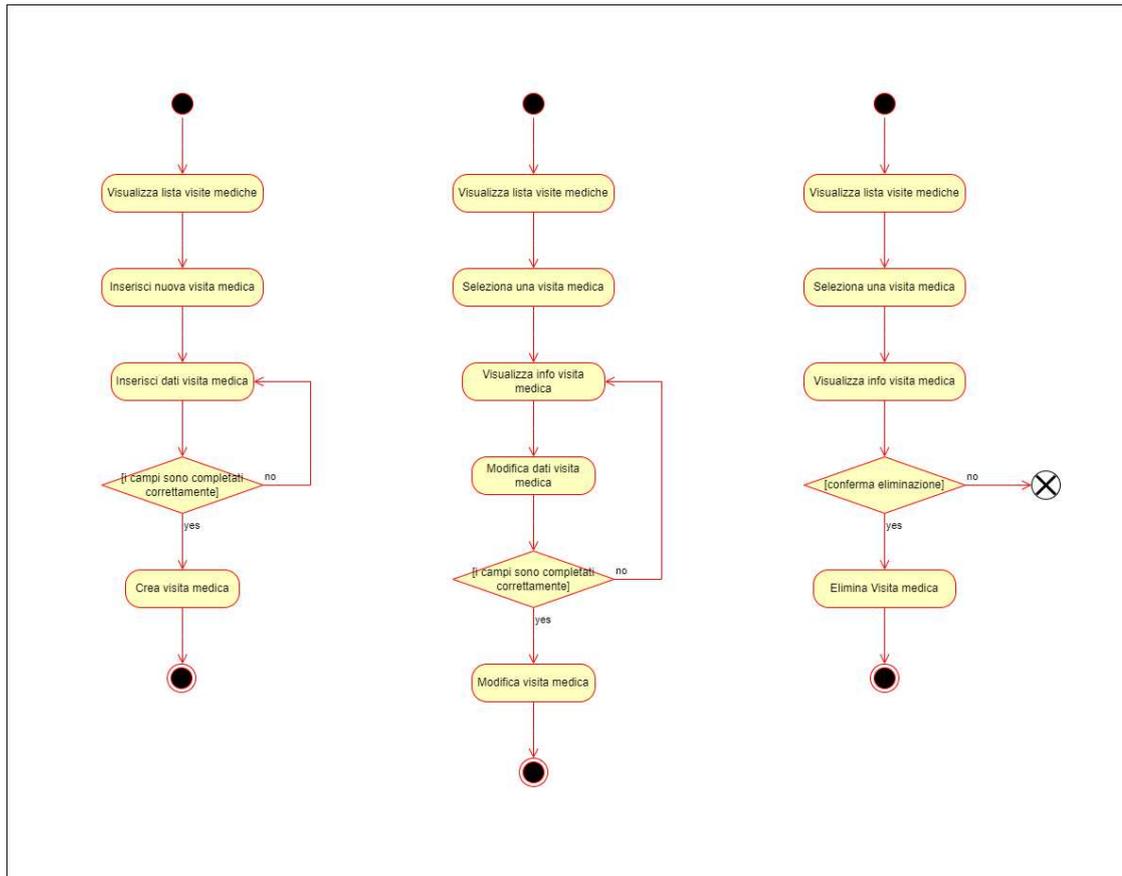


Figura 3.24: Diagrammi di attività relativi ai casi d'uso della sottosezione "Visite Mediche"

modifica i dati, viene effettuato un semplice controllo sulla completezza e correttezza dei dati e, se questo va a buon fine, viene completata l'operazione di modifica.

Questo diagramma può essere applicato anche a tutti gli altri casi d'uso che riguardano la modifica semplificata.

3.4.4 Delete

Elimina bilancio

Nella Figura 3.42 è illustrato il diagramma di sequenza relativo al caso d'uso "Elimina bilancio". L'operazione in questione consiste nell'invocazione, da parte dell'utente, della funzione `onbutton_elimina_click()` della classe `vistaListaBilanci` e nella ricerca, da parte del sistema, dell'anno del bilancio selezionato, che, alla fine, verrà dato in input alla funzione che gestisce l'eliminazione vera e propria.

Elimina cliente

Nella Figura 3.43 è illustrato il diagramma di sequenza relativo al caso d'uso "Elimina cliente". Nella prima parte l'utente visualizza sullo schermo i dettagli del cliente da eliminare; successivamente, all'invocazione della funzione `on_button_elimina_click()` della classe `vistaCliente`, il sistema effettua un controllo sul cliente selezionato, attraverso le classi che si occupano della gestione dei clienti; l'eliminazione vera e propria avviene soltanto se l'esito di questo controllo risulta andato a buon fine, ovvero se il cliente è libero da lavori/preventivi.

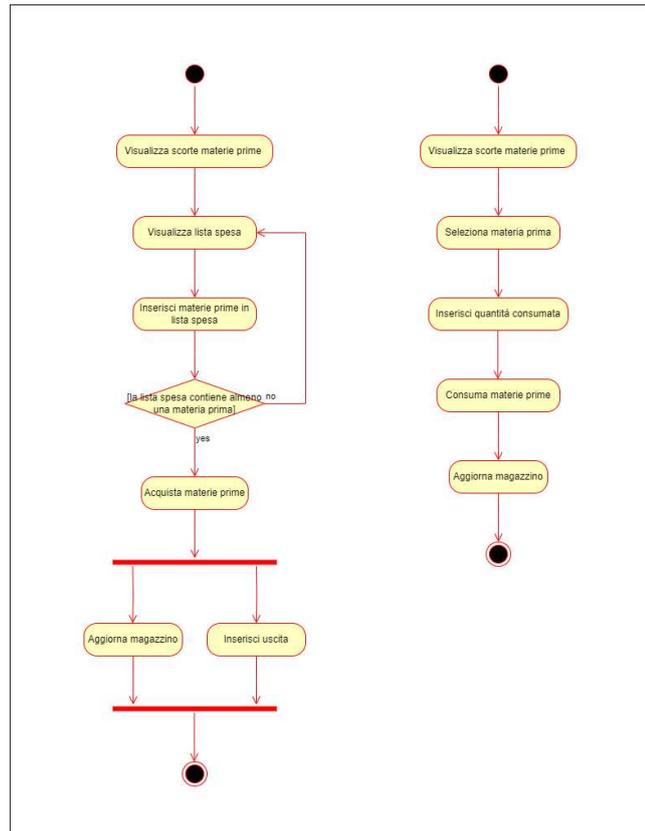


Figura 3.25: Diagrammi di attività relativi ai casi d’uso della sottosezione "Materie Prime"

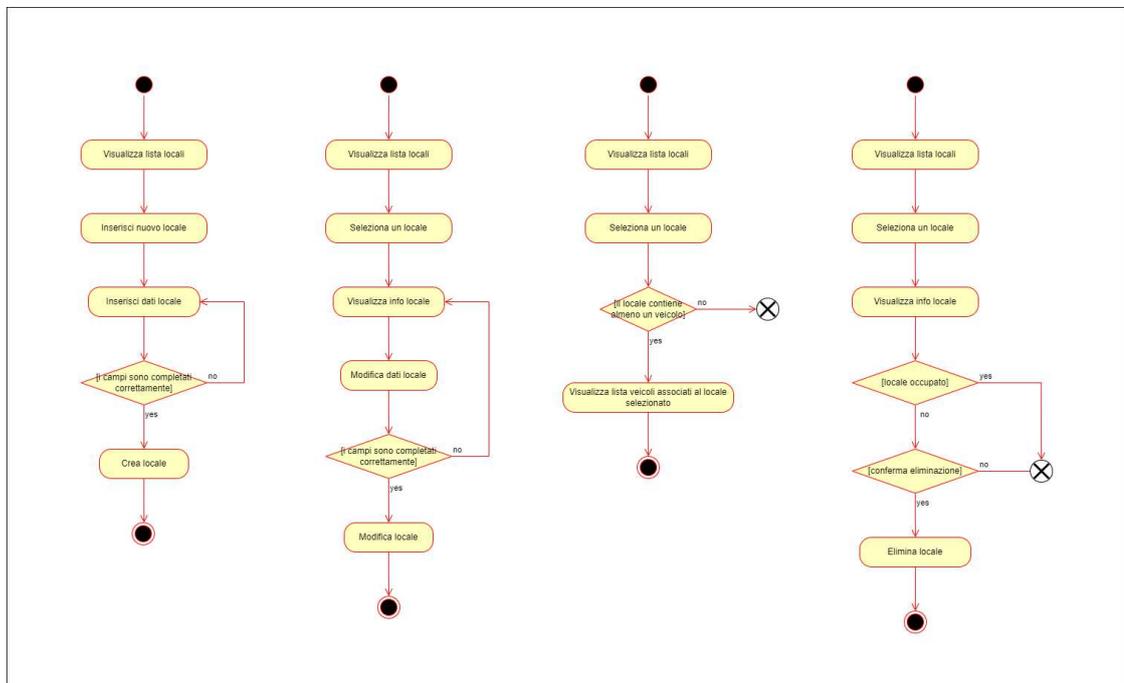


Figura 3.26: Diagrammi di attività relativi ai casi d’uso della sottosezione "Locali"

Elimina fornitore

Nella Figura 3.44 è illustrato il diagramma di sequenza relativo al caso d’uso "Elimina fornitore". Si tratta di un modello semplificato rispetto a quello precedente mostrato in

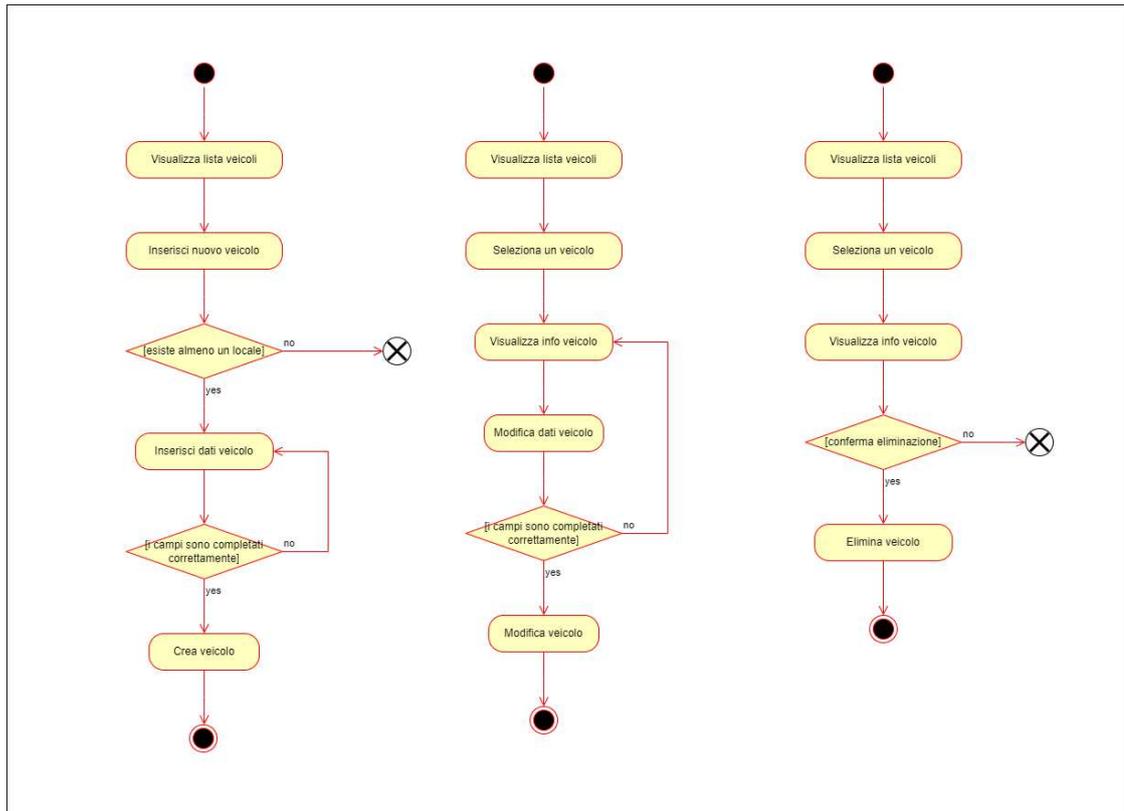


Figura 3.27: Diagrammi di attività relativi ai casi d'uso della sottosezione "Veicoli"

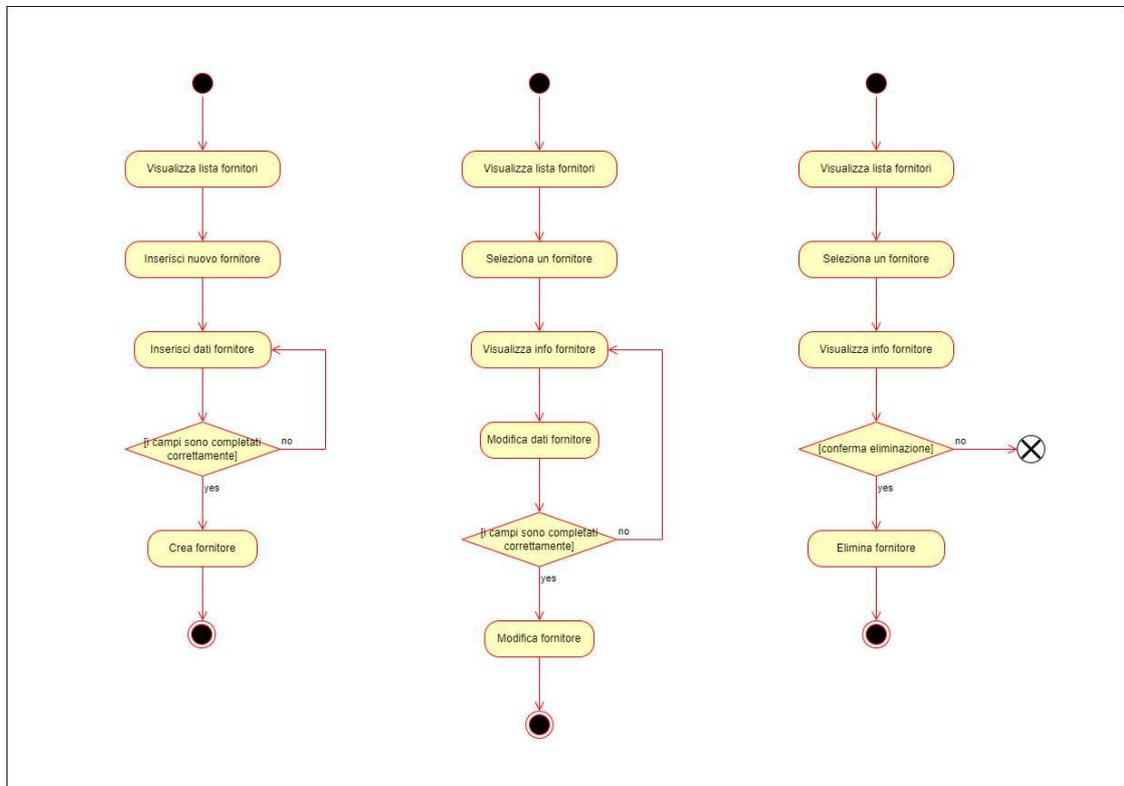


Figura 3.28: Diagrammi di attività relativi ai casi d'uso della sezione "Fornitori"

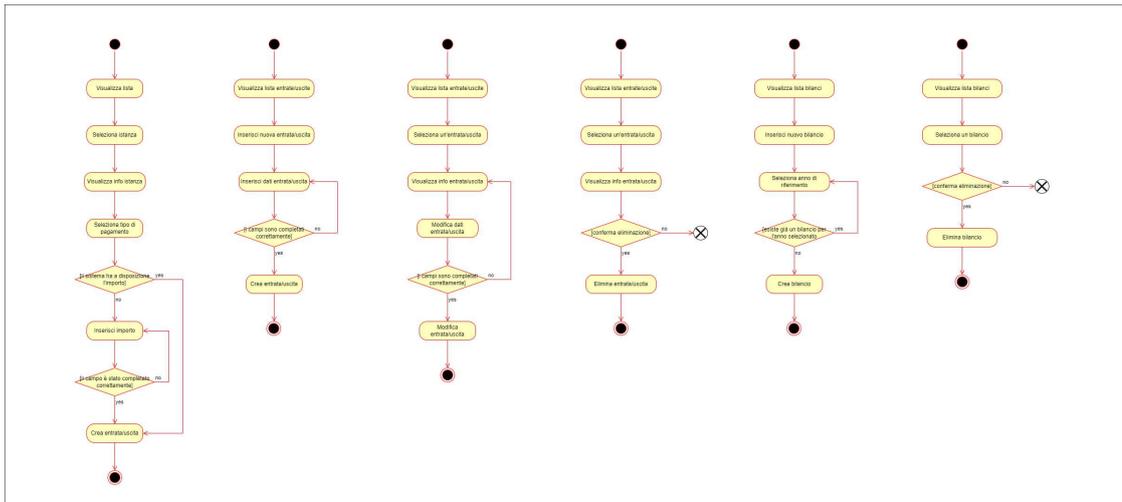


Figura 3.29: Diagrammi di attività relativi ai casi d'uso della sezione "Contabilità"

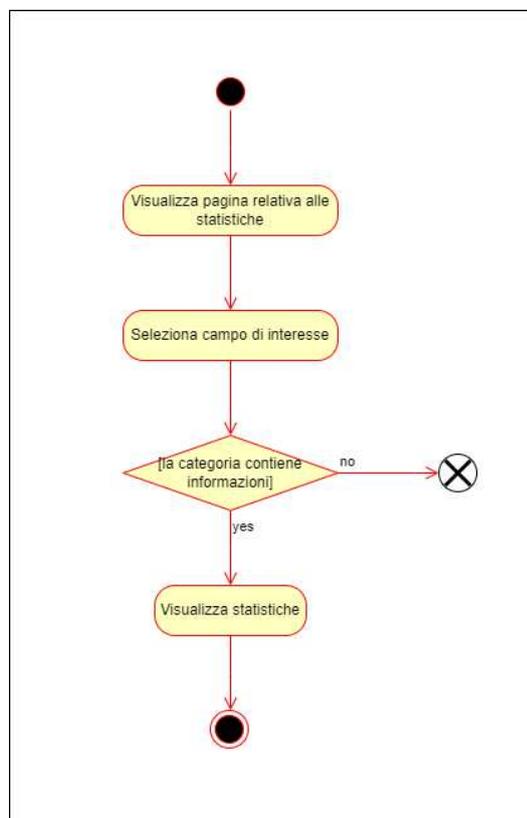


Figura 3.30: Diagrammi di attività relativi ai casi d'uso della sezione "Statistiche"

Figura 3.43; in questo caso, l'eliminazione avviene semplicemente previa conferma da parte dell'utente.

Questo diagramma può essere applicato anche ai casi d'uso che riguardano l'eliminazione semplificata di qualsiasi altro oggetto all'interno del sistema.

Elimina lavoro

Nella Figura 3.45 è illustrato il diagramma di sequenza relativo al caso d'uso "Elimina lavoro". Si tratta di un diagramma analogo a quello mostrato in Figura 3.44. Vale la pena



Figura 3.31: Diagramma di attività generalizzato relativo ai casi d'uso che riguardano il filtraggio

evidenziare che, a differenza delle altre operazioni che riguardano i lavori, in questo caso non entrano in gioco i clienti. Questo diagramma e questo ragionamento possono essere applicati analogamente alla sottosezione "Preventivi".

3.4.5 Altre operazioni

Contatta cliente a partire da un lavoro

Nella Figura 3.46 è illustrato il diagramma di sequenza relativo al caso d'uso "Contatta cliente a partire da un lavoro". L'interazione inizia con l'utente che invoca la funzione `contatta()` della classe `vistaListaLavori`; successivamente il sistema recupera i dati del lavoro selezionato, estrae il cliente, da quest'ultimo recupera l'email e la dà in input alla funzione `open()` di `webbrowser`. Il risultato è l'apertura di una nuova mail da inviare, con il campo "Destinatario" precompilato con l'e-mail desiderata.

Questo diagramma può essere applicato anche ai casi d'uso "Contatta cliente a partire da un preventivo" e "Contatta dipendente a partire da una visita medica".

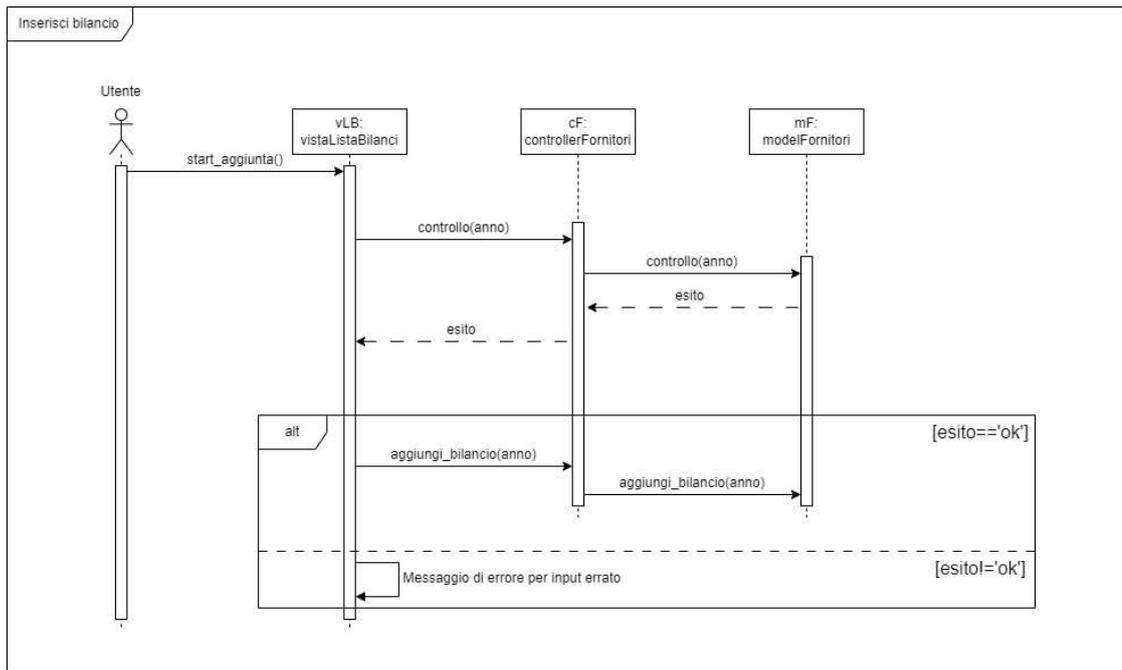


Figura 3.32: Diagramma di sequenza relativo al caso d'uso "Inserisci bilancio"

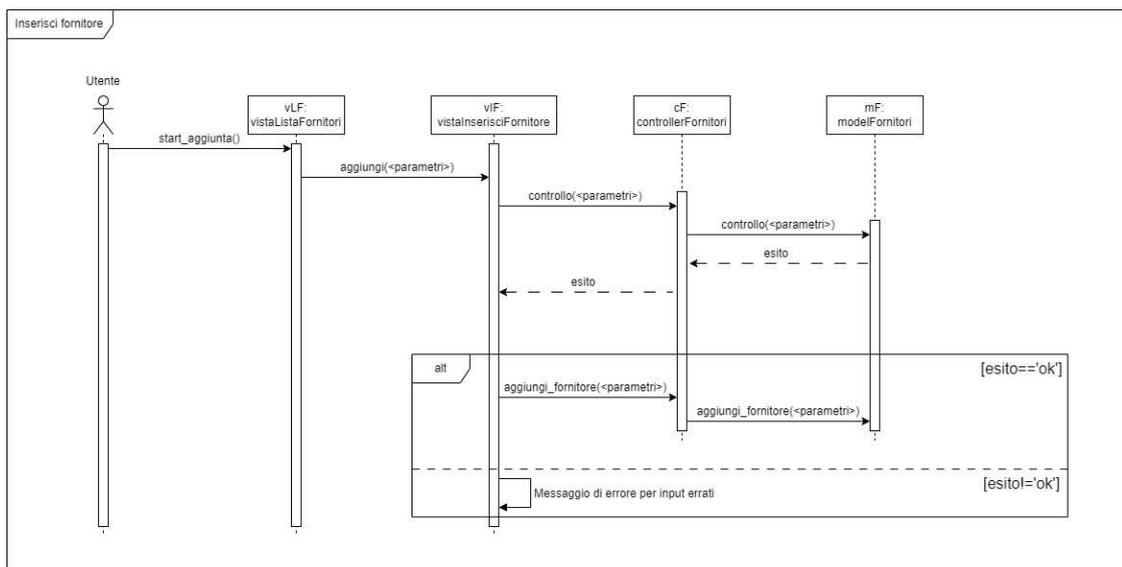


Figura 3.33: Diagramma di sequenza relativo al caso d'uso "Inserisci fornitore"

Contatta fornitore

Nella Figura 3.47 è illustrato il diagramma di sequenza relativo al caso d'uso "Contatta fornitore". Si tratta di una versione semplificata del diagramma mostrato in Figura 3.46: la differenza sta nel fatto che qui il sistema recupera soltanto il fornitore cliccato e la rispettiva e-mail.

Questo diagramma può essere applicato anche ai casi d'uso "Contatta cliente" e "Contatta dipendente".

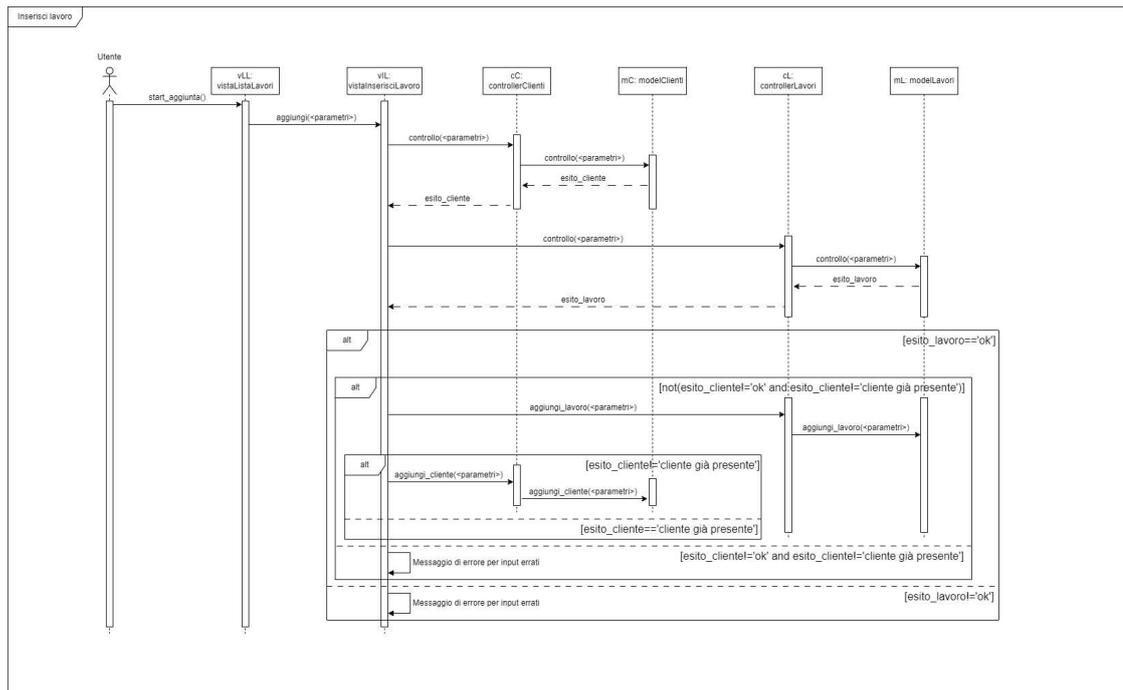


Figura 3.34: Diagramma di sequenza relativo al caso d'uso "Inserisci lavoro"

Visualizza magazzino e consuma materie prime

Nella Figura 3.48 è illustrato il diagramma di sequenza relativo alle interazioni che avvengono durante i casi d'uso "Visualizza magazzino", prima, e "Consuma materie prime", poi. Riguardo al primo caso d'uso, l'utente invoca `genera_lista()` di `vistaMagazzino` e il sistema, attraverso le classi che si occupano della gestione del magazzino, recupera e mostra sullo schermo le scorte presenti.

Per quanto riguarda il secondo, il sistema recupera la materia prima in magazzino, selezionata dall'utente attraverso `on_consumata_button_click()` della classe `vistaMagazzino`, e, se la rispettiva quantità è maggiore di 0, ne modifica il numero di scorte secondo il valore dato in input sempre dall'utente. Se non ci sono scorte, il sistema restituisce un messaggio di errore.

Lista spesa e conferma ordine

Nella Figura 3.49 è illustrato il diagramma di sequenza relativo alle interazioni che avvengono durante i casi d'uso che riguardano la gestione della lista della spesa (aggiunta/eliminazione di prodotti nella/dalla lista della spesa e conferma dell'ordine). Come prima cosa, il sistema recupera le materie prime in vendita e la lista della spesa, che sarà inizialmente vuota, e genera il totale, che sarà inizialmente €0 (da questo momento in poi queste ultime due operazioni verranno racchiuse in un'unica funzione chiamata `genera_lista_spesa()`); successivamente il cliente, attraverso le funzioni `on_inserisci_button_clicked()` e `on_rimuovi_button_clicked()` della classe `vistaAcquistaMateriePrime`, può aggiungere o rimuovere prodotti dalla lista spesa; dopo ognuna di queste operazioni, il sistema invocherà la funzione `genera_lista_spesa()` in modo che i cambiamenti siano dinamici. Quando il cliente è soddisfatto, invoca la funzione `on_conferma_button_clicked()` e il sistema, attraverso le classi che gestiscono il magazzino e quelle che gestiscono le uscite economiche, aggiorna la quantità di materie prime e inserisce l'uscita relativa alla spesa effettuata. Inoltre, nella fase di conferma dell'ordine, viene anche svuotata la lista della spesa.

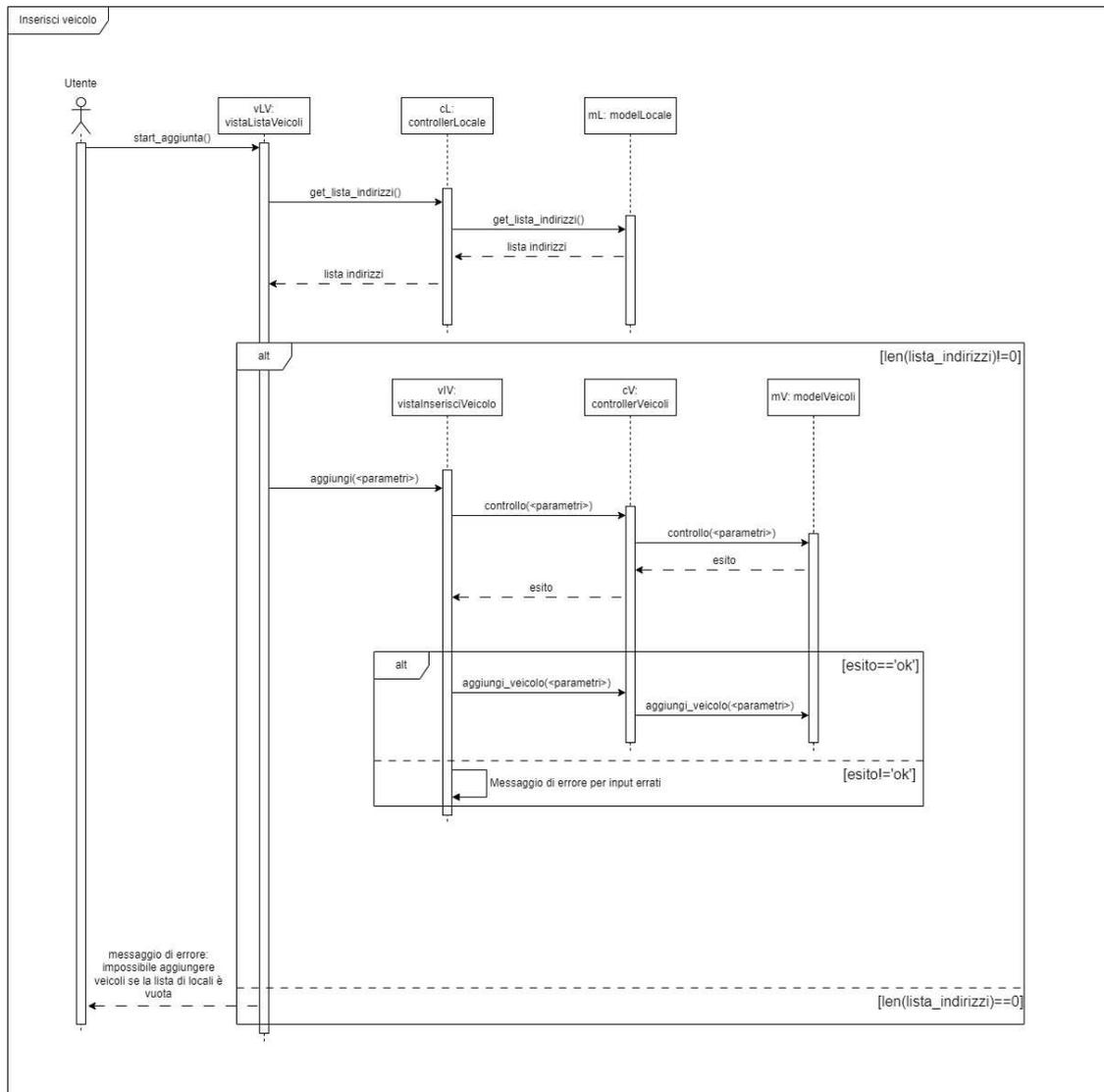


Figura 3.35: Diagramma di sequenza relativo al caso d'uso "Inserisci veicolo"

Scarica PDF preventivo

Nella Figura 3.50 è illustrato il diagramma di sequenza relativo al caso d'uso "Scarica PDF preventivo". L'interazione consiste nell'utente che, dopo aver visualizzato i dettagli del preventivo da scaricare, attraverso l'invocazione di `save_as_pdf()` di `vistaPreventivo`, avvia la procedura di download. Il sistema, attraverso particolari librerie Python, si occupa di comporre il documento, come vedremo nel capitolo successivo.

Questo diagramma può essere applicato anche al caso d'uso "Scarica PDF contratto dipendente"

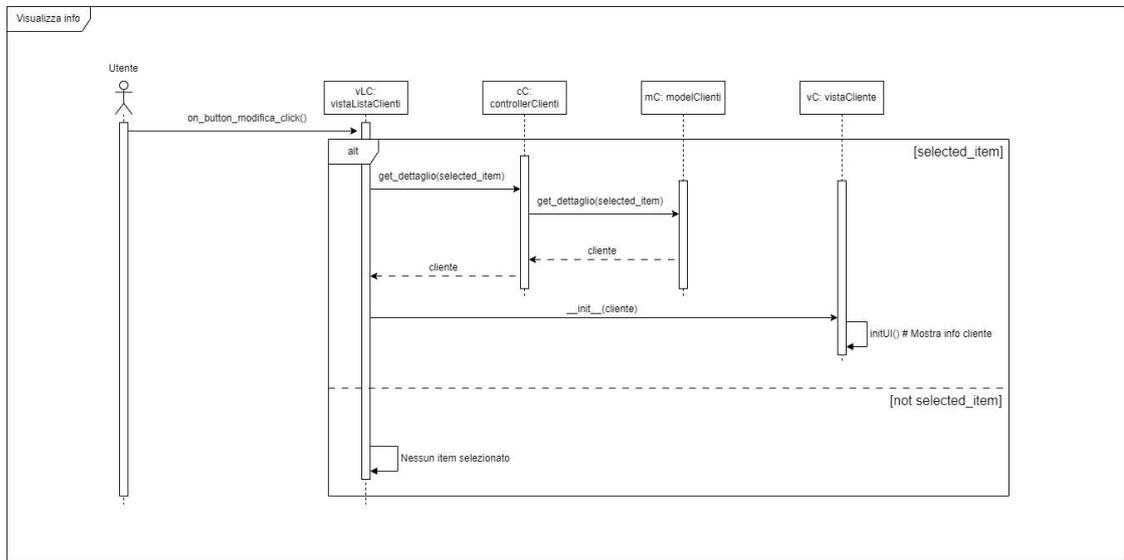


Figura 3.36: Diagramma di sequenza relativo al caso d’uso "Visualizza dettaglio cliente"

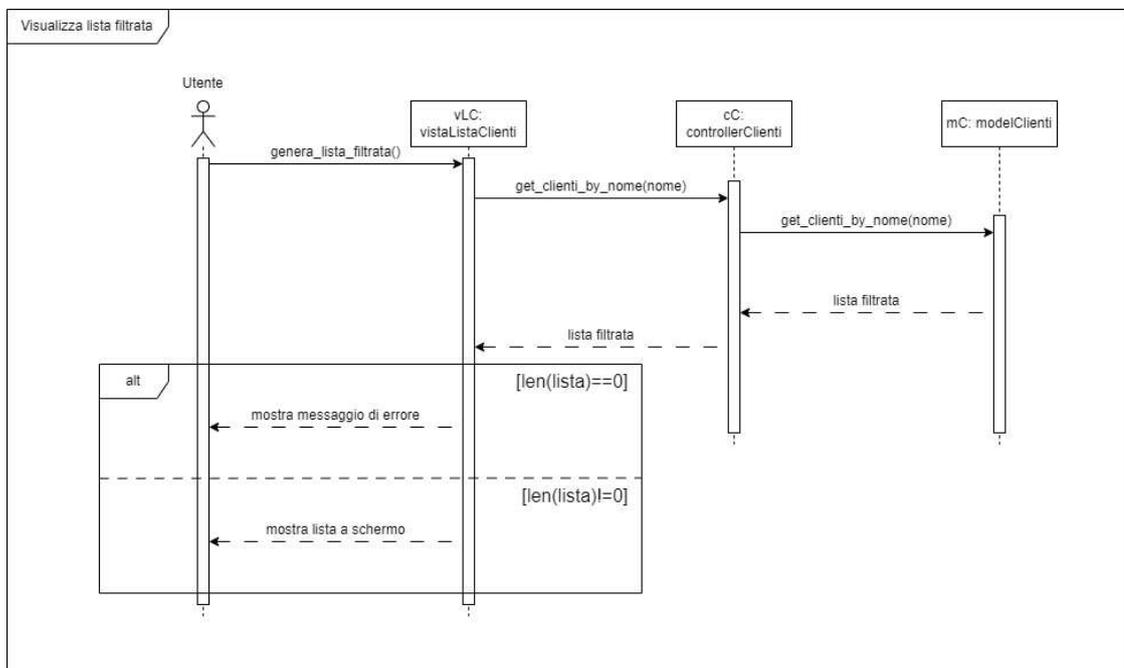


Figura 3.37: Diagramma di sequenza relativo al caso d’uso "Filtra clienti per nome"

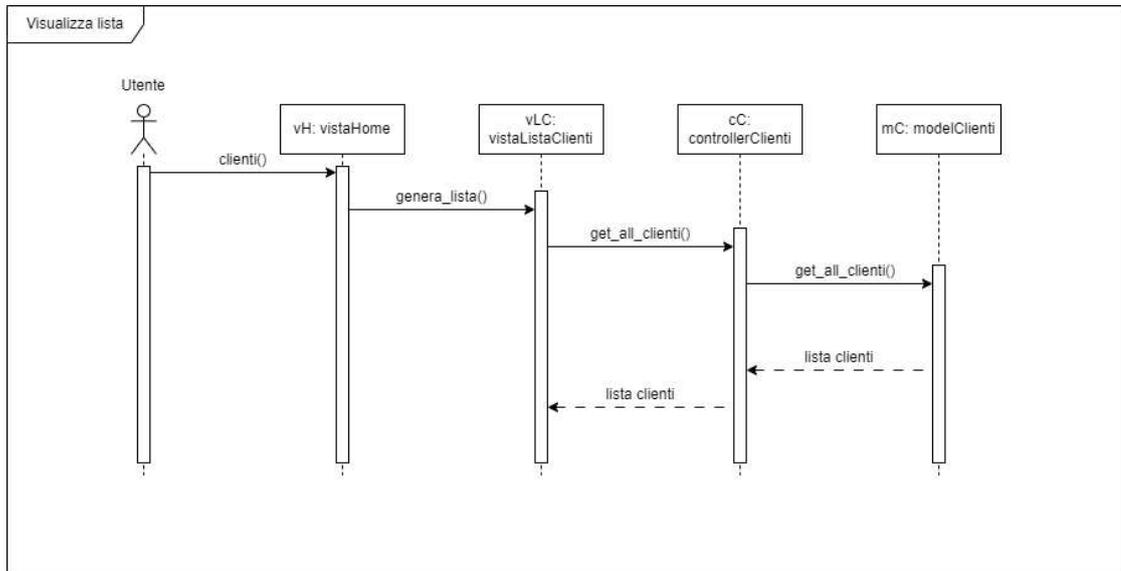


Figura 3.38: Diagramma di sequenza relativo al caso d'uso "Visualizza lista clienti"

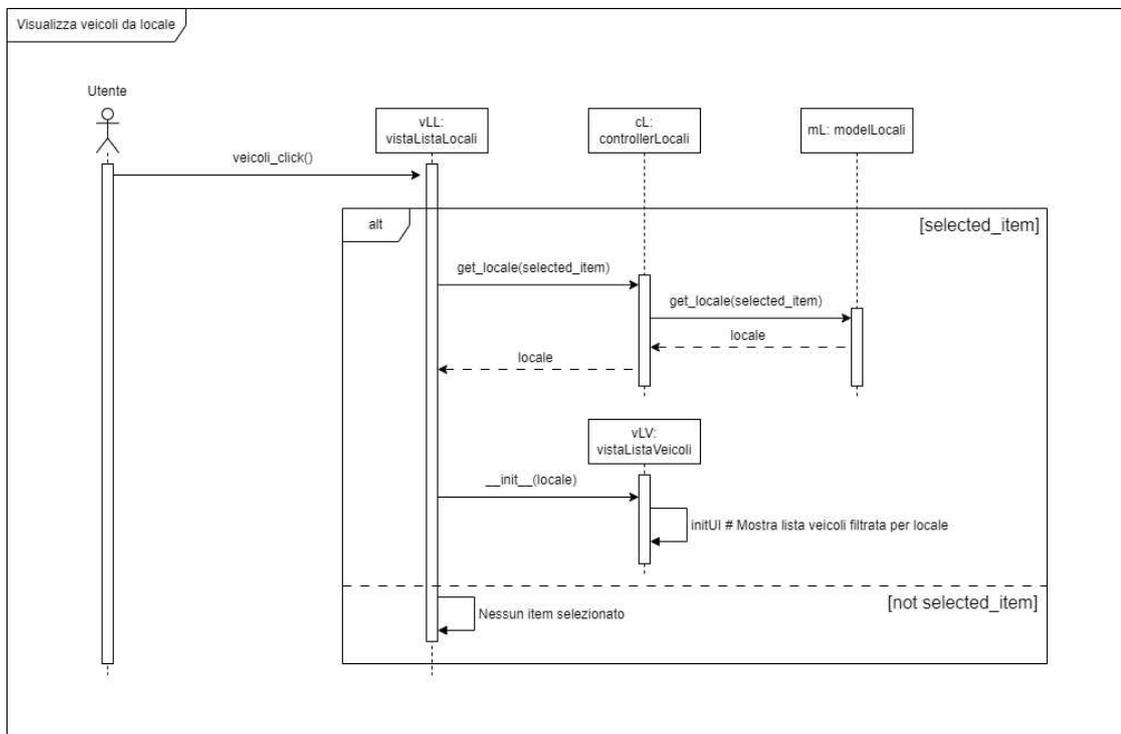


Figura 3.39: Diagramma di sequenza relativo al caso d'uso "Visualizza lista veicoli a partire da un locale"

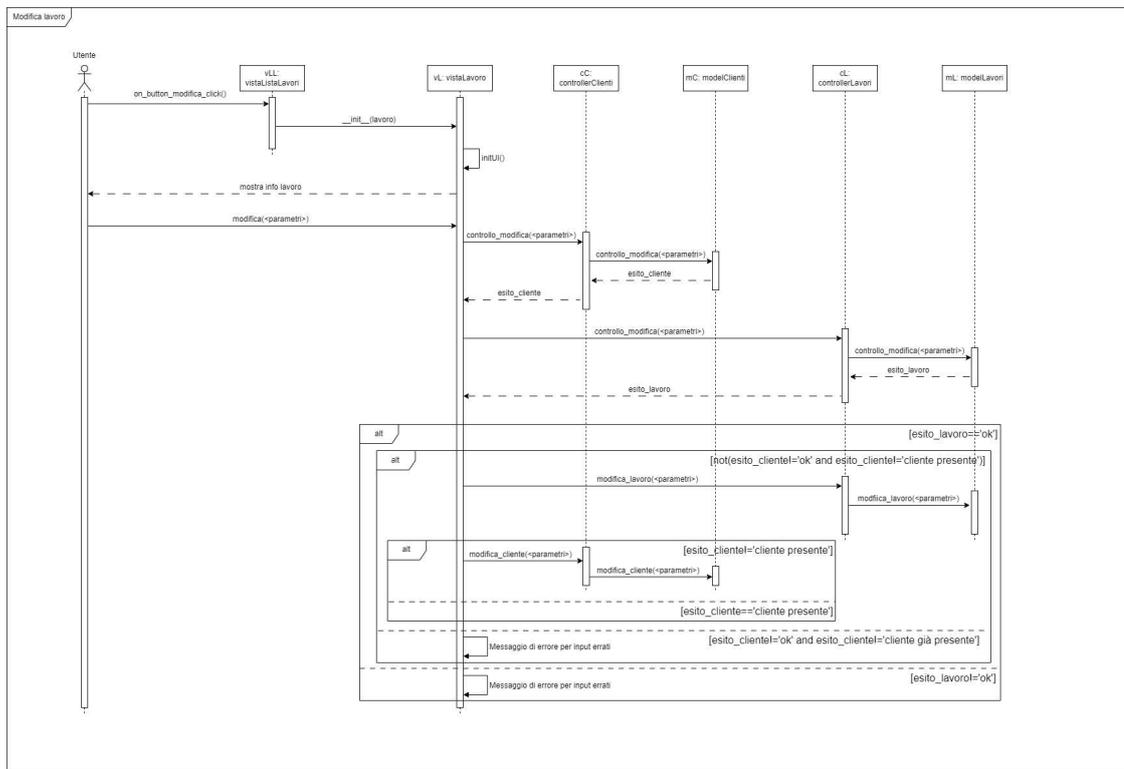


Figura 3.40: Diagramma di sequenza relativo al caso d'uso "Modifica lavoro"

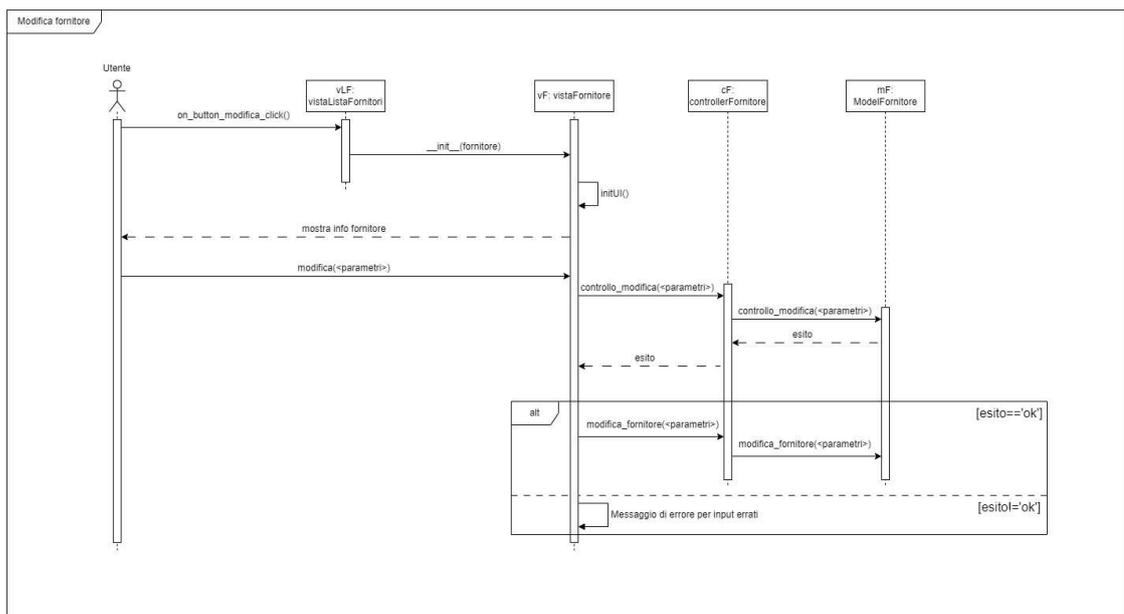


Figura 3.41: Diagramma di sequenza relativo al caso d'uso "Modifica fornitore"

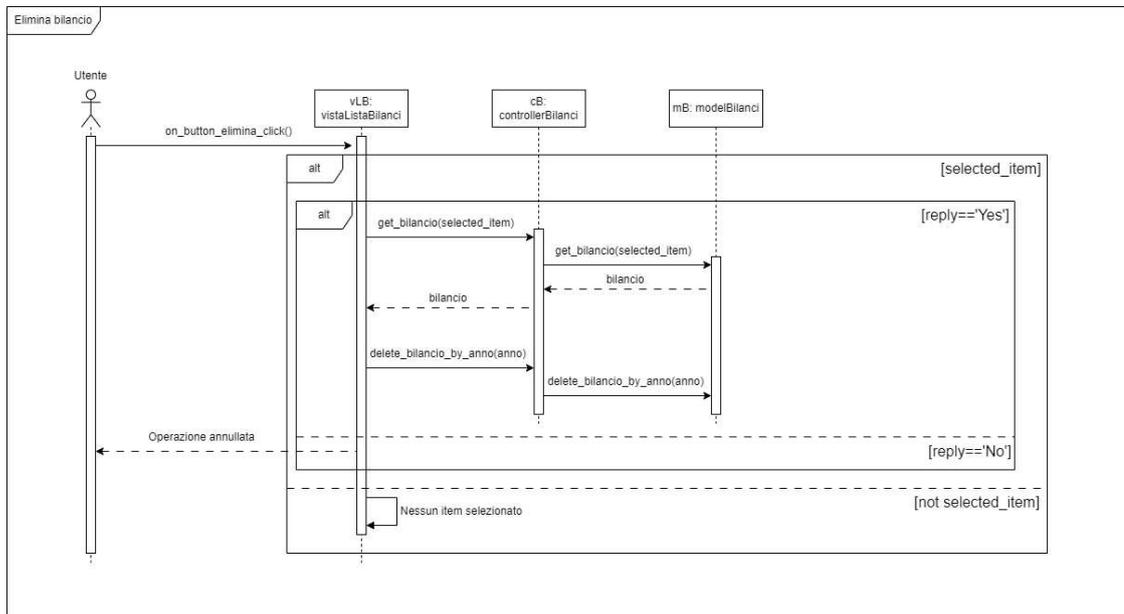


Figura 3.42: Diagramma di sequenza relativo al caso d'uso "Elimina bilancio"

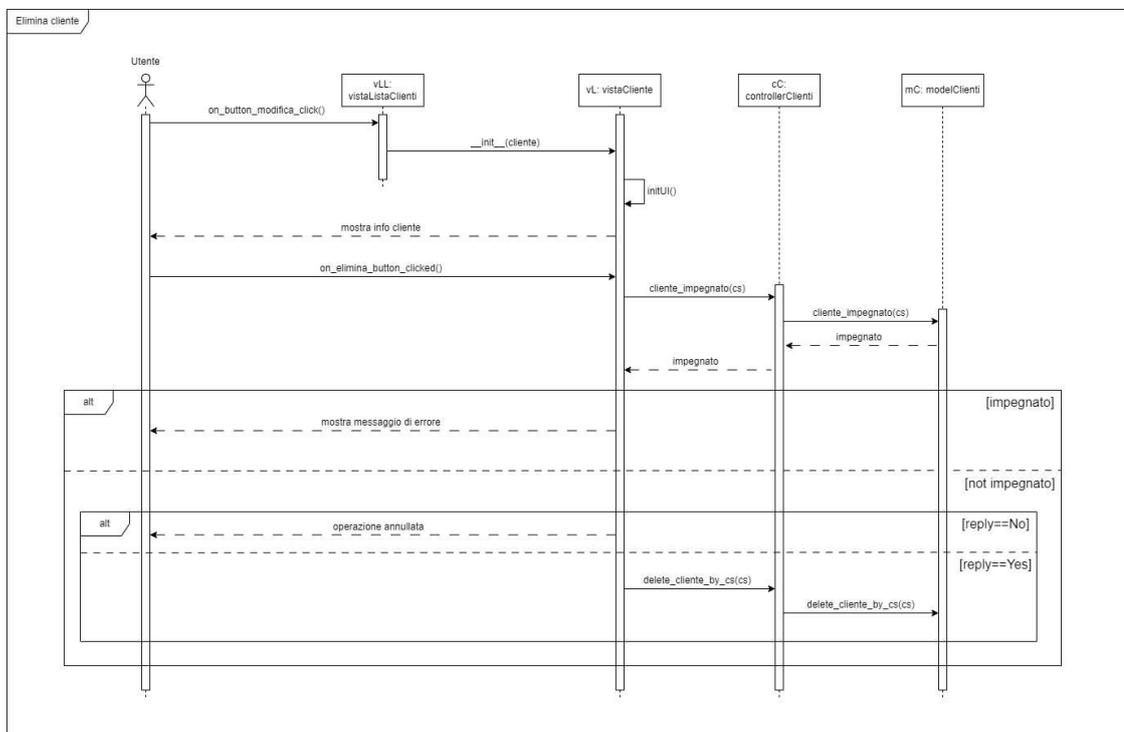


Figura 3.43: Diagramma di sequenza relativo al caso d'uso "Elimina cliente"

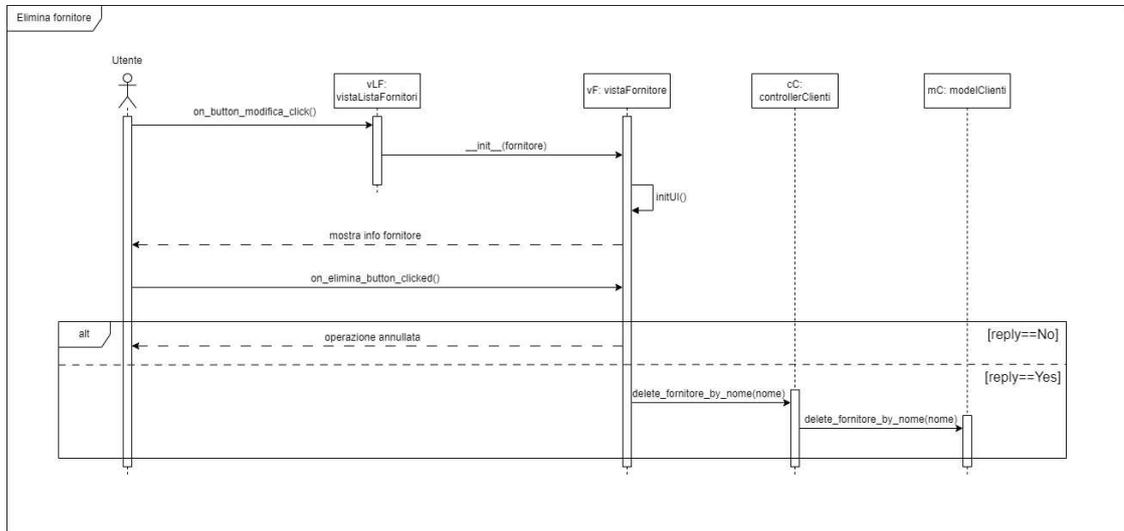


Figura 3.44: Diagramma di sequenza relativo al caso d’uso "Elimina fornitore"

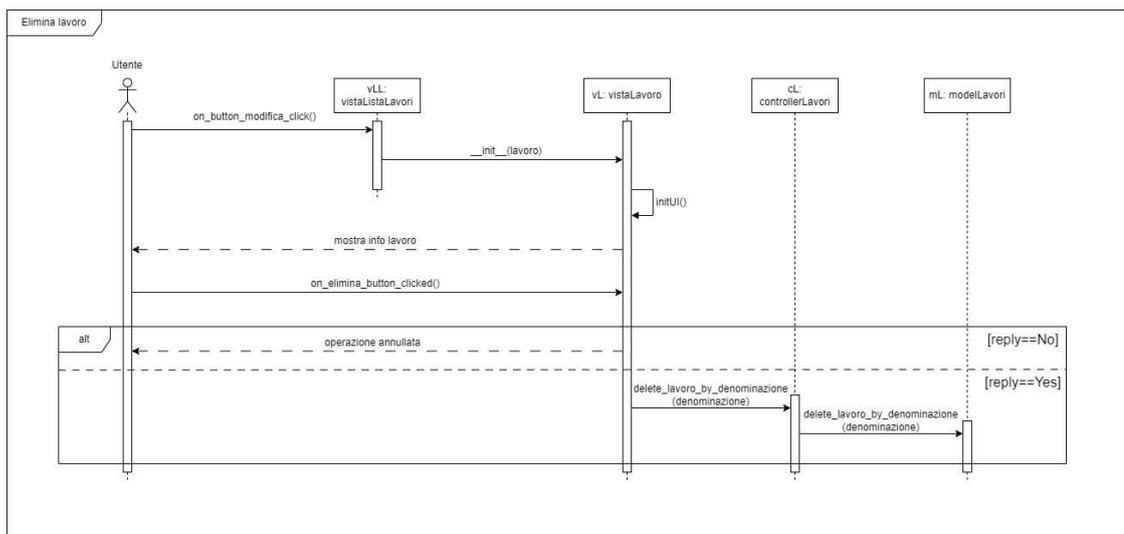


Figura 3.45: Diagramma di sequenza relativo al caso d’uso "Elimina lavoro"

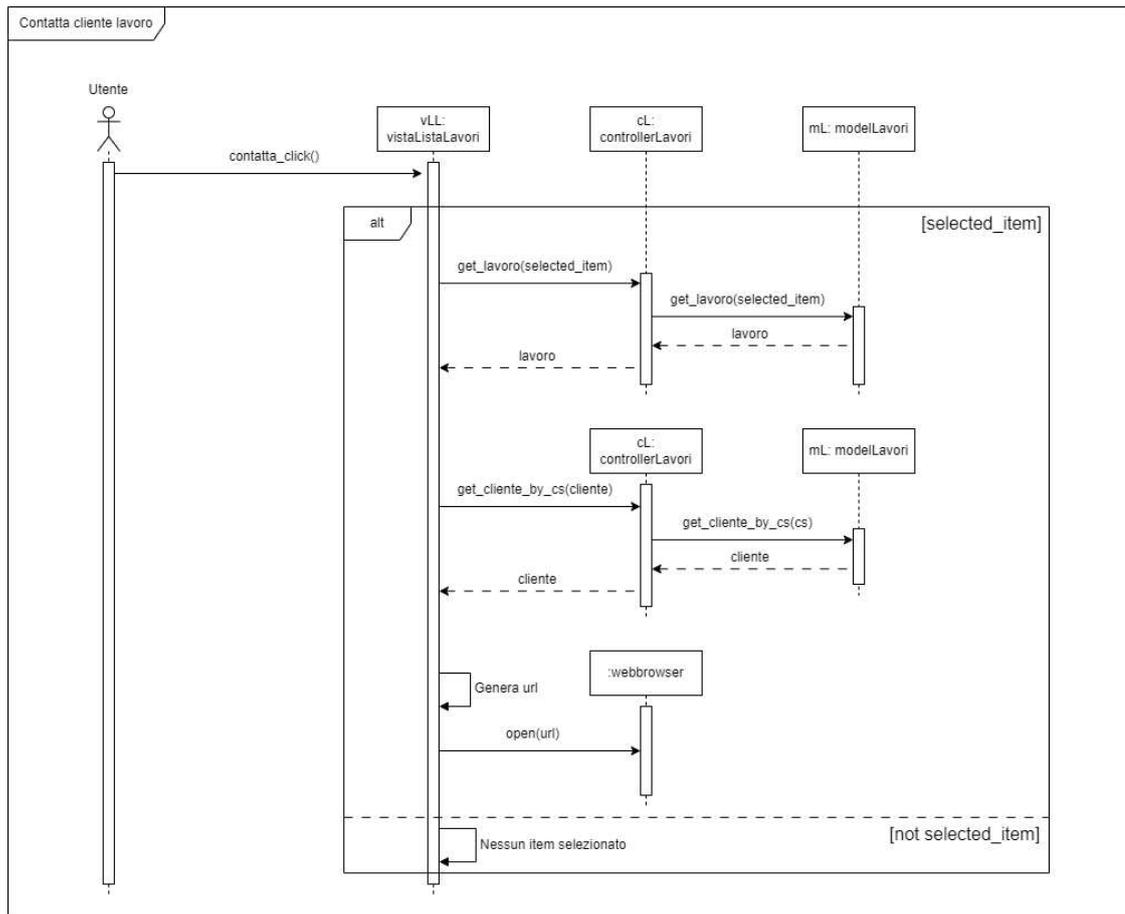


Figura 3.46: Diagramma di sequenza relativo al caso d’uso "Contatta cliente a partire da un lavoro"

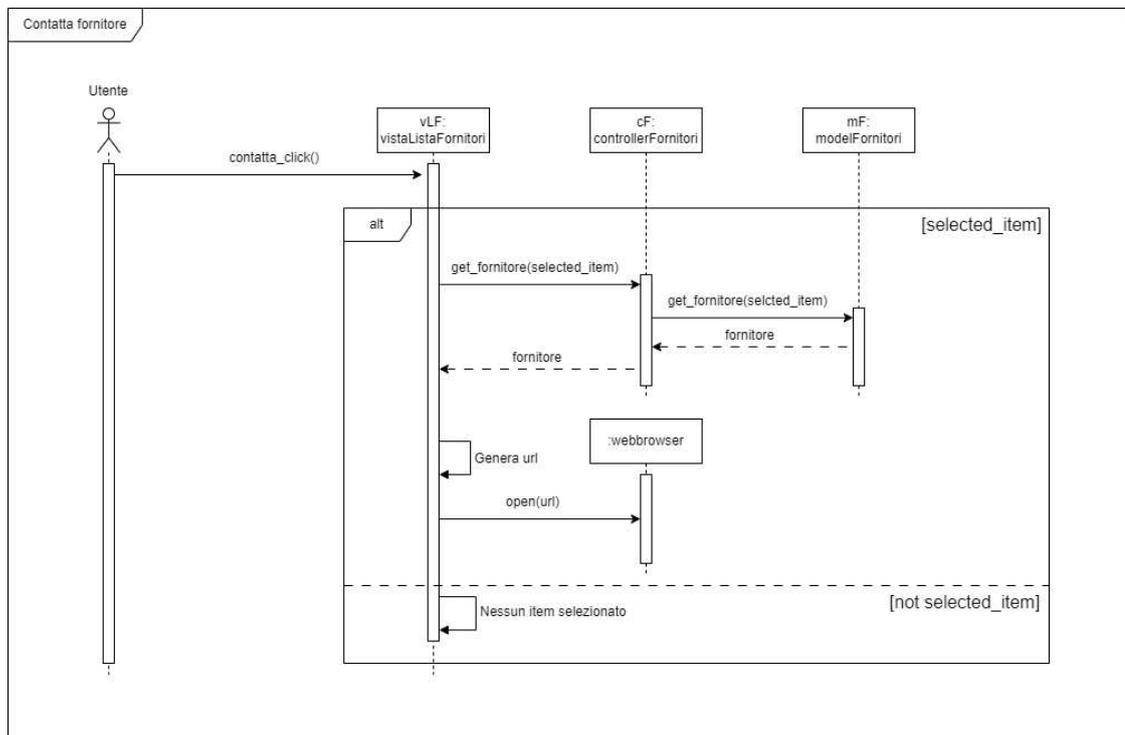


Figura 3.47: Diagramma di sequenza relativo al caso d’uso "Contatta fornitore"

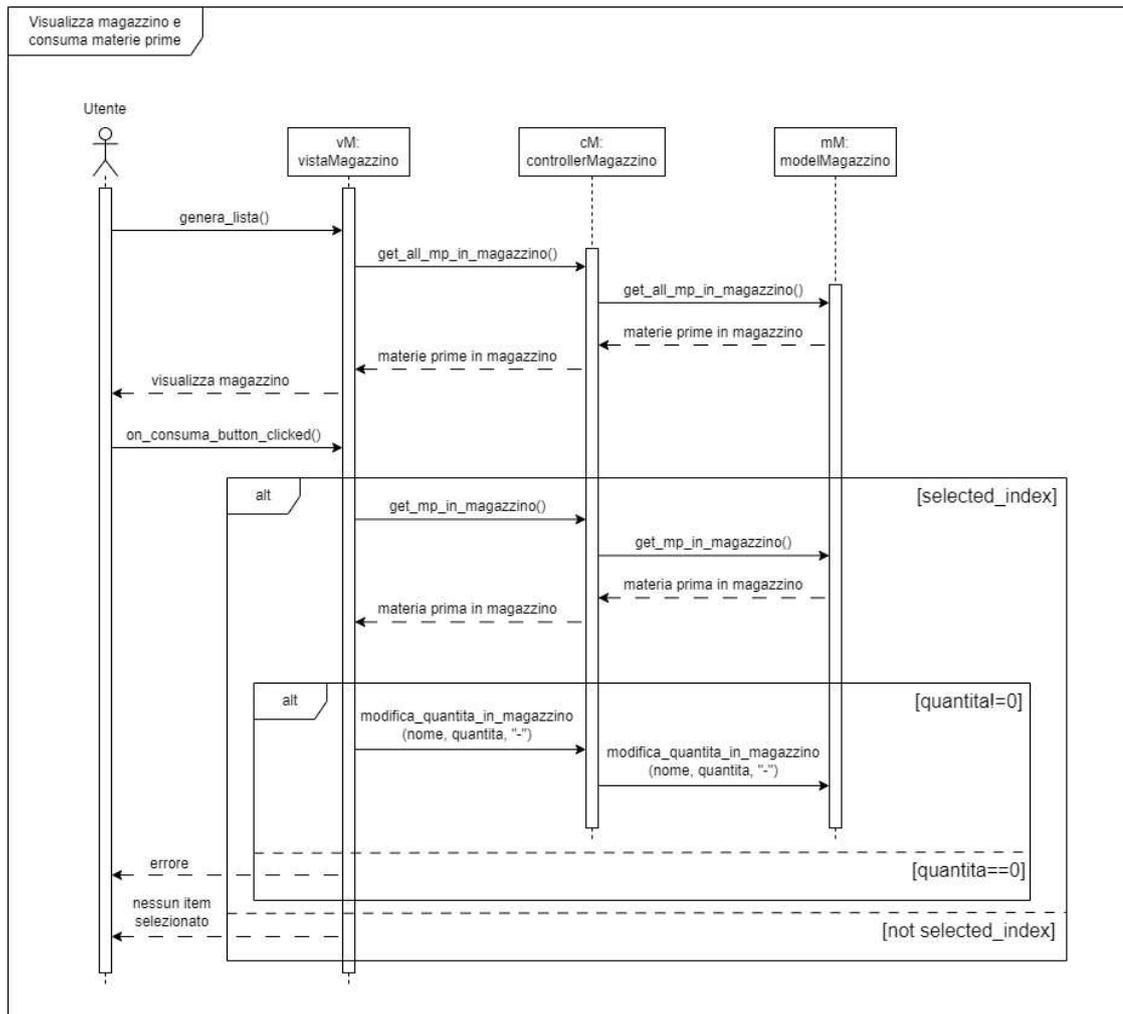


Figura 3.48: Diagramma di sequenza relativo ai casi d’uso "Visualizza magazzino" e "Consuma materie prime"

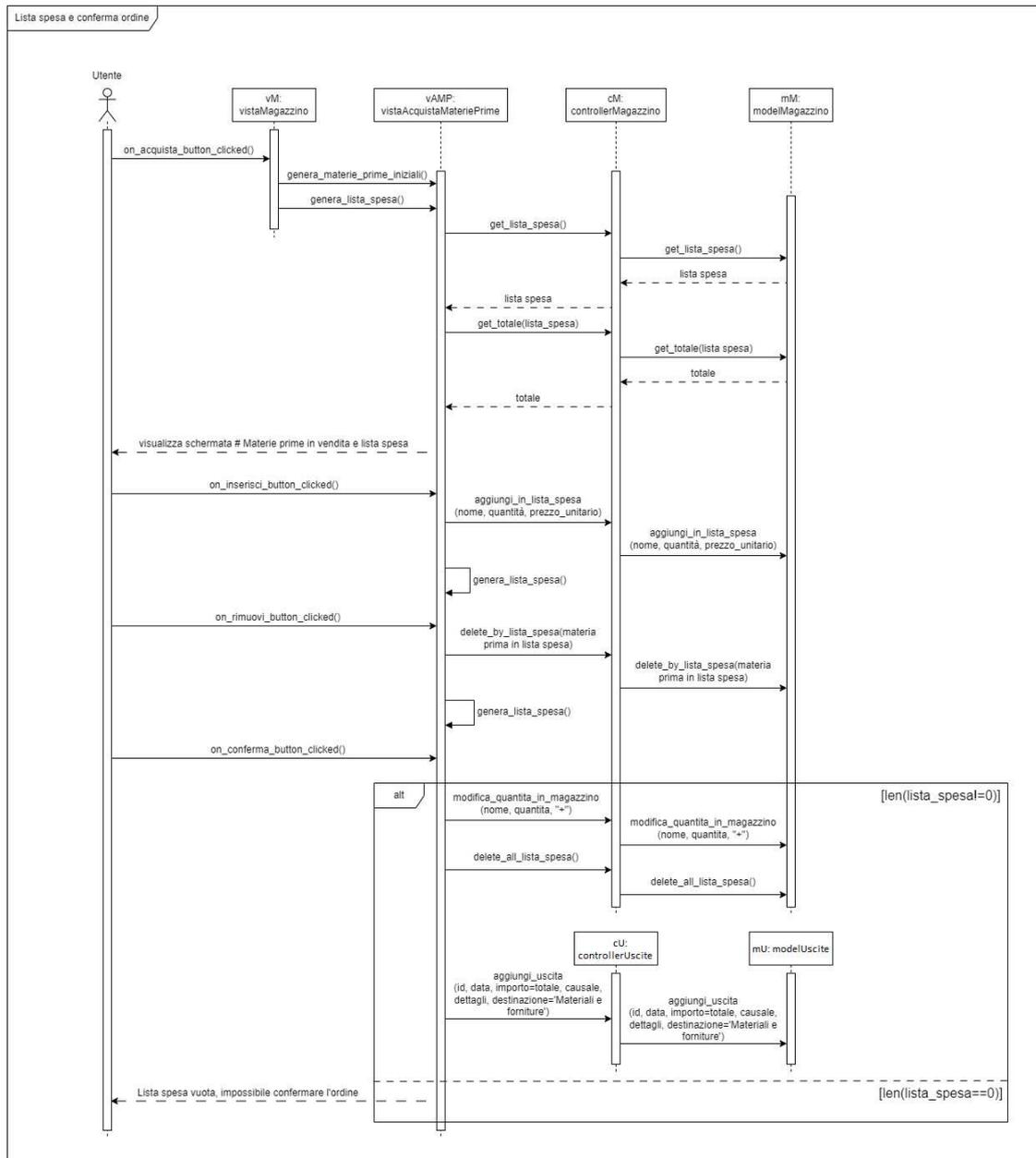


Figura 3.49: Diagramma di sequenza relativo all'interazione dell'utente con la lista della spesa e al caso d'uso "Conferma ordine"

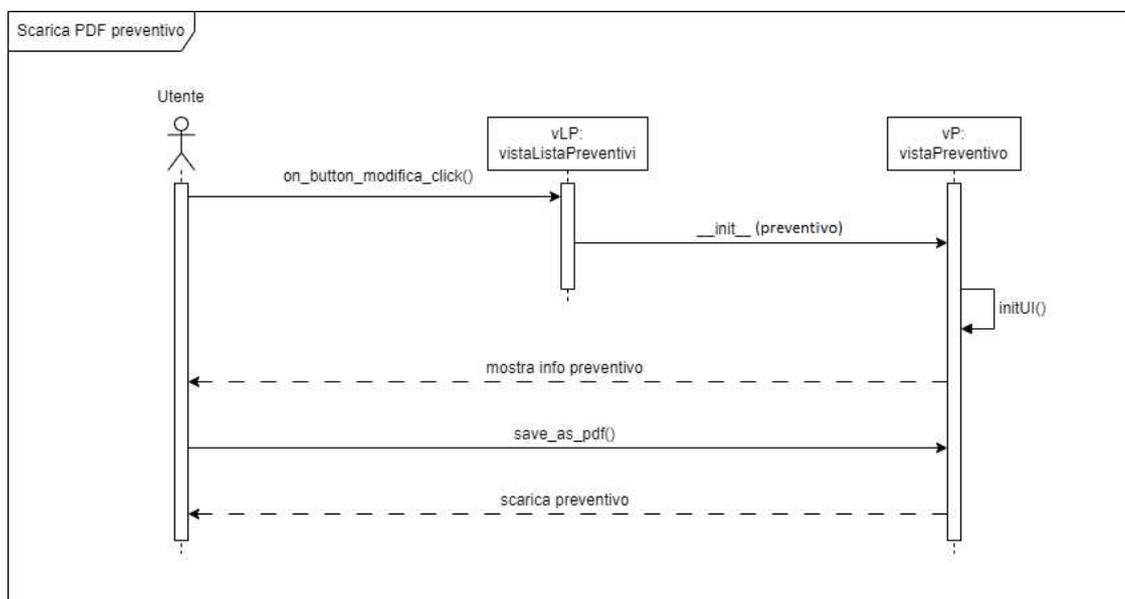


Figura 3.50: Diagramma di sequenza relativo al caso d'uso "Scarica preventivo"

Nel quarto capitolo ci occuperemo, come prima cosa, dell'implementazione; in particolare, vedremo i principali strumenti tecnici, come programmi applicativi, linguaggi di programmazione e librerie varie, e come applicarli al fine della realizzazione del prodotto finale. Nella seconda sezione, testeremo alcune funzionalità del sistema, al fine di verificare che queste vengano eseguite correttamente. Per fare questo, ci serviremo del framework di automazione dei test PyUnit.

4.1 Tecnologie utilizzate

4.1.1 Python

Python è un linguaggio di programmazione avanzato che ben si presta ad essere utilizzato in ambito didattico per la sua immediatezza e semplicità. Viene rilasciato nel 1991 dal ricercatore olandese Guido van Rossum; il suo nome (in italiano "pitone") deriva dall'irriverente e divertente serie televisiva inglese Monty Python's Flying Circus, di cui Guido era appassionato. Python si caratterizza per essere:

- *di alto livello e interpretato*: il programmatore non deve occuparsi degli aspetti di basso livello legati all'hardware; il programma interprete di Python si occupa di tradurre le righe scritte in Python in istruzioni direttamente comprensibili al computer (linguaggio macchina);
- *multiplatforma*: i programmi scritti in Python sono portabili, cioè possono essere eseguiti su differenti sistemi operativi, tra i quali Windows, Mac OS X e Linux;
- *open source e gratuito*: chiunque può usarlo per scrivere programmi e vedere il codice sorgente, con cui è stato scritto il programma interprete;
- *semplice e conciso*: la sintassi del linguaggio è, in generale, facile da imparare e usare;
- *general purpose*: Python non è legato allo sviluppo di specifiche applicazioni, ma si presta alla realizzazione di programmi di varia natura.

Python è composto da un ecosistema formato da più elementi. Oltre al linguaggio, infatti, ci sono:

- un *programma che interpreta* ed esegue codice scritto nel linguaggio Python;

- un *ambiente di sviluppo (IDE)*, che permette di scrivere ed eseguire i programmi, richiamando il programma interprete;
- un insieme di *librerie*, che raccolgono codice scritto da altri sviluppatori, utilizzabile nei nostri programmi;
- una *comunità* di persone, aziende ed enti che ruotano attorno a Python, occupandosi del suo sviluppo e della sua divulgazione.

Python viene utilizzato in diversi settori (dalla chimica all'astrofisica, dalla geologia alla biologia) e realtà (da Google alla NASA, dalla Pixar alla Intel) ed è stato scelto da molti ricercatori per l'analisi di grandi quantità di dati (big data), dalle quali ricavare informazioni su cui effettuare analisi e predizioni (data mining). Python è molto utilizzato anche nel nostro settore applicativo, ovvero quello di sviluppo di programmi gestionali.

Primo esempio di un programma scritto in Python

Nel Listato 4.1 è mostrato un esempio di codice scritto in Python; nello specifico, si tratta dell'entry point del nostro sistema.

```
1 # Importo tutte le librerie che mi servono
2 import sys
3 import time
4
5 from PyQt5.QtCore import Qt
6 from PyQt5.QtGui import QPixmap
7 from PyQt5.QtWidgets import QApplication, QSplashScreen
8
9 from view.autenticazione.vistaAccedi import vistaAccedi
10
11 # Istruzione condizionale: se questa cosa e' vera, allora vai avanti
12 if __name__ == "__main__":
13     # Creo un'istanza della classe QApplication e la chiamo app
14     # Tra parentesi c'e' il parametro che passiamo
15     app = QApplication(sys.argv)
16
17     # Creo un'istanza della classe QSplashScreen e la chiamo splash
18     # Prende in input due parametri, un'istanza di QPixmap, che, a sua volta,
19     # prende in input una stringa, e un attributo della classe Qt (e' possibile
20     # accedere agli attributi di una classe con la dot notation)
21     splash = QSplashScreen(QPixmap("images\\logo_splash.png"), Qt.
22     WindowStaysOnTopHint)
23     # Effettuo una chiamata al metodo show() dell'oggetto splash, precedentemente
24     # istanziato (chiamata di metodi con la dot notation)
25     splash.show()
26     # Effettuo una chiamata al metodo sleep(<time>) del modulo time
27     time.sleep(1)
28     # Instancio un oggetto della classe vistaAccedi e lo chiamo finestra
29     finestra = vistaAccedi()
30     # Chiamo il metodo show dell'oggetto finestra istanziato in precedenza
31     finestra.show()
32     # Chiamo il metodo finish dell'oggetto splash
33     # Passo in input l'oggetto finestra
34     splash.finish(finestra)
35     # Avvio il ciclo principale dell'applicazione, gestendo eventi e interazioni
36     # dell'utente, assicurandomi che il programma esca correttamente quando l'
37     # applicazione viene chiusa
38     # Esempio di funzione di ordine superiore: funzione che prende in input un'
39     # altra funzione, detta funzione di callback
40     sys.exit(app.exec())
```

Listing 4.1: Esempio di codice Python**Definizione di una classe in Python**

Nel Listato 4.2 vediamo un esempio di definizione di una classe (concetto molto importante nella Programmazione Orientata agli Oggetti), nello specifico, si tratta di `vistaAccedi`, ovvero la view che mostra la schermata di login del sistema.

```
1 from PyQt5.QtCore import Qt
2 # Altri import...
3
4 # Definizione di una classe
5 # Tra parentesi, la classe da cui eredita
6 class vistaAccedi(QMainWindow):
7     # Costruttore, viene invocato ogni volta che viene creata un'istanza della
8     # classe
9     def __init__(self):
10        # Richiama il costruttore della classe da cui eredita
11        super().__init__()
12
13        # Attributi della classe inizializzati
14        self.password_visibile = False
15        self.controller = self.genera_controller()
16
17        # Richiamo un metodo della superclasse
18        self.setWindowIcon(QIcon("images\\logo_senza_nome.png"))
19
20        # Richiamo il metodo initUI della classe
21        self.initUI()
22
23        # Definizione dei metodi della classe
24        def initUI(self):
25            # Implementazione interfaccia grafica
26
27            # Esempio di funzione che prende in input un'altra funzione
28            self.login_button.clicked.connect(self.accedi)
29
30            # Altro codice
31
32        def accedi(self):
33            # Corpo della funzione
```

Listing 4.2: Esempio di definizione di una classe**4.1.2 Git & GitHub****GitHub**

GitHub è una piattaforma cloud-based in cui è possibile archiviare, condividere e collaborare con altri utenti per scrivere codice. Memorizzare del codice in un "repository" su GitHub consente all'utente di:

- mostrare e condividere il proprio lavoro;
- tenere traccia e gestire le modifiche apportate al codice nel tempo;

- lasciare che altri utenti rivedano il proprio codice, in modo che possano fornire suggerimenti per migliorarlo;
- collaborare a un progetto condiviso.

Il lavoro collaborativo, una delle caratteristiche fondamentali di GitHub, è reso possibile dal software open source Git, su cui GitHub è basato.

Git

Git è un sistema di controllo delle versioni che traccia in modo intelligente le modifiche nei file; è particolarmente utile quando un gruppo di persone apporta modifiche agli stessi file contemporaneamente. Per fare questo, c'è un flusso di azioni che ogni sviluppatore che lavora ad un progetto dovrebbe seguire:

- creare un ramo dalla copia principale dei file su cui si sta lavorando;
- apportare modifiche ai file in modo autonomo e sicuro sul proprio ramo personale;
- lasciare che Git unisca in modo intelligente le proprie modifiche specifiche alla copia principale dei file, in modo che le proprie modifiche non influiscano sugli aggiornamenti di altre persone;
- lasciare che Git tenga traccia delle proprie modifiche e di quelle degli altri, in modo da poter continuare a lavorare sulla versione più aggiornata del progetto.

Come lavorano insieme Git e GitHub

Quando un utente carica file su GitHub, li memorizza in un "repository Git". Ciò significa che quando vengono apportate delle modifiche (o "commit") ai file su GitHub, Git inizierà automaticamente a tracciare e gestire queste modifiche.

Ci sono moltissime azioni correlate a Git che si possono completare su GitHub direttamente nel browser, come la creazione di un repository Git, la creazione di branch e il caricamento e la modifica di file.

Tuttavia, la maggior parte delle persone lavora sui propri file localmente (sul proprio computer), quindi sincronizza continuamente queste modifiche locali, e tutti i dati Git correlati, con il repository centrale "remoto" su GitHub.

Quando un utente vuole iniziare a collaborare con altri e tutti hanno bisogno di lavorare contemporaneamente sullo stesso repository, ci sono due azioni da fare in continuazione:

- *pull*: recuperare tutte le ultime modifiche apportate dagli altri collaboratori dal repository remoto su GitHub;
- *push*: caricare le proprie modifiche allo stesso repository remoto su GitHub;

Schermate di GitHub

Nella Figura 4.1 vediamo la Home Page di GitHub, versione web. Di nostro interesse è la sezione sulla sinistra, dove possiamo trovare un elenco di tutti i repository a cui l'utente collabora. Ogni item di questo elenco è così formato: <nome_proprietario>/<nome_repository>. Al primo posto troviamo il repository del nostro progetto; cliccandoci sopra, possiamo visualizzare la schermata che vediamo in Figura 4.2. Le sezioni che ci interessano maggiormente in questa pagina sono:

- *la barra di navigazione*: in alto, sotto al logo di GitHub, consente di navigare tra le diverse funzionalità e sezioni del repository
- *il pannello centrale*: qui troviamo, nella prima riga, il messaggio e la data dell'ultimo commit eseguito e il numero di commit totali effettuati; inoltre, possiamo vedere, nelle righe successive, tutte le cartelle e i file di progetto, affiancati dal nome e dalla data dell'ultimo commit effettuato su ognuno di essi;
- *il readme*: subito sotto al pannello centrale, in questa sezione possiamo leggere il titolo e una breve descrizione del progetto;
- *il pannello sulla destra*: qui vediamo alcune informazioni sul repository, come activity, stars, watching, forks, oltre che, più in basso, un grafico che mostra i linguaggi di programmazione utilizzati.

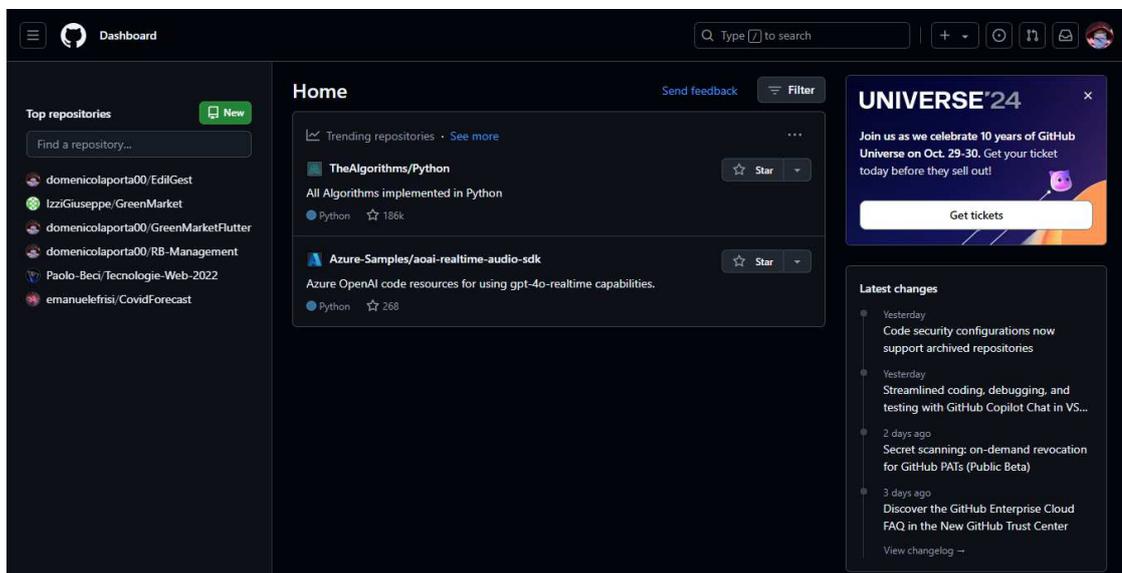


Figura 4.1: Home Page di GitHub

4.1.3 PyCharm

Per lo sviluppo professionale, sono disponibili numerosi IDE (Integrated Development Environment) più evoluti di IDLE. Sono applicazioni più complesse da imparare a utilizzare che, però, consentono una migliore scrittura e test di programmi Python, integrando editor, debugger, e ambienti di prova ed esecuzione. Tra gli IDE disponibili in versione gratuita citiamo Wing101, PyDev e PyCharm, su cui è andata la nostra scelta finale.

Nella Figura 4.3 vediamo l'interfaccia utente di PyCharm. Ne elenchiamo le principali sezioni e, per ognuna, ne spieghiamo brevemente il funzionamento:

- *Window header*: contiene un set di widget che forniscono un accesso rapido alle azioni più popolari; in particolare, attraverso il triangolino verde sulla destra, è possibile mandare il programma in esecuzione.
- *Editor*: spazio al centro, dove, trascinando o selezionando un qualsiasi file, è possibile visualizzarlo e modificarlo; può contenere più file aperti (di cui, però, solo uno alla volta visualizzabile), e ciò rende la navigazione più facile.

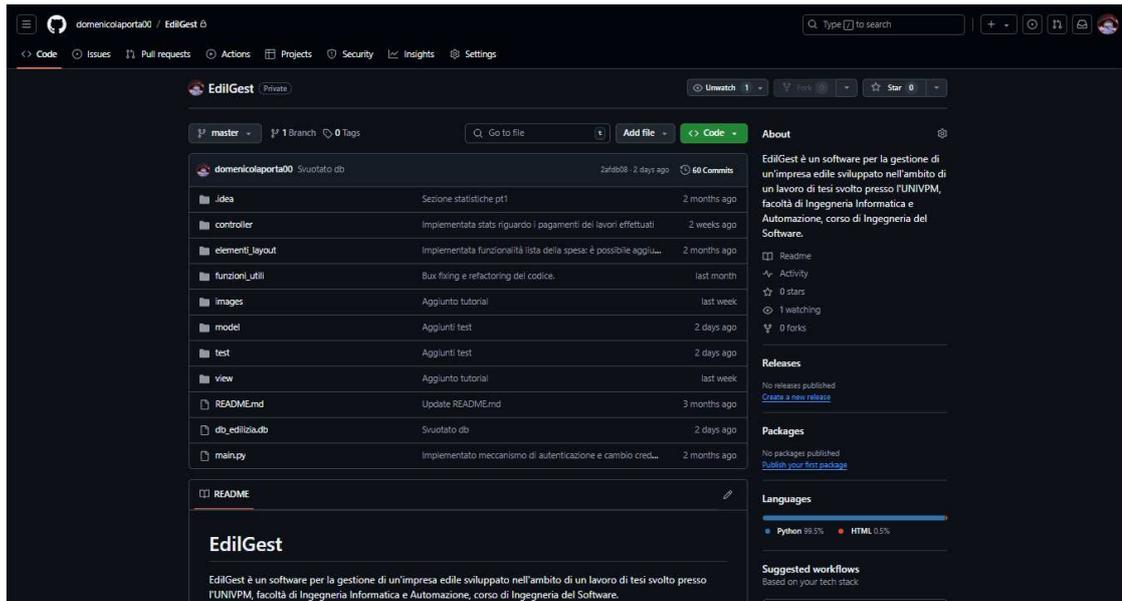


Figura 4.2: Repository del progetto

- *Tool window bar*: sono le barre ai lati destro e sinistro della finestra; forniscono l'accesso a compiti tipici come l'esecuzione, i test, il debug, la gestione dei database, la visualizzazione di grafici, l'integrazione con Git, e così via.
- *Tool windows*: sono il pannello in basso e quello sulla destra; gestiscono le operazioni elencate nel punto precedente; in questo caso è aperta la Run tool window, ovvero la finestra che gestisce l'esecuzione dell'applicazione, in basso, e la Database tool window, cioè la finestra che gestisce il database, sulla destra.
- *Project tool window*: è il pannello sulla sinistra e fa parte sempre delle tool window; contiene tutti i file e le cartelle di progetto.

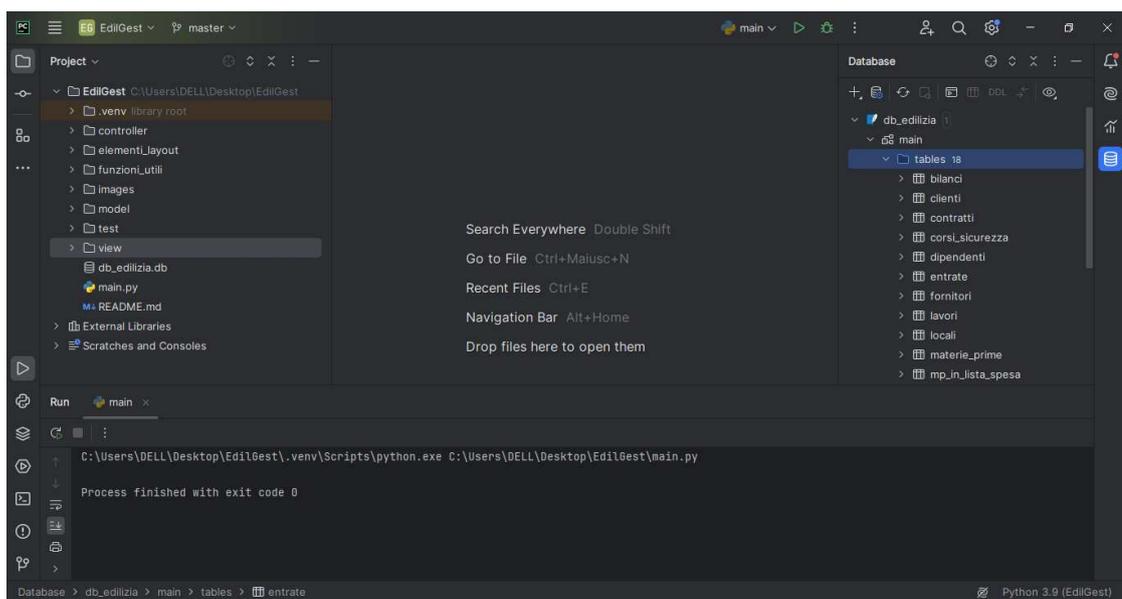


Figura 4.3: UI PyCharm

PyCharm mette a disposizione anche delle funzionalità integrate con Git e GitHub; dopo aver collegato il nostro progetto con il repository su GitHub, abbiamo potuto lavorare utilizzando esclusivamente queste funzionalità; questo approccio è stato sicuramente più efficiente, rispetto all'utilizzo di GitBash, ovvero la console di Git. Nella Figura 4.4 possiamo vedere alcune di queste funzionalità; più specificatamente:

- Nella tool window sulla sinistra vediamo i file che sono stati modificati rispetto all'ultimo aggiornamento; selezionandoli e cliccando sul pulsante "Commit and Push", possiamo effettuare il commit e il push (accompagnati anche da un messaggio) e aggiornare la versione online del repository con le ultime modifiche effettuate.
- Nella tool window in basso vediamo tutta la storia del repository, quindi tutti i commit che sono stati effettuati, con l'utente e la data.
- Nell'editor vediamo il confronto tra la versione del file `main.py` che si trova sul repository online e quella che abbiamo in locale.

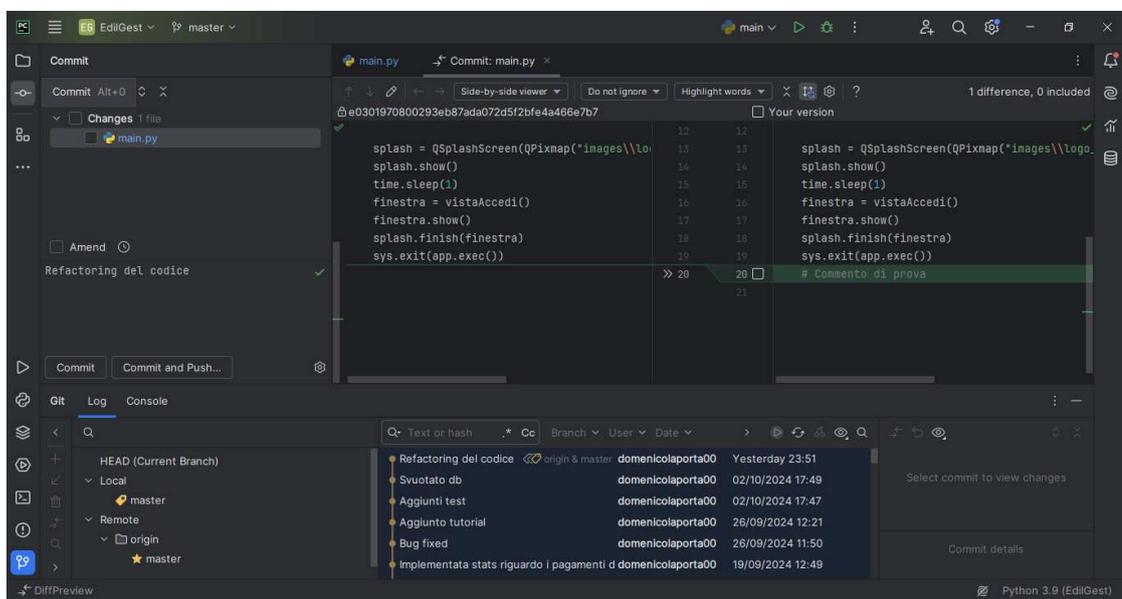


Figura 4.4: Funzionalità di Git integrate in PyCharm

4.1.4 SQLite3

SQLite è una libreria scritta in linguaggio C che implementa un gestore di database SQL piccolo, veloce, auto-contenuto, altamente affidabile e completo. SQLite è il gestore di database più utilizzato al mondo. È integrato in tutti i telefoni cellulari e nella maggior parte dei computer, ed è incluso in innumerevoli altre applicazioni che le persone utilizzano ogni giorno.

A differenza della maggior parte degli altri database SQL, non ha un processo server separato, ma legge e scrive direttamente su normali file su disco. Il formato del file del database è multipiattaforma. Queste caratteristiche rendono SQLite una scelta popolare come formato di file per applicazioni. I file di database SQLite sono un formato di archiviazione raccomandato dalla Biblioteca del Congresso degli Stati Uniti.

SQLite è una libreria compatta. Con tutte le funzionalità abilitate, la dimensione della libreria può essere inferiore a 750KB, a seconda della piattaforma di destinazione e delle impostazioni di ottimizzazione del compilatore.

Nel nostro progetto, abbiamo utilizzato, per integrare SQLite, il modulo `sqlite3`, scritto da Gerhard Häring. Come prima cosa abbiamo creato il database e aperto una connessione con esso attraverso il comando `self.conn = sqlite3.connect(db_name)`. Dopo questa fase, abbiamo potuto subito effettuare delle query, in particolare quelle che non restituiscono dati, richiamando `self.conn.execute(query)`. Per le query, invece, che restituiscono dei dati, come, ad esempio, `SELECT`, abbiamo dovuto fare prima un passaggio intermedio, ovvero istanziare un cursore attraverso `self.conn.cursor()`; una volta istanziato il cursore, lo abbiamo utilizzato per recuperare i dati necessari, con i comandi `cursor.execute(query)` e `cursor.fetchall()`. L'ultima cosa da fare, dopo ogni query di scrittura, è salvare le modifiche in modo permanente: abbiamo utilizzato, per fare questo, `self.conn.commit()`.

Nel Listato 4.3 vediamo, con un esempio pratico, come viene utilizzata la libreria `sqlite3` per la gestione del database e le interazioni con esso.

```
1  def __init__(self, db_name='db_edilizia.db'):  
2      # Si stabilisce la connessione con il db  
3      self.conn = sqlite3.connect(db_name)  
4      # Si attivano i vincoli di integrità referenziale  
5      self.conn.execute("PRAGMA_foreign_keys_=_ON;")  
6      # Chiamata del metodo create_table_fornitori()  
7      self.create_table_fornitori()  
8  
9      # Metodo per la creazione della tabella  
10     def create_table_fornitori(self):  
11         query = '''  
12             CREATE TABLE IF NOT EXISTS fornitori (  
13                 nome TEXT PRIMARY KEY,  
14                 titolare TEXT NOT NULL,  
15                 attivita TEXT NOT NULL,  
16                 residenza TEXT NOT NULL,  
17                 email TEXT NOT NULL,  
18                 telefono TEXT NOT NULL  
19             )  
20         '''  
21         self.conn.execute(query)  
22         self.conn.commit()  
23  
24     Metodo per aggiungere una nuova istanza  
25     def aggiungi_fornitore(self, nome, titolare, attivita, residenza, email,  
26         telefono):  
27         query = '''  
28             INSERT INTO fornitori (nome, titolare, attivita, residenza, email,  
29                 telefono)  
30             VALUES (?, ?, ?, ?, ?, ?)  
31         '''  
32         self.conn.execute(query, (nome, titolare, attivita, residenza, email,  
33             telefono))  
34         self.conn.commit()  
35  
36     # Metodo per la modifica di una nuova istanza  
37     def modifica_fornitore(self, nome, titolare, attivita, residenza, email,  
38         telefono, nome_vecchio):  
39         query = '''  
40             UPDATE fornitori  
41             SET nome = ?, titolare = ?, attivita = ?, residenza = ?, email = ?,  
42                 telefono = ?  
43             WHERE nome = ?  
44         '''  
45         self.conn.execute(query, (nome, titolare, attivita, residenza, email,  
46             telefono, nome_vecchio))  
47         self.conn.commit()
```

```
42
43 # Metodo per ottenere tutte le istanze
44 def get_all_fornitori(self):
45     cursor = self.conn.cursor()
46     cursor.execute('''SELECT * FROM fornitori''')
47     return cursor.fetchall()
48
49 # Metodo per ottenere una particolare istanza dal nome
50 def get_fornitore_by_nome(self, nome):
51     cursor = self.conn.cursor()
52     cursor.execute('''SELECT * FROM fornitori WHERE nome = ?''', (nome,))
53     return cursor.fetchall()
54
55 # Metodo per ottenere tutte le istanze con una certa attivita'
56 def get_fornitori_by_attivita(self, attivita):
57     cursor = self.conn.cursor()
58     cursor.execute('''SELECT * FROM fornitori WHERE attivita = ?''', (
59         attivita, ))
60     return cursor.fetchall()
61
62 # Metodo per eliminare un'istanza, dato il nome in input
63 def delete_fornitore_by_nome(self, nome):
64     cursor = self.conn.cursor()
65     cursor.execute('''DELETE FROM fornitori WHERE nome = ?''', (nome,))
66     self.conn.commit()
67
68 # Metodo per svuotare tutta la tabella
69 def delete_all(self):
70     cursor = self.conn.cursor()
71     cursor.execute('''DELETE FROM fornitori''')
72     self.conn.commit()
```

Listing 4.3: Esempio di utilizzo di sqlite3

4.1.5 PyQt5

GUI vs terminale

Molte persone non giudicano un libro dalla sua copertina: sono interessate al contenuto, non all'apparenza. Al contrario, molti utenti di software giudicano il prodotto dalla sua interfaccia dedicata all'utente. Esiste in Python, così come in altri linguaggi di programmazione, la possibilità di implementare programmi che girano sul terminale. Questo tipo di interfaccia è adeguato per alcune applicazioni, oltre che più semplice da utilizzare; tuttavia, nella maggior parte dei casi, gli utilizzatori di computer nel mondo non vedono mai un'interfaccia come questa, perché la maggior parte del software interattivo adotta un'interfaccia grafica per l'utente (Graphical User Interface, GUI). Una GUI visualizza tutte le informazioni per l'utente, compreso il testo, in modo grafico, consentendone la manipolazione in modo diretto con un dispositivo di puntamento.

Un programma dotato di GUI è guidato dagli eventi, cioè rimane inattivo fino a quando l'utente usa il mouse per premere un pulsante grafico o selezionare un'opzione di un menù. Al contrario, un programma con interfaccia basata su terminale mantiene costantemente il controllo delle interazioni con l'utente. Detto in modo diverso, un programma basato su terminale chiede all'utente di fornire il successivo dato in ingresso, mentre un programma basato su GUI delega il controllo all'utente, consentendo l'inserimento di dati in qualsiasi ordine e attendendo la pressione di un pulsante o la selezione di un'opzione dal menù.

Per tutta questa serie di motivi, uno dei requisiti non funzionali del sistema è l'implementazione delle interfacce grafiche, in modo che l'utente possa interagire con esse durante l'esecuzione del software.

PyQt5

Nel nostro progetto, abbiamo utilizzato, per l'implementazione delle GUI, PyQt5, un insieme di classi (oltre 1000) che consentono l'accesso alle librerie grafiche Qt. PyQt5 supporta le piattaforme Windows, Linux, UNIX, Android, MacOS e iOS. Tutte le classi messe a disposizione da PyQt5 sono raggruppate in circa 50 moduli; di questi, per sviluppare le nostre GUI, ne abbiamo utilizzato principalmente tre:

- `QtCore`: contiene le classi base di PyQt, quali le astrazioni per le animazioni, macchine a stati, `thread`, file, memoria condivisa, espressioni regolari;
- `QtGui`: fornisce un insieme di classi per la gestione delle finestre, immagini, font e testi; da usare congiuntamente al modulo `QtWidgets`;
- `QWidget`: contiene una vasta gamma di widget (form, pulsanti, label, treeview, tabelle, checkbox e quant'altro) adatti alla creazione di interfacce grafiche desktop.

QtCore

Nella Tabella 4.1 vediamo le principali classi utilizzate, appartenenti al modulo `QtCore`.

Classe	Descrizione	Esempio
<code>Qt</code>	Insieme di costanti necessarie per programmare in modo efficiente con PyQt5	<code>outer_layout.setAlignment(Qt.AlignCenter)</code>
<code>QDate</code>	Mette a disposizione un set di metodi per la gestione delle date	<code>self.data_stipula_input.setDate(QDate.currentDate())</code>
<code>QRegExp</code>	È una classe con cui definire pattern complessi per lavorare con testo; essa segue un formato ben definito, come, ad esempio, numeri di telefono	<code>self.telefono_input.setValidator(QRegExpValidator(QRegExp("[0-9]+")))</code>

Tabella 4.1: Classi del modulo `QtCore` utilizzate nel progetto

QtGui

Nella Tabella 4.2 vediamo le principali classi utilizzate, appartenenti al modulo `QtGui`.

QtWidget

Nella Tabella 4.3 vediamo le principali classi utilizzate, appartenenti al modulo `QWidget`.

Costruzione della "Home Page"

Nel Listato 4.4 vediamo come viene costruita un'interfaccia grafica utilizzando gli strumenti che PyQt5 mette a disposizione. La classe utilizzata come esempio è `vistaHome`, ovvero la Home Page del sistema.

Classe	Descrizione	Esempio
QFont	Rappresenta le proprietà di un font utilizzato per il rendering del testo (o di altre componenti grafiche) in un'applicazione	<code>self.conferma_button.setFont(QFont("Times Roman", 11, QFont.Bold))</code>
QPixmap	È utilizzata per rappresentare immagini in applicazioni grafiche	<code>self.logo.setPixmap(QPixmap("images\logo_splash.png"))</code>
QIcon	È utilizzata per rappresentare icone in pulsanti, menù o finestre	<code>self.setWindowIcon(QIcon("images\logo_senza_nome.png"))</code>
QStandardItemModel	Fornisce un modello generico per l'immagazzinamento di dati personalizzati nelle QListView	<code>model = QStandardItemModel(), self.list_view.setModel(model)</code>
QStandardItem	Mette a disposizione l'elemento cardine da utilizzare con la classe QStandardItemModel	<code>item = QStandardItem(f"{d[1]} {d[2]}, Matricola N. {d[0]}")</code>
QRegExpValidator	È utilizzata per validare l'input dell'utente in base a un'istanza di QRegExp	<code>self.telefono_input.setValidator(QRegExpValidator(QRegExp("[0-9]+")))</code>

Tabella 4.2: Classi del modulo QtGui utilizzate nel progetto

```

1 class HomePage(QMainWindow):
2     def __init__(self):
3         super().__init__()
4
5         # Codice per costruire la barra del menu in alto con i comandi rapidi
6         self.menu_bar = self.menuBar()
7         self.menu_bar.setStyleSheet("background-color:_rgb(240,_240,_240)")
8         self.config_menubar("File", QIcon("images\exit.png"), "Exit", 'Ctrl+Q').
9             triggered.connect(self.close)
10        self.config_menubar("Clienti", QIcon("images\clienti.jpg"), "Lista_
11            clienti", 'Ctrl+W').triggered.connect(self.clienti)
12        self.config_menubar("Veicoli", QIcon("images\veicoli.png"), "Lista_
13            veicoli", 'Ctrl+E').triggered.connect(self.veicoli)
14        self.config_menubar("Fornitori", QIcon("images\fornitori.jpg"), "
15            Visualizza_fornitori", 'Ctrl+R').triggered.connect(self.fornitori)
16        self.config_menubar("Statistiche", QIcon("images\statistiche.jpg"), "
17            Visualizza_statistiche", 'Ctrl+T').triggered.connect(self.statistiche
18            )
19        self.config_menubar("Tutorial", QIcon("images\tutorial.png"), "
20            Visualizza_tutorial", 'Ctrl+Y').triggered.connect(self.tutorial)
21
22        # Imposta l'icona della finestra
23        self.setWindowIcon(QIcon("images\logo_senza_nome.png"))
24
25        self.initUI()
26
27    def initUI(self):
28        # Imposta titolo e misure della finestra
29        self.setWindowTitle('EdilGest')
30        self.setGeometry(0, 0, 750, 650)
31
32        central_widget = QWidget()
33        self.setCentralWidget(central_widget)
34
35        outer_layout = QVBoxLayout(central_widget)
36
37        main_layout = QVBoxLayout()

```

Classe	Descrizione	Esempio
QApplication	Classe che gestisce il flusso della UI e l'inizializzazione della nostra applicazione; ad ogni applicazione con UI, corrisponde un oggetto QApplication	app = QApplication(sys.argv)
QSplashScreen	Fornisce uno "splashscreen" da mostrare durante l'avvio dell'applicazione, per rendere più fluida la user experience	splash = QSplashScreen(QPixmap("images/logo_splash.png"), Qt.WindowStaysOnTopHint) splash.show()
QMainWindow	Fornisce la finestra principale dell'applicazione; tutte le classi view ereditano da questa classe	class vistaAccedi(QMainWindow):
QWidget	Widget contenitore per layout e altri widget; necessario aggiungere un'istanza di QWidget a QMainWindow con setCentralWidget(widget)	central_widget = QWidget() self.setCentralWidget(central_widget)
QStackedWidget	Mette a disposizione una pila di widget visibili solo uno alla volta	self.stacked_widget = QStackedWidget() self.stacked_widget.addWidget(dipendenti_widget) self.stacked_widget.addWidget(contratto_widget)
QBoxLayout	Permette di organizzare i widget in modo ordinato in una riga o in una colonna	button_layout = QHBoxLayout() button_layout.addWidget(self.login_button) button_layout.addWidget(self.change_credentials_button)
QGridLayout	Come sopra, ma in una griglia	button_layout = QGridLayout() insert_button_in_grid(self.button_dipendenti, button_layout, 1, 1) insert_button_in_grid(self.button_lavori, button_layout, 1, 2) insert_button_in_grid(self.button_magazzino, button_layout, 2, 1) insert_button_in_grid(self.button_sicurezza, button_layout, 2, 2)
QFormLayout	Permette di aggiungere una riga (etichetta e campo) ad una form	form_layout = QFormLayout() self.addFieldToForm(self.username_label, self.username_input, form_layout) self.addFieldToForm(self.password_label, self.password_input, form_layout)
QLabel	Rappresenta un semplice testo immutabile	welcome_label = QLabel('Completa i seguenti campi')
QLineEdit	Rappresenta un editor di testo a una riga	self.matricola_input = QLineEdit(str(self.controller.get_last_matricola()))
QComboBox	Permette la selezione di un'opzione da un elenco a discesa	self.classificazione_input = QComboBox() self.classificazione_input.addItem(self.controller.get_tipi())
QDateEdit	Serve per gestire gli oggetti QDate	self.data_nascita_input = QDateEdit() self.data_nascita_input.setDate(QDate.currentDate())
QPushButton	Rappresenta un pulsante	self.login_button = QPushButton('Accedi') self.login_button.clicked.connect(self.accedi)
QCompleter	Fornisce suggerimenti di completamento automatico mentre l'utente digita	options = self.controller.get_nomi_mp() completer = QCompleter(options) line_edit.setCompleter(completer)
QCheckBox	Rappresenta una checkbox; nel nostro progetto è stata utilizzata per implementare lo switch tra due widget sovrapposti	Listato 4.7
QListView	Permette di visualizzare una lista di elementi e di recuperare quello selezionato	self.list_view = QListView() selected_index = self.list_view.selectedIndexes()[0]
QMessageBox	Permette di fornire informazioni all'utente (es. errori), o fare delle domande ("Sicuro sì/no")	QMessageBox.warning(self, "Nessuna selezione", "Nessun item è stato selezionato.")
QInputDialog	Mette a disposizione un dialog dove è possibile inserire un testo	text, ok = QInputDialog.getText(self, 'EdilGest', 'Eventuali dettagli pagamento:')
QMenu	Fornisce un menù	menuSicurezza = QMenu() menuSicurezza.addAction("Corsi di sicurezza sul lavoro", self.show_corsi) menuSicurezza.addAction("Visite mediche", self.show_visite) self.button_sicurezza.setMenu(menuSicurezza)

Tabella 4.3: Classi del modulo Qt utilizzate nel progetto

```

31     main_layout.setSpacing(50)
32     main_layout.setAlignment(Qt.AlignCenter)
33
34     # Label di benvenuto
35     welcome_label = QLabel("Benvenuto_in_EdilGest.\nPregho_selezionare_un'
36         opzione_dal_menu")
37     insert_label(welcome_label, main_layout)
38
39     # Griglia dei pulsanti
40     button_layout = QGridLayout()
41     button_layout.setSpacing(50)
42
43     # Gestione pulsanti
44     self.button_dipendenti = QPushButton('Dipendenti')
45     insert_button_in_grid(self.button_dipendenti, button_layout, 1, 1)

```

```

45     self.button_dipendenti.clicked.connect(self.show_dipendenti)
46
47     self.button_lavori = QPushButton('Lavori_e_preventivi')
48     insert_button_in_grid(self.button_lavori, button_layout, 1, 2)
49
50     # Menu pulsante
51     menuLavori = QMenu()
52     menuLavori.addAction("Lavori", self.show_lavori)
53     menuLavori.addAction("Preventivi", self.show_preventivi)
54     self.button_lavori.setMenu(menuLavori)
55
56     self.button_magazzino = QPushButton('Magazzino')
57     insert_button_in_grid(self.button_magazzino, button_layout, 2, 1)
58     self.button_magazzino.clicked.connect(self.show_magazzino)
59
60     self.button_sicurezza = QPushButton('Sicurezza')
61     insert_button_in_grid(self.button_sicurezza, button_layout, 2, 2)
62
63     menuSicurezza = QMenu()
64     menuSicurezza.addAction("Corsi_di_sicurezza_sul_lavoro", self.show_corsi)
65     menuSicurezza.addAction("Visite_mediche", self.show_visite)
66     self.button_sicurezza.setMenu(menuSicurezza)
67
68     self.button_locali = QPushButton('Locali')
69     insert_button_in_grid(self.button_locali, button_layout, 3, 1)
70     self.button_locali.clicked.connect(self.show_locali)
71
72     self.button_contabilita = QPushButton('Contabilita')
73     insert_button_in_grid(self.button_contabilita, button_layout, 3, 2)
74
75     menuContabilita = QMenu()
76     menuContabilita.addAction("Entrate", self.show_entrata)
77     menuContabilita.addAction("Uscite", self.show_uscite)
78     menuContabilita.addAction("Bilanci", self.show_bilanci)
79     self.button_contabilita.setMenu(menuContabilita)
80
81     main_layout.addLayout(button_layout)
82
83     outer_layout.addLayout(main_layout)
84
85     self.center()
86
87     # Altre funzioni
88
89     # Funzione per impostare le opzioni della barra del menu in alto
90     def config_menubar(self, str, img, _str2, tasti):
91         self.menu_def = self.menu_bar.addMenu(str)
92         icon = QIcon(img)
93         action = QAction(icon, _str2, parent=self)
94         action.setShortcut(tasti)
95         self.menu_def.addAction(action)
96         return action
97
98     # Funzione per centrare la finestra
99     def center(self):
100         qr = self.frameGeometry()
101         cp = QDesktopWidget().availableGeometry().center()
102         qr.moveCenter(cp)
103         self.move(qr.topLeft())

```

Listing 4.4: Costruzione della "Home Page" del sistema

Costruzione "Vista Lista Fornitori"

Vediamo anche, nel Listato 4.5, come viene costruita un'interfaccia in cui è possibile visualizzare una lista di oggetti del sistema. Prendiamo in esame la classe `vistaListaFornitori`.

```
1 def initUI(self):
2     self.setWindowTitle('EdilGest')
3     self.setGeometry(0, 0, 750, 650)
4
5     central_widget = QWidget()
6     self.setCentralWidget(central_widget)
7
8     outer_layout = QVBoxLayout(central_widget)
9
10    main_layout = QVBoxLayout()
11    main_layout.setSpacing(20)
12    main_layout.setAlignment(Qt.AlignCenter) # Centra verticalmente
13
14    label = QLabel("Lista_fornitori")
15    insert_label(label, main_layout)
16
17    # Istanziata la lista
18    self.list_view = QListView()
19    self.list_view.setFixedSize(500, 400)
20    self.list_view.setStyleSheet(stile_liste())
21    # Popolata con i dati di interesse
22    self.genera_lista()
23    main_layout.addWidget(self.list_view, alignment=Qt.AlignCenter)
24
25    button_layout = QHBoxLayout()
26    button_layout.setSpacing(10)
27    button_layout.setAlignment(Qt.AlignCenter) # Centra orizzontalmente
28
29    self.aggiungi = QPushButton("Aggiungi")
30    insert_button(self.aggiungi, button_layout)
31    self.aggiungi.clicked.connect(self.start_aggiunta)
32
33    self.elimina = QPushButton("Elimina_tutto")
34    insert_button(self.elimina, button_layout)
35    self.elimina.clicked.connect(self.on_button_elimina_clicked)
36
37    self.modifica = QPushButton("Modifica")
38    insert_button(self.modifica, button_layout)
39    self.modifica.clicked.connect(self.on_button_modifica_clicked)
40
41    self.contatta = QPushButton("Contatta")
42    insert_button(self.contatta, button_layout)
43    self.contatta.clicked.connect(self.contatta_click)
44
45    self.button_filtro = QPushButton("Filtri")
46    insert_button(self.button_filtro, button_layout)
47
48    menu_filtri = QMenu()
49    menu_filtri.addAction("Cerca_per_tipologia", self.
50        on_button_filtro_tipo_clicked)
51    menu_filtri.addAction("Cerca_per_nome", self.
52        on_button_filtro_nome_clicked)
53    menu_filtri.addAction("Resetta_filtro", self.on_button_reset_clicked)
54    self.button_filtro.setMenu(menu_filtri)
55
56    main_layout.addLayout(button_layout)
57
58    outer_layout.addLayout(main_layout)
```

```

57
58     self.center()
59
60     # Funzione per popolare una lista
61     def genera_lista(self):
62         # Variabili per il filtraggio
63         self.nome_filtro = ''
64         self.tipo_filtro = ''
65
66         # Modello degli item
67         model = QStandardItemModel()
68         for f in self.controller.get_all_fornitori():
69             # Singolo item
70             item = QStandardItem(f[0] + "_di_" + f[1])
71             item.setEditable(False)
72             item.setFont(QFont("Times_Roman", 11))
73             # Aggiunge l'item al modell
74             model.appendRow(item)
75         # Assegna il modello alla lista
76         self.list_view.setModel(model)

```

Listing 4.5: Costruzione di un'interfaccia per la visualizzazione di liste

Costruzione "Vista Inserisci Fornitore"

Osserviamo ancora, nel Listato 4.6, la costruzione di un'interfaccia in cui l'utente può inserire un nuovo oggetto nel sistema. Prendiamo in considerazione, in questo caso, la classe `vistaInserisciFornitore`.

Le schermate per la visualizzazione dei dettagli di un oggetto del sistema sono costruite allo stesso modo, con la differenza che le edit text vengono popolate con i valori desiderati; inoltre, ovviamente, sono presenti pulsanti diversi.

```

1     def initUI(self):
2         self.setWindowTitle('EdilGest')
3         self.setGeometry(100, 100, 750, 650)
4
5         central_widget = QWidget()
6         self.setCentralWidget(central_widget)
7
8         outer_layout = QVBoxLayout(central_widget)
9
10        main_layout = QVBoxLayout()
11        main_layout.setSpacing(20)
12        main_layout.setAlignment(Qt.AlignCenter) # Centra verticalmente
13
14        welcome_label = QLabel('Completa_i_seguenti_campi')
15        welcome_label.setAlignment(Qt.AlignCenter)
16        welcome_label.setFont(QFont("Times_Roman", 20, QFont.Bold))
17        welcome_label.setStyleSheet("color:_blue")
18        main_layout.addWidget(welcome_label, alignment=Qt.AlignCenter)
19
20        # Creazione del form
21        form_layout = QFormLayout()
22        form_layout.setSpacing(10)
23
24        form_container = QVBoxLayout()
25        form_container.addLayout(form_layout)
26        form_container.setContentsMargins(150, 0, 150, 0)
27
28        # Etichetta

```

```
29     self.nome_label = QLabel('Nome')
30     # Elemento di input
31     self.nome_input = QLineEdit()
32     # Aggiunto al form
33     self.addFieldToForm(self.nome_label, self.nome_input, form_layout)
34
35     self.titolare_label = QLabel('Titolare')
36     self.titolare_input = QLineEdit()
37     # Validatore: fa in modo che l'utente possa inserire solo certi caratteri
38     # ; in questo caso solo lettere minuscole e maiuscole
39     self.titolare_input.setValidator(QRegExpValidator(QRegExp("[a-zA-Z- ]+")))
40     self.addFieldToForm(self.titolare_label, self.titolare_input, form_layout)
41
42     self.attivita_label = QLabel('Attivita')
43     self.attivita_input = QComboBox()
44     self.attivita_input.addItem(self.controller.get_lista_attivita())
45     self.addFieldToForm(self.attivita_label, self.attivita_input, form_layout)
46
47     self.residenza_label = QLabel('Residenza')
48     self.residenza_input = QLineEdit()
49     self.addFieldToForm(self.residenza_label, self.residenza_input,
50                        form_layout)
51
52     self.email_label = QLabel('Email')
53     self.email_input = QLineEdit()
54     self.addFieldToForm(self.email_label, self.email_input, form_layout)
55
56     self.telefono_label = QLabel('Telefono')
57     self.telefono_input = QLineEdit()
58     # Validatore: fa in modo che l'utente possa inserire solo numeri
59     self.telefono_input.setValidator(QRegExpValidator(QRegExp("[0-9]+")))
60     self.addFieldToForm(self.telefono_label, self.telefono_input, form_layout)
61
62     main_layout.addLayout(form_container)
63
64     button_layout = QHBoxLayout()
65     button_layout.setSpacing(10)
66     button_layout.setAlignment(Qt.AlignCenter)
67
68     self.conferma_button = QPushButton('Conferma')
69     insert_button(self.conferma_button, button_layout)
70     self.conferma_button.clicked.connect(self.on_conferma_button_clicked)
71
72     main_layout.addLayout(button_layout)
73
74     outer_layout.addLayout(main_layout)
75
76     self.center()
77
78     # Metodo per inserire etichetta e campo nel form
79     def addFieldToForm(self, label, input, form):
80         input.setStyleSheet(
81             "border-radius: 10px; border: 2px solid blue; color: black; padding: 5px")
82         label.setFont(QFont("Times_Roman", 11, QFont.Bold))
83         label.setStyleSheet("color: blue")
84         form.addRow(label, input)
```

Listing 4.6: Costruzione di un'interfaccia per l'inserimento di nuovi oggetti**Costruzione "Vista Inserisci Dipendente"**

Meritano una menzione le schermate in cui l'utente deve compilare due form; la gestione di queste interfacce è stata fatta attraverso uno switch che permette di passare da una sezione all'altra. Prendiamo, come esempio, la classe `vistaInserisciDipendente` (le altre vengono costruite allo stesso modo) e mostriamo, nel Listato 4.7, come essa sia stata costruita; in questo caso, lo switch permette di passare dalla sezione "Dati anagrafici" a quella "Dati contratto".

```

1     def initUI(self):
2         self.setWindowTitle('EdilGest')
3         self.setGeometry(0, 0, 750, 650)
4
5         central_widget = QWidget()
6         self.setCentralWidget(central_widget)
7
8         outer_layout = QVBoxLayout(central_widget)
9         outer_layout.setAlignment(Qt.AlignCenter) # Centra verticalmente
10
11        welcome_label = QLabel('Completa i seguenti campi')
12        insert_label(welcome_label, outer_layout)
13
14        # Layout dello switch
15        switcher_layout = QHBoxLayout()
16        switcher_layout.setAlignment(Qt.AlignCenter)
17
18        self.anagrafica_label = QLabel("Anagrafica")
19        self.anagrafica_label.setFont(QFont("Times_Roman", 11, QFont.Bold))
20        self.anagrafica_label.setStyleSheet("color:_blue")
21
22        # Creato lo switcher
23        self.section_switcher = QCheckBox()
24        self.section_switcher.setFixedSize(60, 30)
25        self.section_switcher.setStyleSheet(stile_checkbox())
26        # Quando cambia, chiama questa funzione self.section_switcher.
27           stateChanged.connect(self.switch_section)
28
29        self.contratto_label = QLabel("Contratto")
30        self.contratto_label.setFont(QFont("Times_Roman", 11, QFont.Bold))
31        self.contratto_label.setStyleSheet("color:_blue")
32
33        # Aggiunte etichette e switcher al layout
34        switcher_layout.addWidget(self.anagrafica_label)
35        switcher_layout.addWidget(self.section_switcher)
36        switcher_layout.addWidget(self.contratto_label)
37
38        switcher_container = QVBoxLayout()
39        switcher_container.addLayout(switcher_layout)
40        switcher_container.setAlignment(Qt.AlignCenter)
41
42        outer_layout.addLayout(switcher_container)
43
44        # Contenitore delle due sezioni
45        self.stacked_widget = QStackedWidget()
46        self.stacked_widget.setFixedWidth(600) # Imposta dimensioni fisse per il
           QStackedWidget

```

```
47     # Centra il QStackedWidget orizzontalmente e verticalmente
48     stacked_container = QVBoxLayout()
49     stacked_container.addWidget(self.stacked_widget, alignment=Qt.AlignCenter
50     )
51     stacked_container.setAlignment(Qt.AlignCenter)
52
53     outer_layout.addLayout(stacked_container)
54
55     # Sezione "Dati anagrafici"
56     dipendenti_widget = QWidget()
57     self.init_dipendenti_ui(dipendenti_widget)
58     self.stacked_widget.addWidget(dipendenti_widget)
59
60     # Sezione "Contratto"
61     contratto_widget = QWidget()
62     self.init_contratto_ui(contratto_widget)
63     self.stacked_widget.addWidget(contratto_widget)
64
65     self.conferma_button = QPushButton('Conferma')
66     insert_button(self.conferma_button, outer_layout)
67     self.conferma_button.clicked.connect(self.on_conferma_button_clicked)
68
69     self.center()
70
71     # Layout sezione "Dati anagrafici"
72     def init_dipendenti_ui(self, widget):
73         main_layout = QVBoxLayout(widget)
74         main_layout.setSpacing(20)
75         main_layout.setAlignment(Qt.AlignCenter) # Centra verticalmente
76
77         form_layout = QFormLayout()
78         form_layout.setSpacing(10)
79
80         form_container = QVBoxLayout()
81         form_container.addLayout(form_layout)
82
83         self.matricola_label = QLabel('Matricola')
84         self.matricola_input = QLineEdit(str(self.controller.get_last_matricola()
85         ))
86         self.matricola_input.setReadOnly(True)
87         self.addFieldToForm(self.matricola_label, self.matricola_input,
88         form_layout)
89
90         self.nome_label = QLabel('Nome')
91         self.nome_input = QLineEdit()
92         self.nome_input.setValidator(QRegExpValidator(QRegExp("[a-zA-Z-_]+")))
93         self.addFieldToForm(self.nome_label, self.nome_input, form_layout)
94
95         self.cognome_label = QLabel('Cognome')
96         self.cognome_input = QLineEdit()
97         self.cognome_input.setValidator(QRegExpValidator(QRegExp("[a-zA-Z-_]+")))
98         self.addFieldToForm(self.cognome_label, self.cognome_input, form_layout)
99
100         self.cs_label = QLabel('Codice_fiscale')
101         self.cs_input = QLineEdit()
102         self.addFieldToForm(self.cs_label, self.cs_input, form_layout)
103
104         self.residenza_label = QLabel('Residenza')
105         self.residenza_input = QLineEdit()
106         self.addFieldToForm(self.residenza_label, self.residenza_input,
107         form_layout)
```

```
104
105     self.data_nascita_label = QLabel('Data_nascita')
106     self.data_nascita_input = QDateEdit()
107     self.data_nascita_input.setCalendarPopup(True)
108     self.data_nascita_input.setDisplayFormat("dd/MM/yyyy")
109     self.data_nascita_input.setDate(QDate.currentDate())
110     self.addFieldToForm(self.data_nascita_label, self.data_nascita_input,
111                          form_layout)
112
113     self.email_label = QLabel('Email')
114     self.email_input = QLineEdit()
115     self.addFieldToForm(self.email_label, self.email_input, form_layout)
116
117     self.telefono_label = QLabel('Telefono')
118     self.telefono_input = QLineEdit()
119     self.telefono_input.setValidator(QRegExpValidator(QRegExp("[0-9]+")))
120     self.addFieldToForm(self.telefono_label, self.telefono_input, form_layout
121                          )
122
123     main_layout.addLayout(form_container)
124
125     # Layout sezione "Contratto"
126     def init_contratto_ui(self, widget):
127         main_layout = QVBoxLayout(widget)
128         main_layout.setSpacing(20)
129         main_layout.setAlignment(Qt.AlignCenter) # Centra verticalmente
130
131         form_layout = QFormLayout()
132         form_layout.setSpacing(10)
133
134         # Aggiungi margini laterali al form layout
135         form_container = QVBoxLayout()
136         form_container.addLayout(form_layout)
137
138         self.stipendio_label = QLabel('Stipendio')
139         self.stipendio_input = QLineEdit()
140         self.stipendio_input.setValidator(QRegExpValidator(QRegExp("
141             ^[0-9]*\.[0-9]*$" )))
142         self.addFieldToForm(self.stipendio_label, self.stipendio_input,
143                             form_layout)
144
145         self.ruolo_label = QLabel('Ruolo')
146         self.ruolo_input = QLineEdit()
147         self.addFieldToForm(self.ruolo_label, self.ruolo_input, form_layout)
148
149         self.data_stipula_label = QLabel('Data_Stipula')
150         self.data_stipula_input = QDateEdit()
151         self.data_stipula_input.setCalendarPopup(True)
152         self.data_stipula_input.setDisplayFormat("dd/MM/yyyy")
153         self.data_stipula_input.setDate(QDate.currentDate())
154         self.addFieldToForm(self.data_stipula_label, self.data_stipula_input,
155                             form_layout)
156
157         self.durata_label = QLabel('Durata')
158         self.durata_input = QLineEdit()
159         self.addFieldToForm(self.durata_label, self.durata_input, form_layout)
160
161         self.orario_label = QLabel('Orario')
162         self.orario_input = QLineEdit()
163         self.addFieldToForm(self.orario_label, self.orario_input, form_layout)
164
165         self.extra_label = QLabel('Extra')
```

```

161     self.extra_input = QLineEdit()
162     self.addFieldToForm(self.extra_label, self.extra_input, form_layout)
163
164     main_layout.addLayout(form_container)
165
166     # Logica dello switcher
167     def switch_section(self, state):
168         if state == Qt.Checked:
169             self.stacked_widget.setCurrentIndex(1)
170         else:
171             self.stacked_widget.setCurrentIndex(0)

```

Listing 4.7: Costruzione di un'interfaccia con switch tra due sezioni

Altri metodi utili per la costruzione di interfacce grafiche

Nella classe `funzioniUtili` si trovano altri metodi utilizzati per gestire l'interfaccia grafica; ne mostriamo alcuni nel Listato 4.8.

```

1 # Metodo che restituisce uno stile CSS da applicare alle liste
2 def stile_liste():
3     return """
4         QListView {
5             background-color: white; /* Colore di sfondo del widget */
6             border: 1px solid blue;
7             border-radius: 15px;
8         }
9
10
11        QListView::item {
12            background-color: white; /* Colore di sfondo degli elementi,
13            simula il padding bianco */
14            border-radius: 10px; /* Arrotonda gli angoli degli elementi
15            */
16            margin: 10px 10px 0 10px; /* Spazio tra gli elementi */
17            border: 1px solid blue; /* Bordo degli item */
18            padding: 5px; /* Spazio tra il contenuto e il bordo */
19        }
20
21        QListView::item:selected {
22            background-color: lightblue; /* Colore di sfondo dell'
23            elemento selezionato */
24            color: black; /* Colore del testo dell'elemento selezionato
25            */
26        }
27        """
28
29 # Metodo che restituisce uno stile CSS da applicare agli switcher
30 def stile_checkbox():
31     return """
32         QCheckBox {
33             background-color: blue;
34             border-radius: 15px;
35             padding: 2px;
36         }
37
38         QCheckBox::indicator {
39             width: 26px;
40             height: 26px;
41             border-radius: 13px;
42             background-color: white;
43             position: absolute;

```

```

39         }
40         QCheckBox::indicator:checked {
41             background-color: white;
42             left: 30px;
43         }
44         QCheckBox::indicator:unchecked {
45             background-color: white;
46             left: 2px;
47         }
48     """
49
50 # Metodo che inserisce i pulsanti nel layout
51 def insert_button(button, layout):
52     button.setFont(QFont("Times_Roman", 11, QFont.Bold))
53     button.setStyleSheet(
54         "background-color:_blue;_border-radius:_15px;_color:_white;_padding:_10px
55         "
56     )
57     layout.addWidget(button)
58
59 # Metodo che inserisce i pulsanti in una griglia
60 def insert_button_in_grid(button, layout, x, y):
61     button.setFont(QFont("Times_Roman", 11, QFont.Bold))
62     button.setStyleSheet(
63         "background-color:_blue;_border-radius:_15px;_color:_white;_padding:_10px
64         "
65     )
66     layout.addWidget(button, x, y)
67
68 # Metodo che inserisce le etichette
69 def insert_label(label, layout, color="color:_blue"):
70     label.setAlignment(Qt.AlignCenter)
71     label.setFont(QFont("Times_Roman", 20, QFont.Bold))
72     label.setStyleSheet(color)
73     layout.addWidget(label, alignment=Qt.AlignCenter)

```

Listing 4.8: Metodi utili per la costruzione di interfacce

4.1.6 Altre librerie

Nella Tabella 4.4 vediamo altri strumenti utilizzati nella fase di implementazione del codice.

Nel Listato 4.9 vediamo un esempio di utilizzo di `pathlib`.

```

1 def tutorial(self):
2     # Serve per recuperare il percorso in cui si trova il file tutorial.html
3     p = pathlib.Path().absolute()
4     p = str(p) + "\\images\\tutorial.html"
5     webbrowser.open(p)

```

Listing 4.9: Esempio di utilizzo di `pathlib`

Nel Listato 4.10 vediamo un esempio di utilizzo di `webbrowser`.

```

1 def contatta_click(self):
2     selected_index = self.list_view.selectedIndexes()
3
4     if not selected_index:
5         QMessageBox.warning(self, "Nessuna_selezione", "Nessun_item_ _stato_
6             selezionato.")

```

Libreria/Modulo	Descrizione	Esempio
pathlib	Fornisce una comoda interfaccia orientata agli oggetti per gestire i percorsi dei file e delle directory	Listato 4.9
webbrowser	Fornisce un'interfaccia di altissimo livello per consentire la visualizzazione di documenti web agli utenti	Listato 4.10
re	Fornisce operazioni di corrispondenza delle espressioni regolari	Listato 4.11
codicefiscale	Libreria per lavorare con i codici fiscali italiani	Listato 4.12
numpy	Fornisce funzioni ottimizzate per la creazione di array a partire da intervalli	Listato 4.13
pyplot	Sottolibreria di Matplotlib, utilizzata per creare grafici in modo semplice e rapido	Listato 4.14
QPdfWriter	Classe di Qt che permette di creare file PDF con contenuto generato dall'applicazione	Listato 4.15
QPainter	Fornisce le funzionalità per disegnare su widget immagini o altri oggetti grafici in Qt	Listato 4.15

Tabella 4.4: Descrizione di ulteriori librerie/moduli utilizzati nel progetto

```

7     else:
8         selected_item = selected_index[0].row()
9         # Costruzione dell'URL di Gmail con solo il destinatario
10        url = f"https://mail.google.com/mail/?view=cm&fs=1&to={self.controller.
            getDipendente(selected_item)[6]}"
11
12        # Apri l'URL nel browser predefinito
13        webbrowser.open(url)

```

Listing 4.10: Esempio di utilizzo di webbrowser

Nel Listato 4.11 vediamo un esempio di utilizzo di re.

```

1 # Definizione della regex per un'email valida
2 email_regex = r'^[a-zA-Z0-9_+.]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-]+$',
3
4 # Verifica se l'email corrisponde al pattern della regex
5 if not re.match(email_regex, email):
6     return 'Formato_email_non_valido!'

```

Listing 4.11: Esempio di utilizzo di re

Nel Listato 4.12 vediamo un esempio di utilizzo di codicefiscale.

```

1 if not codicefiscale.isvalid(codice_fiscale):
2     return 'Codice_fiscale_non_valido!'

```

Listing 4.12: Esempio di utilizzo di codicefiscale

Nel Listato 4.13 vediamo un esempio di utilizzo di numpy.

```

1 # Intervallo personalizzato sull'asse y
2 # Con np.arange generiamo una sequenza di numeri a partire da 0 fino a
   max_importo * 1.1, con un incremento di 200. Questo crea una lista di valori
   che vanno da 0 fino a poco oltre il massimo importo (circa il 10% in pi ),
   con passi di 200.

```

```
3 plt.yticks(np.arange(0, max_importo * 1.1, 200))
```

Listing 4.13: Esempio di utilizzo di numpy

Nel Listato 4.14 vediamo un esempio di utilizzo di pyplot.

```
1 def stats_lavori(self):
2     lista_lavori = self.controller.get_lavori()
3     importi = []
4     lavori = []
5
6     if not lista_lavori:
7         QMessageBox.warning(None, "EdilGest", "Nessun_lavoro_presente!")
8     else:
9         for lavoro in lista_lavori:
10            importi.append(float(lavoro[1]))
11            lavori.append(f"{lavoro[0]}")
12
13            index = np.arange(len(lavori))
14
15            plt.figure(figsize=(12, 6))
16            plt.plot(index, importi, marker='o', linestyle='-', color='blue')
17
18            plt.title('Importi_Lavori')
19            plt.xlabel('Lavori')
20            plt.ylabel('Importo_( )')
21
22            # Rotazione e allineamento delle etichette
23            plt.xticks(index, lavori, rotation=45, ha='right')
24
25            # Intervallo sull'asse y con margine extra
26            max_importo = float(self.controller.get_max_importo_lavori())
27            plt.ylim(0, max_importo * 1.1)
28
29            # Intervallo personalizzato sull'asse y
30            plt.yticks(np.arange(0, max_importo * 1.1, 200))
31
32            plt.grid(axis='y', linestyle='--', alpha=0.7)
33            plt.tight_layout()
34            plt.show()
```

Listing 4.14: Esempio di utilizzo di pyplot

Nel Listato 4.15 vediamo un esempio di utilizzo di QPdfWriter e di QPainter.

```
1 def save_as_pdf(self):
2     # Crea un PDF writer
3     pdf_writer = QPdfWriter(f"Contratto_{self.dipendente[0]}_{self.dipendente[1]}_
4         _{self.dipendente[2]}.pdf")
5     pdf_writer.setPageSize(QPdfWriter.A4) # Imposta il formato della pagina su
6         A4
7     pdf_writer.setResolution(300) # Imposta la risoluzione del PDF a 300 DPI
8
9     # Ottiene la dimensione della pagina A4 in punti (1 mm = 3.7795275591 punti)
10    page_width = 630 * 3.7795275591 # Larghezza della pagina A4 in punti
11    page_height = 891 * 3.7795275591 # Altezza della pagina A4 in punti
12
13    # Crea un QPainter per disegnare sul PDF
14    painter = QPainter(pdf_writer)
15    painter.setFont(QFont("Times_Roman", 12)) # Imposta il font a "Times Roman"
16        con dimensione 12
17
18    # Ottiene la dimensione del widget contenitore (stacked widget)
```

```
16 widget_size = self.stacked_widget.size()
17 widget_height = widget_size.height() # Altezza del widget
18 widget_width = widget_size.width() # Larghezza del widget
19
20 # Crea un QPixmap che il doppio dell'altezza del widget impilato
21 combined_pixmap = QPixmap(widget_width, 2 * widget_height)
22 combined_pixmap.fill(Qt.white) # Riempi il QPixmap con uno sfondo bianco
23
24 # Disegna la prima sezione (Preventivo) nella met superiore del pixmap
  combinato
25 self.stacked_widget.setCurrentIndex(0) # Seleziona la prima pagina del
  widget impilato
26 pixmap1 = QPixmap(widget_size) # Crea un QPixmap della dimensione del widget
27 self.stacked_widget.render(pixmap1) # Renderizza il widget sulla pixmap
28 painter_combined = QPainter(combined_pixmap)
29 painter_combined.drawPixmap(0, 0, pixmap1) # Disegna la prima parte del
  widget sul pixmap combinato
30
31 # Disegna la seconda sezione (Cliente) nella met inferiore del pixmap
  combinato
32 self.stacked_widget.setCurrentIndex(1) # Seleziona la seconda pagina del
  widget impilato
33 pixmap2 = QPixmap(widget_size) # Crea un altro QPixmap della dimensione del
  widget
34 self.stacked_widget.render(pixmap2) # Renderizza la seconda pagina sulla
  pixmap
35 painter_combined.drawPixmap(0, widget_height, pixmap2) # Disegna la seconda
  parte sul pixmap combinato
36
37 painter_combined.end() # Termina il disegno sul pixmap combinato
38
39 # Calcola il fattore di scala per adattare il pixmap combinato alla pagina A4
40 scale_x = page_width / combined_pixmap.width() # Fattore di scala
  orizzontale
41 scale_y = page_height / combined_pixmap.height() # Fattore di scala
  verticale
42 scale_factor = min(scale_x, scale_y) # Prende il fattore di scala minore per
  mantenere le proporzioni
43
44 # Scala e disegna il pixmap combinato nel PDF
45 scaled_pixmap = combined_pixmap.scaled(
46     combined_pixmap.size() * scale_factor, # Scala il pixmap mantenendo le
  proporzioni
47     Qt.KeepAspectRatio, # Mantieni il rapporto di aspetto
48     Qt.SmoothTransformation # Usa una trasformazione fluida per la scala
49 )
50
51 # Disegna il pixmap scalato nel PDF
52 painter.drawPixmap(0, 0, scaled_pixmap) # Disegna il pixmap scalato sulla
  pagina PDF
53 painter.end() # Termina il disegno sul PDF
54
55 self.stacked_widget.setCurrentIndex(0) # Ritorna alla prima sezione del
  widget impilato
56
57 # Mostra un messaggio di conferma all'utente
58 QMessageBox.information(self, "EdilGest", "PDF_salvato_correttamente!")
```

Listing 4.15: Esempio di utilizzo di QPdfWriter e di QPainter

4.1.7 HTML

Abbiamo deciso di implementare una sezione “Tutorial” dove viene mostrato un video in cui si fanno vedere tutte le funzionalità del sistema. Per fare ciò, abbiamo creato una semplice pagina web, in locale, che mostra il suddetto video. In generale, le pagine web si presentano come schermate dotate di una struttura grafica complessa e curata dal punto di vista estetico. Spesso sono corredate da elementi multimediali e, di norma, contengono numerosi collegamenti ad altre pagine dello stesso sito o di siti esterni. Nel nostro caso la situazione è molto più semplice, in quanto sono presenti soltanto una lista con i titoli delle sezioni e il video stesso. Tuttavia, anche per implementare la nostra semplice pagina web, ci siamo basati sulle istruzioni HTML (HyperText Markup Language), uno dei più famosi linguaggi di markup.

Nel Listato 4.16, possiamo vedere il metodo che gestisce il click sull’opzione “Tutorial” del menu-bar in alto nella Home Page.

```
1 def tutorial(self):
2     p = pathlib.Path().absolute()
3     p = str(p) + "\\images\\tutorial.html"
4     webbrowser.open(p)
```

Listing 4.16: Metodo per la visualizzazione del tutorial

Nel Listato 4.17 vediamo come viene costruita la pagina web (in locale) in cui viene mostrato all’utente il video tutorial del sistema. Abbiamo anche inserito uno script, scritto in JavaScript, che permette all’utente di saltare alla sezione desiderata attraverso un click sulla rispettiva etichetta. Inoltre, nella head, sono presenti anche alcune righe di codice CSS per dare uno stile alla pagina.

```
1 <!DOCTYPE html>
2 <html lang="it">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, _initial-scale=1.0">
6     <title>Video tutorial</title>
7     <!-- Stile CSS -->
8     <style type="text/css">
9         h1 { margin-top: 30px }
10        div { display: flex; flex-direction: row; justify-content: center; align-
11            items: center }
12        ul { margin-right: 50px }
13        video { margin-left: 50px }
14        li { cursor: pointer; color: blue; text-decoration: underline; }
15    </style>
16 </head>
17 <body>
18 <h1 align="center">Video tutorial per l'utilizzo di EdilGest</h1>
19 <div>
20     <!-- Lista di sezioni del video -->
21     <ul>
22         <!-- Item -->
23         <li data-time="5">0.05 - autenticazione e cambio credenziali</li>
24         <li data-time="60">1.00 - gestione dipendenti</li>
25         <li data-time="120">2.00 - gestione lavori</li>
26         <li data-time="180">3.00 - gestione preventivi</li>
27         <li data-time="240">4.00 - gestione clienti</li>
28         <li data-time="300">5.00 - gestione magazzino</li>
29         <li data-time="360">6.00 - gestione locali</li>
30         <li data-time="420">7.00 - gestione veicoli</li>
31         <li data-time="480">8.00 - gestione visite mediche</li>
32         <li data-time="540">9.00 - gestione corsi di sicurezza sul lavoro</li>
```

```
32     <li data-time="600">10.00 - gestione entrate</li>
33     <li data-time="660">11.00 - gestione uscite</li>
34     <li data-time="720">12.00 - gestione bilanci</li>
35     <li data-time="780">13.00 - gestione fornitori</li>
36     <li data-time="840">14.00 - statistiche</li>
37 </ul>
38 <!-- Video vero e proprio -->
39 <video id="videoTutorial" width="480" height="480" controls>
40     <source src="Video/video_tutorial.mp4" type="video/mp4">
41     Your browser does not support the video tag.
42 </video>
43 </div>
44
45 <!-- Script che rende gli item cliccabili, in modo che l'utente possa passare a
46     una qualsiasi sezione del video tutorial -->
47 <script>
48     // Seleziona tutti gli elementi <li> e aggiungi l'evento click
49     document.querySelectorAll('ul li').forEach(item => {
50         item.addEventListener('click', function() {
51             const time = this.getAttribute('data-time');
52             const video = document.getElementById('videoTutorial');
53             video.currentTime = time; // Salta al tempo corrispondente
54             video.play(); // Riproduce il video
55         });
56     });
57 </script>
58 </body>
59 </html>
```

Listing 4.17: Pagina web per la visualizzazione del tutorial

4.2 Testing

In quest'ultima sezione ci occupiamo del test del software. L'obiettivo di questa fase è dimostrare che il nostro programma svolge i compiti per i quali è stato realizzato e identificare eventuali errori prima di metterlo in uso. Esistono tre tipi di test:

- *test di sviluppo*: includono tutte le attività di test che sono svolte dal team di sviluppo; comprendono test sulle unità, test dei componenti e test del sistema;
- *test della release*: è il processo che testa una particolare release di un sistema che dovrà essere utilizzata all'esterno del team di sviluppo;
- *test degli utenti*: fase del processo di test di un sistema in cui gli utenti o i clienti forniscono i loro input e i loro suggerimenti sui test del sistema; ce ne sono di tre tipi, ovvero alpha test, beta test e test di accettazione.

In questa fase, ci concentriamo sul testing dei componenti, utilizzando il framework di automazione dei test PyUnit. Abbiamo sviluppato quindici test, che verificano il corretto funzionamento di altrettante funzionalità del nostro programma.

4.2.1 Test sui lavori

Nel Listato 4.18 vediamo come abbiamo implementato i test per verificare la corretta esecuzione di alcune funzionalità relative alla sezione "Lavori".

```
1 def test_aggiungi_lavoro(self):
2     # Testa che le istanze non siano nulle
3     self.assertIsNotNone(self.cliente)
4     self.assertIsNotNone(self.lavoro)
5     self.assertIsNotNone(self.cliente2)
6     self.assertIsNotNone(self.lavoro2)
7     self.assertIsNotNone(self.lavoro3)
8
9     # Effettua l'inserimento
10    self.controller_clienti.aggiungi_cliente(*self.cliente)
11    self.controller_lavori.aggiungi_lavoro(*self.lavoro)
12    self.controller_clienti.aggiungi_cliente(*self.cliente2)
13    self.controller_lavori.aggiungi_lavoro(*self.lavoro2)
14    self.controller_lavori.aggiungi_lavoro(*self.lavoro3)
15
16    # Controllo preventivo prima di un eventuale tentativo di inserimento di
17    # self.cliente
18    # (che e' stato gia' inserito precedentemente)
19    self.controllo = self.controller_clienti.controllo(*self.cliente)
20
21    # Testa il corretto inserimento delle istanze
22    self.assertTrue(self.cliente in self.controller_clienti.get_all_clienti()
23                    )
24    self.assertTrue(self.lavoro in self.controller_lavori.get_lavori())
25    self.assertTrue(self.cliente2 in self.controller_clienti.get_all_clienti
26                    ())
27    self.assertTrue(self.lavoro2 in self.controller_lavori.get_lavori())
28    self.assertTrue(self.lavoro3 in self.controller_lavori.get_lavori())
29
30    # Testa che il controllo a riga 62 abbia restituito il valore corretto
31    self.assertEqual(self.controllo, "cliente_gia_presente")
32
33 def test_filtro(self):
34     # Ricava la lista filtrata
35     lista_lavori_cliente = self.controller_lavori.get_lavori_by_cliente(self.
36                               cliente[2])
37
38     # Testa che la lista filtrata non sia nulla
39     self.assertIsNotNone(lista_lavori_cliente)
40
41     # Testa che la lista sia stata filtrata correttamente
42     self.assertTrue(self.lavoro in lista_lavori_cliente)
43     self.assertFalse(self.lavoro2 in lista_lavori_cliente)
44     self.assertTrue(self.lavoro3 in lista_lavori_cliente)
45
46 def test_elimina(self):
47     # Controlla se cliente e cliente3 sono impegnati
48     cliente_impegnato = self.controller_clienti.cliente_impegnato(self.
49                               cliente[2])
50     cliente3_impegnato = self.controller_clienti.cliente_impegnato(self.
51                               cliente3[2])
52
53     # Testa il corretto funzionamento della funzione che controlla
54     # se il locale dati in ingresso e' impegnato o meno
55     self.assertTrue(cliente_impegnato)
56     self.assertFalse(cliente3_impegnato)
57
58     # Effettua l'eliminazione del cliente non impegnato e testa
59     # se l'esecuzione sia stata eseguita in maniera corretta
60     self.controller_clienti.delete_cliente_by_cs(self.cliente3[2])
61     self.assertTrue(self.cliente3 not in self.controller_clienti.
```

```
get_all_clienti()
```

Listing 4.18: Test sui lavori

4.2.2 Test sui dipendenti

Nel Listato 4.19 vediamo come abbiamo implementato i test per verificare la corretta esecuzione di alcune funzionalità relative alla sezione "Dipendenti".

```

1  def test_aggiungi_dipendente(self):
2      # Testa che le istanze non siano nulle
3      self.assertIsNotNone(self.dipendente)
4      self.assertIsNotNone(self.contratto)
5      self.assertIsNotNone(self.dati_anagrafici)
6
7      # Effettua l'inserimento
8      self.controller_dipendenti.aggiungi_dipendente(*self.dipendente)
9      contratto = self.controller_dipendenti.get_contratto_by_matricola(self.
10         dati_anagrafici[0])[0]
11
12     # Testa se le istanze sono state inserite correttamente
13     # e se il contratto inserito e' quello corretto
14     self.assertTrue(self.dati_anagrafici in self.controller_dipendenti.
15         getDipendenti())
16     self.assertEqual(contratto, self.contratto)
17
18     def test_elimina_dipendente(self):
19         # Effettua l'eliminazione del dipendente
20         self.controller_dipendenti.delete_dipendente(self.dati_anagrafici[0])
21         contratto = self.controller_dipendenti.get_contratto_by_matricola(self.
22             dati_anagrafici[0])
23
24         # Controlla che le eliminazioni, sia del dipendente, sia del contratto,
25         # siano andate a buon fine
26         self.assertTrue(self.dati_anagrafici not in self.controller_dipendenti.
27             getDipendenti())
28         self.assertEqual(contratto, [])

```

Listing 4.19: Test sui dipendenti

4.2.3 Test sull'aggiornamento del magazzino

Nel Listato 4.20 vediamo come abbiamo implementato i test per verificare la corretta esecuzione di alcune funzionalità relative alla sezione "Materie prime".

```

1  def test_ricerca_prodotto(self):
2      prodotto = self.controller_magazzino.get_mp_by_nome(self.
3         prodotto_da_cercare[0])[0]
4      self.assertEqual(self.prodotto_da_cercare, prodotto)
5
6  def test_lista_spesa(self):
7      # Testo l'inserimento del primo prodotto in lista spesa
8      self.controller_magazzino.aggiungi_in_lista_spesa(*self.prodotto_in_ls)
9      prezzo_totale = self.prodotto_in_ls[2] * self.quantita_da_aggiungere
10     self.assertTrue(
11         self.prodotto_in_ls + (prezzo_totale,) in self.controller_magazzino.
12             get_lista_spesa())
13
14     # Testo l'eliminazione del primo prodotto dalla lista della spesa

```

```

13     self.controller_magazzino.delete_by_lista_spesa(self.prodotto_in_ls[0])
14     self.assertTrue(
15         self.prodotto_in_ls + (prezzo_totale,) not in self.
            controller_magazzino.get_lista_spesa())
16
17     # Testo l'inserimento del secondo prodotto in lista spesa
18     self.controller_magazzino.aggiungi_in_lista_spesa(*self.prodotto2_in_ls)
19     prezzo_totale2 = self.prodotto2_in_ls[2] * self.quantita_da_aggiungere
20     self.assertTrue(
21         self.prodotto2_in_ls + (prezzo_totale2,) in self.controller_magazzino
            .get_lista_spesa())
22
23     # Testo l'eliminazione di tutta la lista della spesa
24     self.controller_magazzino.delete_all_lista_spesa()
25     self.assertEqual(self.controller_magazzino.get_lista_spesa(), [])
26
27     def test_modifica_quantita_in_magazzino(self):
28         # Testa l'acquisto di nuove scorte
29         self.controller_magazzino.modifica_quantita_in_magazzino(
30             self.materia_prima, self.quantita_da_aggiungere, "+")
31         self.assertEqual(
32             self.quantita + self.quantita_da_aggiungere,
33             self.controller_magazzino.get_mp_in_magazzino(0)[1]
34         )
35         self.quantita = self.controller_magazzino.get_mp_in_magazzino(0)[1]
36
37         # Testa il consumo di materie prime
38         self.controller_magazzino.modifica_quantita_in_magazzino(
39             self.materia_prima, self.quantita_da_togliere, "-")
40         self.assertEqual(
41             self.quantita - self.quantita_da_togliere,
42             self.controller_magazzino.get_mp_in_magazzino(0)[1]
43         )

```

Listing 4.20: Test sulle materie prime

4.2.4 Test sulle visite mediche

Nel Listato 4.21 vediamo come abbiamo implementato i test per verificare la corretta esecuzione di alcune funzionalità relative alla sezione "Visite Mediche".

```

1     def test_aggiungi_visita(self):
2         # Testa che le istanze non siano nulle
3         self.assertIsNotNone(self.visita)
4         self.assertIsNotNone(self.visita2)
5         self.assertIsNotNone(self.visita3)
6
7         # Effettua l'inserimento
8         self.controller_visite.aggiungi_visita(*self.visita)
9         self.controller_visite.aggiungi_visita(*self.visita2)
10        self.controller_visite.aggiungi_visita(*self.visita3)
11
12        # Testa se le istanze sono state inserite correttamente
13        self.assertTrue(self.visita + self.dipendente[:8] in self.
            controller_visite.get_all_visite())
14        self.assertTrue(self.visita2 + self.dipendente2[:8] in self.
            controller_visite.get_all_visite())
15        self.assertTrue(self.visita3 + self.dipendente[:8] in self.
            controller_visite.get_all_visite())
16
17    def test_filtro(self):

```

```

18     # Ricava la lista filtrata
19     lista_visite_dipendente = (
20         self.controller_visite.get_visite_by_dipendente(self.dipendente[0]))
21
22     # Testa che la lista filtrata non sia nulla
23     self.assertIsNotNone(lista_visite_dipendente)
24
25     # Testa che la lista sia stata filtrata correttamente
26     self.assertTrue(self.visita + self.dipendente[:8] in
27                     lista_visite_dipendente)
28     self.assertFalse(self.visita2 + self.dipendente2[:8] in
29                      lista_visite_dipendente)
30     self.assertTrue(self.visita3 + self.dipendente[:8] in
31                     lista_visite_dipendente)
32
33 def test_ultimo_elimina(self):
34     # Testa l'eliminazione della visita medica a partire dalla lista di
35     # visite mediche
36     self.controller_visite.delete_visita_by_id(self.visita3[0])
37     self.assertTrue(
38         self.visita3 + self.dipendente[:8] not in self.controller_visite.
39         get_all_visite())
40
41     # Testa l'eliminazione delle visite mediche a partire da un dipendente
42     # eliminato
43     self.controller_dipendenti.delete_dipendente(self.dipendente2[0])
44     self.assertTrue(
45         self.visita2 + self.dipendente2[:8] not in self.controller_visite.
46         get_all_visite())

```

Listing 4.21: Test sulle visite mediche

4.2.5 Test sui locali

Nel Listato 4.22 vediamo come abbiamo implementato i test per verificare la corretta esecuzione di alcune funzionalità relative alla sezione "Locali".

```

1     def test_aggiungi_locale(self):
2         # Testa che le istanze non siano nulle
3         self.assertIsNotNone(self.garage)
4         self.assertIsNotNone(self.garage2)
5         self.assertIsNotNone(self.magazzino)
6         self.assertIsNotNone(self.veicolo)
7
8         # Effettua l'inserimento
9         self.controller_locali.aggiungi_locale(*self.garage)
10        self.controller_locali.aggiungi_locale(*self.garage2)
11        self.controller_locali.aggiungi_locale(*self.magazzino)
12        self.controller_veicoli.aggiungi_veicolo(*self.veicolo)
13
14        # Testa se le istanze sono state inserite correttamente
15        self.assertTrue(self.garage in self.controller_locali.get_all_locali())
16        self.assertTrue(self.garage2 in self.controller_locali.get_all_locali())
17        self.assertTrue(self.magazzino in self.controller_locali.get_all_locali()
18                        )
19        self.assertTrue(self.veicolo in self.controller_veicoli.get_all_veicoli()
20                        )
21
22 def test_filtro(self):
23     # Ricava la lista filtrata
24     lista_garage = self.controller_locali.get_locali_by_tipo("Garage")

```

```
23
24     # Testa che la lista filtrata non sia nulla
25     self.assertIsNotNone(lista_garage)
26
27     # Testa che la lista sia stata filtrata correttamente
28     self.assertTrue(self.garage in lista_garage)
29     self.assertTrue(self.garage2 in lista_garage)
30     self.assertFalse(self.magazzino in lista_garage)
31
32 def test_filtro_veicoli_per_locale(self):
33     # Ricava la lista filtrata sia per garage che per garage2
34     lista_veicoli_garage = self.controller_veicoli.get_veicoli_by_locale(self
35         .garage[0])
36     lista_veicoli_garage2 = self.controller_veicoli.get_veicoli_by_locale(
37         self.garage2[0])
38
39     # Testa che non siano nulle
40     self.assertIsNotNone(lista_veicoli_garage)
41     self.assertIsNotNone(lista_veicoli_garage2)
42
43     # Testano che il filtraggio sia avvenuto correttamente
44     self.assertTrue(self.veicolo in lista_veicoli_garage)
45     self.assertEqual(lista_veicoli_garage2, [])
46
47 def test_ultimo_elimina_locale(self):
48     # Controlla se garage e magazzino sono impegnati
49     garage_impegnato = self.controller_locali.locale_impegnato(self.garage
50         [0])
51     magazzino_impegnato = self.controller_locali.locale_impegnato(self.
52         magazzino[0])
53
54     # Testa il corretto funzionamento della funzione che controlla
55     # se il locale dato in ingresso e' impegnato o meno
56     self.assertTrue(garage_impegnato)
57     self.assertFalse(magazzino_impegnato)
58
59     # Effettua l'eliminazione del locale non impegnato e testa
60     # se l'esecuzione sia stata eseguita in maniera corretta
61     self.controller_locali.delete_locale_by_indirizzo(self.magazzino[0])
62     self.assertTrue(self.magazzino not in self.controller_locali.
63         get_all_locali())
```

Listing 4.22: Test sui locali

In quest'ultimo capitolo presenteremo il manuale utente del software, un documento che, al pari del prodotto finale, va consegnato al cliente nella fase di rilascio, in modo che questi possa comprendere al meglio come orientarsi all'interno dell'applicazione. Verranno mostrate tutte le diverse schermate del software, divise per settori applicativi, ognuna delle quali sarà accompagnata da una descrizione delle modalità di esplorazione da parte degli utenti.

5.1 Autenticazione

Nella Figura 5.1 vediamo la schermata di benvenuto di EdilGest; qui è possibile, dopo aver inserito le credenziali corrette, effettuare il login, attraverso il click sul pulsante "Accedi", oppure andare nella schermata "Cambio credenziali", mostrata in Figura 5.2, attraverso il click sul pulsante corrispondente.

Le credenziali, al primo utilizzo dell'applicazione, sono:

- Nome utente: `utente`
- Password: `pass_temp`

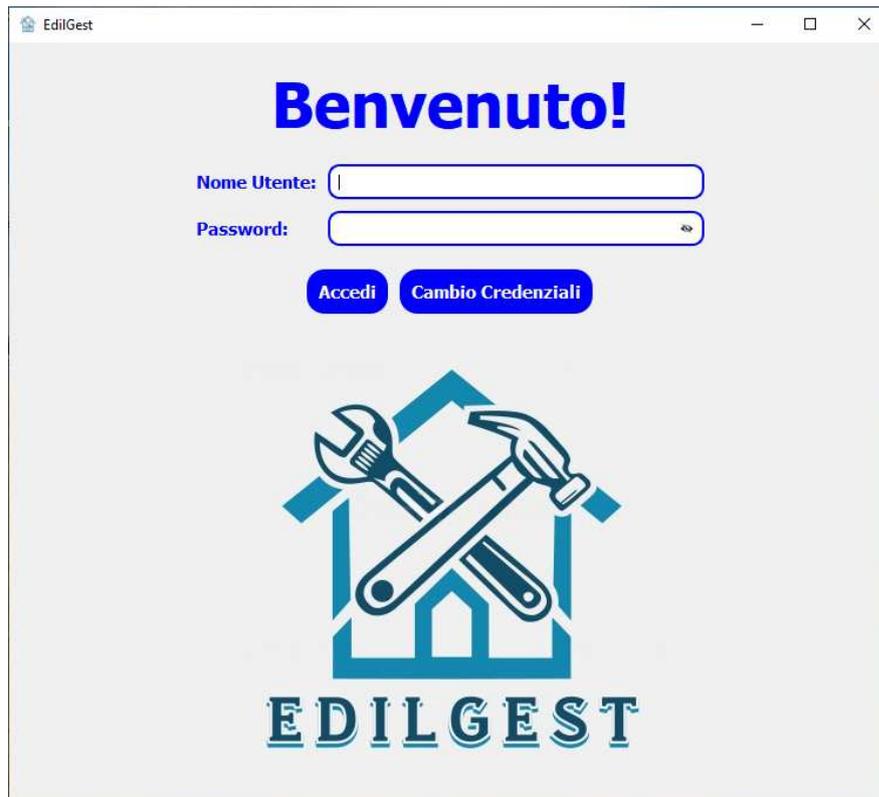
Come accennato prima, per cambiare le credenziali bisogna andare nella schermata "Cambio credenziali" che vediamo nella Figura 5.2; qui l'utente può tornare alla schermata di accesso, mostrata in Figura 5.1, attraverso il click sul pulsante "Torna alla schermata di accesso", senza cambiare, quindi, le credenziali, oppure confermarne le modifiche, dopo aver correttamente completato il form proposto.

Viene data all'utente la possibilità, cliccando sull'icona con l'occhio sulla destra della casella di testo, di rendere visibile la password.

5.2 Home Page

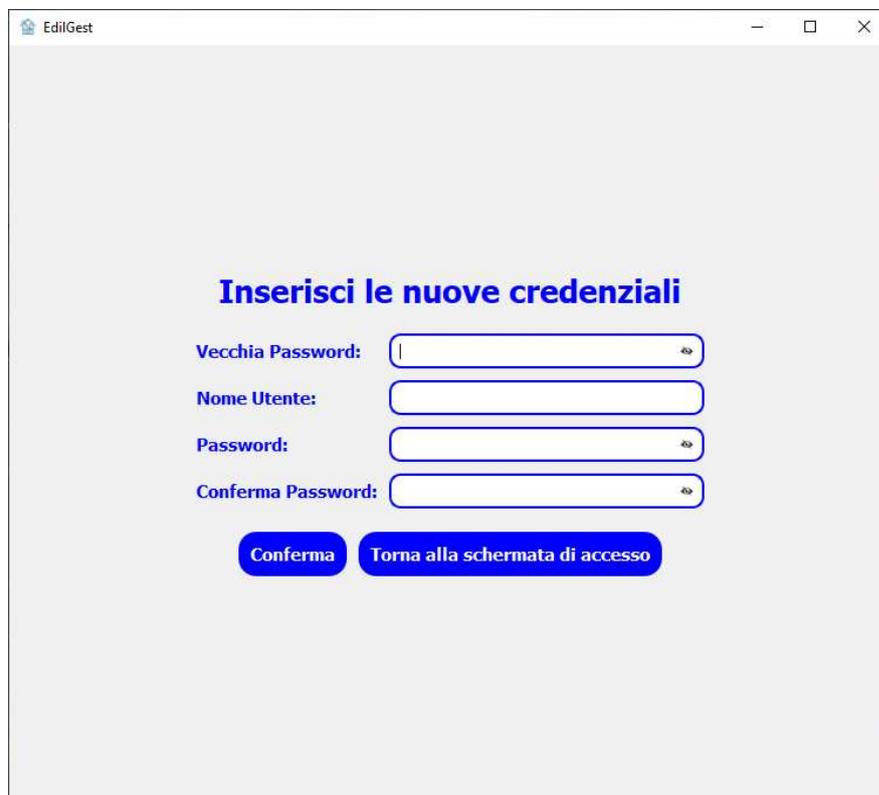
Una volta effettuato correttamente l'accesso, l'utente vede la Home Page di EdilGest, mostrata in Figura 5.3; qui egli può entrare nelle varie sezioni del software in tre modi diversi:

- *click sul pulsante*: vale per le sezioni "Dipendenti", "Magazzino", "Locali";



The screenshot shows a web browser window titled "EdilGest". The main heading is "Benvenuto!" in large blue font. Below it are two input fields: "Nome Utente:" and "Password:". Under the password field is a small eye icon. There are two blue buttons: "Accedi" and "Cambio Credenziali". In the center is a logo featuring a house outline with a wrench and a hammer crossed over it, and the text "EDILGEST" below it.

Figura 5.1: Schermata di Login



The screenshot shows a web browser window titled "EdilGest". The main heading is "Inserisci le nuove credenziali" in blue font. Below it are four input fields: "Vecchia Password:", "Nome Utente:", "Password:", and "Conferma Password:". Each password field has a small eye icon. There are two blue buttons: "Conferma" and "Torna alla schermata di accesso".

Figura 5.2: Schermata per il cambio delle credenziali

- *click sull'opzione del menù a tendina associato al pulsante*: vale per le sezioni "Lavori", "Preventivi" (pulsante "Lavori e preventivi"), "Corsi di sicurezza", "Visite mediche" (pulsante "Sicurezza"), "Entrate", "Uscite" e "Bilanci" (pulsante "Contabilità");
- *click sull'opzione del menù in alto oppure combinazione di tasti*: vale per le sezioni "Clienti", "Veicoli", "Fornitori", "Statistiche" e "Tutorial"; qui, cliccando sull'opzione "File" e selezionando "Esci", l'utente può chiudere l'applicazione.

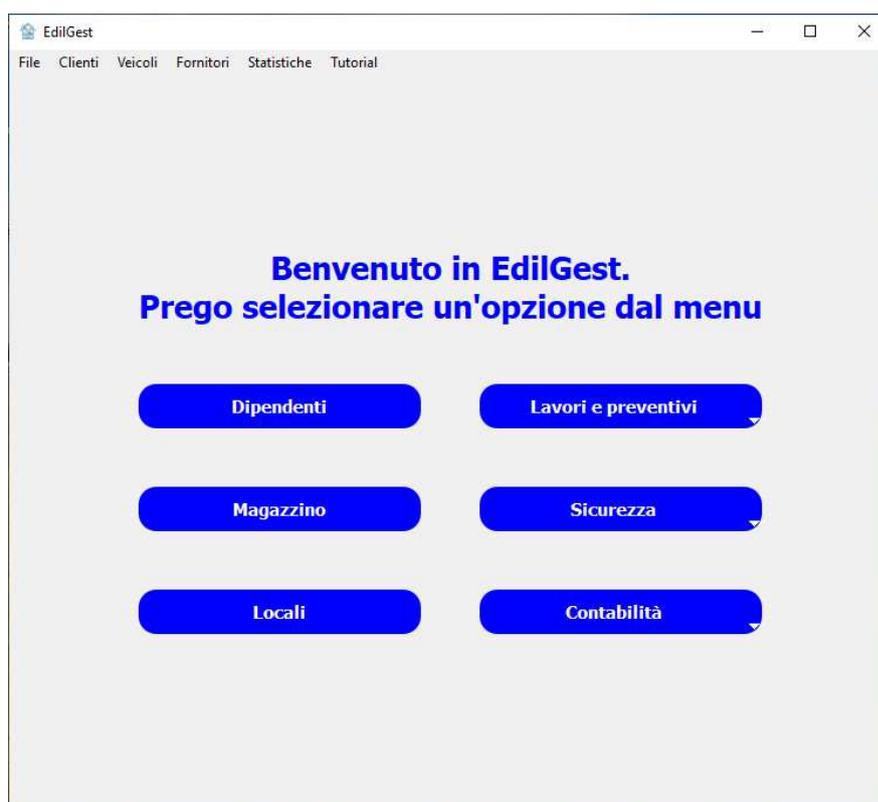


Figura 5.3: Schermata relativa alla Home Page del sistema

5.3 Sezione "Lavori"

Cliccando sull'opzione "Lavori" del menù a tendina associato al pulsante "Lavori e preventivi" della Home Page in Figura 5.3, l'utente visualizza la schermata "Lista lavori", mostrata in Figura 5.4; qui vede una lista contenente le denominazioni di tutti i lavori presenti nel sistema.

Cliccando sul pulsante "Elimina tutto", l'utente può eliminare, previa conferma, per evitare click accidentali, tutti i lavori presenti nel sistema. I rispettivi clienti non verranno eliminati.

Cliccando sull'opzione "Cerca per cliente" del menù a tendina associato al pulsante "Filtri", apparirà una schermata pop-up, come mostrato in Figura 5.5, dove l'utente può selezionare il codice fiscale del cliente di cui vuole visualizzare i lavori; al click dell'opzione "Ok", il sistema visualizza la lista di tutti i lavori associati al cliente selezionato.

Cliccando sul pulsante "Aggiungi" della schermata "Lista lavori", mostrata in Figura 5.4, l'utente può visualizzare la pagina di inserimento di un nuovo lavoro, in cui sono presenti due sezioni:

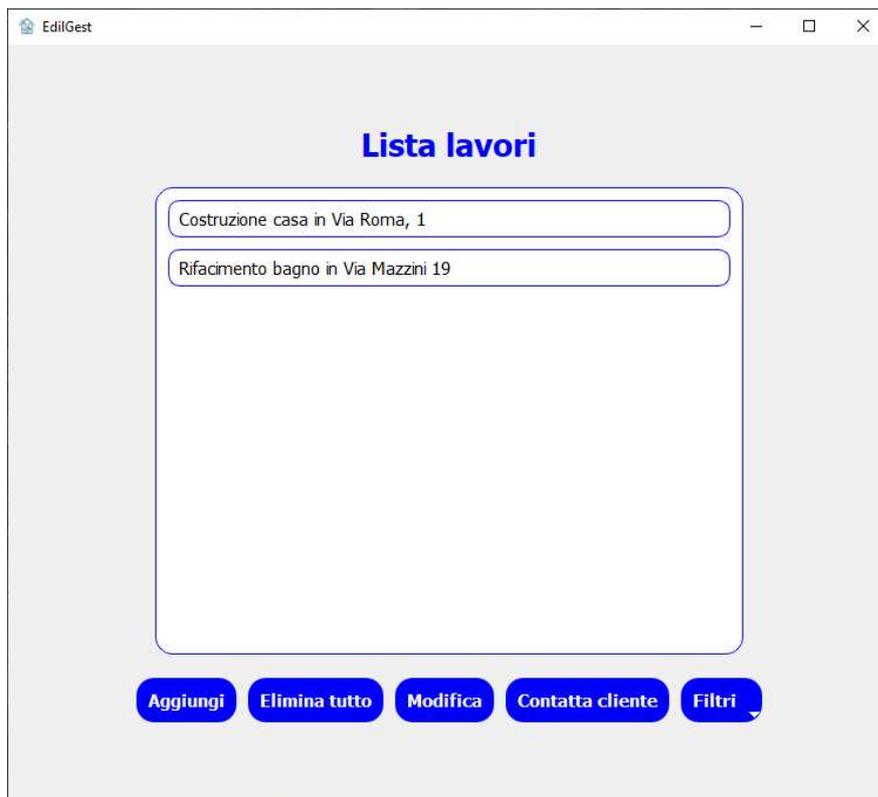


Figura 5.4: Schermata relativa alla lista dei lavori

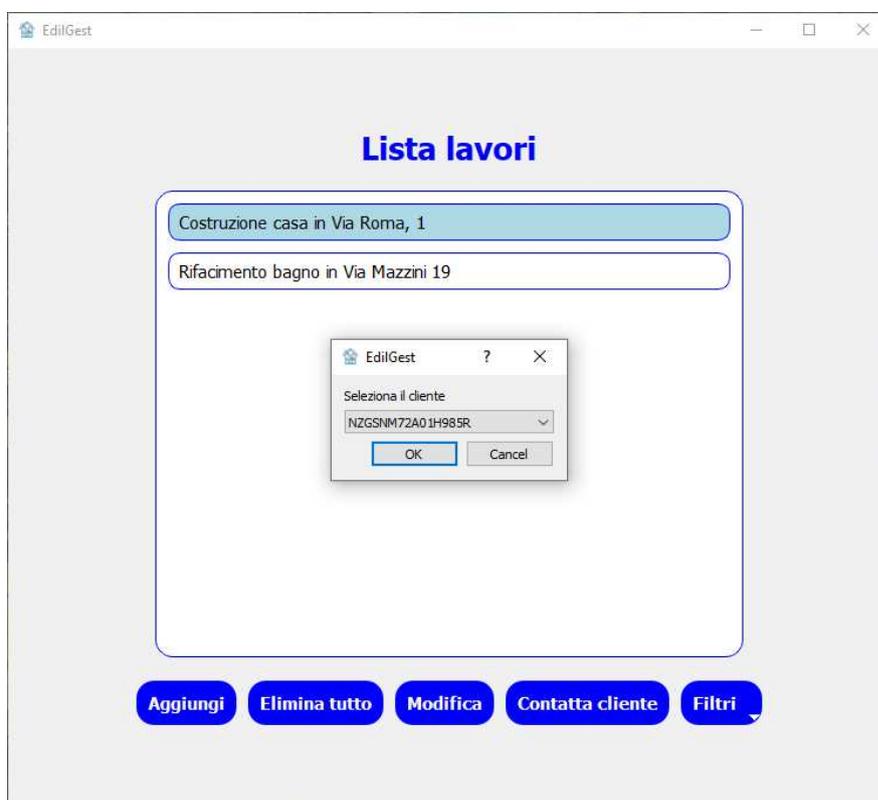


Figura 5.5: Schermata di pop-up che permette il filtraggio dei lavori per cliente

- *sezione lavoro*, visualizzata in Figura 5.6; qui l'utente può compilare i dati relativi al lavoro;
- *sezione cliente*, visualizzata in Figura 5.7; qui l'utente può compilare i dati relativi al cliente del lavoro.

Si può passare da una sezione all'altra attraverso un click sullo switcher presente sotto la scritta "Completa i seguenti campi".

Una volta completati correttamente tutti i campi, l'utente può completare l'inserimento del lavoro e del cliente (se questo non è già presente nel sistema) cliccando sul pulsante "Conferma". Al termine dell'operazione, il sistema visualizza nuovamente la schermata "Lista lavori" aggiornata.

The screenshot shows a web form titled "Completa i seguenti campi" within a window labeled "EdilGest". At the top, there is a toggle switch between "Lavoro" (selected) and "Cliente". Below this, the form is organized into two columns of input fields. The left column includes: "Denominazione" (text input), "Importo [€]" (text input), "Inizio previsto" (date dropdown), "Inizio effettivo" (date dropdown), "Termine previsto" (date dropdown), "Termine effettivo" (date dropdown), and "Pagato" (checkbox). The right column includes: "Foglio" (text input), "Particella" (text input), "Sub" (text input), "Classe" (text input), "Categoria" (text input), and "Consistenza" (text input). Below the "Pagato" checkbox is a "Dettagli pagamento" text input field. At the bottom center, there is a blue "Conferma" button.

Figura 5.6: Schermata per l'inserimento di un nuovo lavoro (sezione "Lavoro")

Dopo aver selezionato un lavoro nella schermata "Lista lavori", mostrata in Figura 5.4, l'utente può effettuare due azioni:

- *visualizzare il dettaglio del lavoro selezionato*, attraverso il click sul pulsante "Modifica"; visualizza la schermata "Vista lavoro", presentata con due sezioni diverse, come vediamo nelle Figure 5.8 e 5.9;
- *contattare il cliente del lavoro selezionato*, attraverso il click sul pulsante "Contatta cliente"; apre una nuova e-mail da inviare con il campo "Destinatario" precompilato con la mail del cliente del lavoro selezionato.

Nella schermata "Vista lavoro", mostrata in Figura 5.8, l'utente può confermare le modifiche effettuate, attraverso il click sul pulsante "Conferma modifica", eliminare il lavoro selezionato, cliccando sul pulsante "Elimina", oppure effettuare il pagamento in entrata dell'importo, attraverso il click sul pulsante "Inserisci pagamento in entrata".

The screenshot shows a web browser window titled 'EdilGest'. The main heading is 'Completa i seguenti campi'. Below the heading is a toggle switch for 'Lavoro' (selected) and 'Cliente'. The form contains the following fields:

- Nome: text input
- Cognome: text input
- Codice fiscale: text input
- Residenza: text input
- Data nascita: dropdown menu with '09/10/2024' selected
- Email: text input
- Telefono: text input

A blue 'Conferma' button is located at the bottom center of the form.

Figura 5.7: Schermata per l'inserimento di un nuovo lavoro (sezione "Cliente")

The screenshot shows a web browser window titled 'EdilGest'. The main heading is 'Info lavoro'. Below the heading is a toggle switch for 'Lavoro' (selected) and 'Cliente'. The form contains the following fields:

- Denominazione: text input with 'Costruzione casa in Via Roma, 1'
- Importo [€]: text input with '100000.00'
- Inizio previsto: dropdown menu with '07/01/2025' selected
- Inizio effettivo: dropdown menu with '07/10/2025' selected
- Termine previsto: dropdown menu with '19/12/2025' selected
- Termine effettivo: dropdown menu with '19/12/2025' selected
- Foglio: text input with '1'
- Particella: text input with '2'
- Sub: text input with '3'
- Classe: text input with '4'
- Categoria: text input with '5'
- Consistenza: text input with '6'

At the bottom, there is a 'Pagato' checkbox (unchecked), a 'Dettagli pagamento' section with a text input containing 'acconto €10000', and three blue buttons: 'Conferma modifica', 'Elimina', and 'Inserisci pagamento in entrata'.

Figura 5.8: Schermata relativa ai dettagli del lavoro selezionato (sezione "Lavoro")



EdilGest

Info lavoro

Lavoro Cliente

Nome

Cognome

Codice fiscale

Residenza

Data nascita

Email

Telefono

Figura 5.9: Schermata relativa ai dettagli del lavoro selezionato (sezione "Cliente")

5.4 Sezione "Preventivi"

Cliccando sull'opzione "Preventivi" del menù a tendina associato al pulsante "Lavori e preventivi" della Home Page, in Figura 5.3, l'utente visualizza la schermata "Lista preventivi", mostrata in Figura 5.10; qui vede una lista contenente gli ID di tutti i preventivi presenti nel sistema.

Cliccando sul pulsante "Elimina tutto", l'utente può eliminare, previa conferma, per evitare click accidentali, tutti i preventivi presenti nel sistema. I rispettivi clienti non verranno eliminati.

Cliccando sull'opzione "Cerca per cliente" del menù a tendina associato al pulsante "Filtri", apparirà una schermata pop-up, come mostrato in Figura 5.11, dove l'utente può selezionare il codice fiscale del cliente di cui vuole visualizzare i preventivi; al click dell'opzione "Ok", il sistema visualizza la lista di tutti i preventivi associati al cliente selezionato.

Cliccando sul pulsante "Aggiungi" della schermata "Lista preventivi", mostrata in Figura 5.10, l'utente può visualizzare la pagina di inserimento di un nuovo preventivo, in cui sono presenti due sezioni:

- *sezione preventivo*, visualizzata in Figura 5.12; qui l'utente può compilare i dati relativi al preventivo;
- *sezione cliente*, visualizzata in Figura 5.13; qui l'utente può compilare i dati relativi al cliente del preventivo.

Si può passare da una sezione all'altra attraverso un click sullo switcher presente sotto la scritta "Completa i seguenti campi".

Una volta completati correttamente tutti i campi, l'utente può completare l'inserimento del preventivo e del cliente (se questo non è già presente nel sistema) cliccando sul pulsante

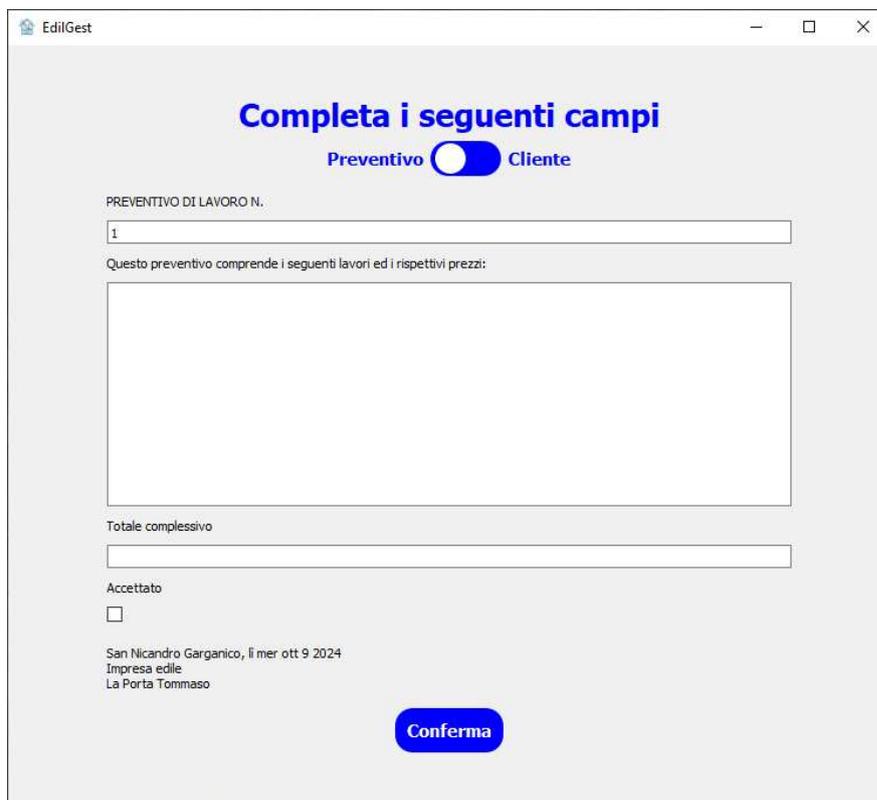


Figura 5.10: Schermata relativa alla lista dei preventivi



Figura 5.11: Schermata di pop-up che permette il filtraggio dei preventivi per cliente

"Conferma". Al termine dell'operazione, il sistema visualizza nuovamente la schermata "Lista preventivi", mostrata in Figura 5.10, aggiornata.



The screenshot shows a web application window titled "EdilGest". The main heading is "Completa i seguenti campi" in blue. Below it is a toggle switch for "Preventivo" (which is turned on) and "Cliente". The form contains the following elements:

- A label "PREVENTIVO DI LAVORO N." followed by a text input field containing the number "1".
- A label "Questo preventivo comprende i seguenti lavori ed i rispettivi prezzi:" followed by a large empty rectangular area.
- A label "Totale complessivo" followed by a text input field.
- A label "Accettato" followed by an unchecked checkbox.
- Footer text: "San Nicandro Garganico, Il mer ott 9 2024", "Impresa edile", "La Porta Tommaso".
- A blue "Conferma" button at the bottom center.

Figura 5.12: Schermata per l'inserimento di un nuovo preventivo (sezione "Preventivo")

Dopo aver selezionato un preventivo nella schermata "Lista preventivi", mostrata in Figura 5.10, l'utente può effettuare due azioni, ovvero:

- *visualizzare il dettaglio del preventivo selezionato*, attraverso il click sul pulsante "Modifica"; visualizza la schermata "Vista preventivo", presentata con due sezioni diverse, come vediamo nelle Figure 5.14 e 5.15;
- *contattare il cliente del preventivo selezionato*, attraverso il click sul pulsante "Contatta cliente"; apre una nuova e-mail da inviare con il campo "Destinatario" precompilato con la mail del cliente del preventivo selezionato.

Nella schermata "Vista preventivo", mostrata in Figura 5.14, l'utente può confermare le modifiche effettuate, attraverso il click sul pulsante "Conferma modifica", eliminare il preventivo selezionato, cliccando sul pulsante "Elimina", oppure scaricarlo in PDF, cliccando sull'opzione "File" del menù in alto e selezionando "Salva come PDF".

5.5 Sezione "Dipendenti"

Cliccando sul pulsante "Dipendenti" della Home Page, in Figura 5.3, l'utente visualizza la schermata "Lista dipendenti", mostrata in Figura 5.16; qui vede una lista contenente il nome, il cognome e la matricola di tutti i dipendenti presenti nel sistema.

Cliccando sul pulsante "Elimina tutto", l'utente può eliminare, previa conferma, per evitare click accidentali, tutti i dipendenti presenti nel sistema. Verranno eliminate anche le rispettive visite mediche.

The screenshot shows a web form titled "Completa i seguenti campi" (Complete the following fields) in the EdilGest application. At the top, there is a toggle switch for "Preventivo" (selected) and "Cliente". Below the toggle, there are seven input fields: "Nome", "Cognome", "Codice fiscale", "Residenza", "Data nascita" (with a dropdown menu showing "09/10/2024"), "Email", and "Telefono". At the bottom of the form is a blue button labeled "Conferma".

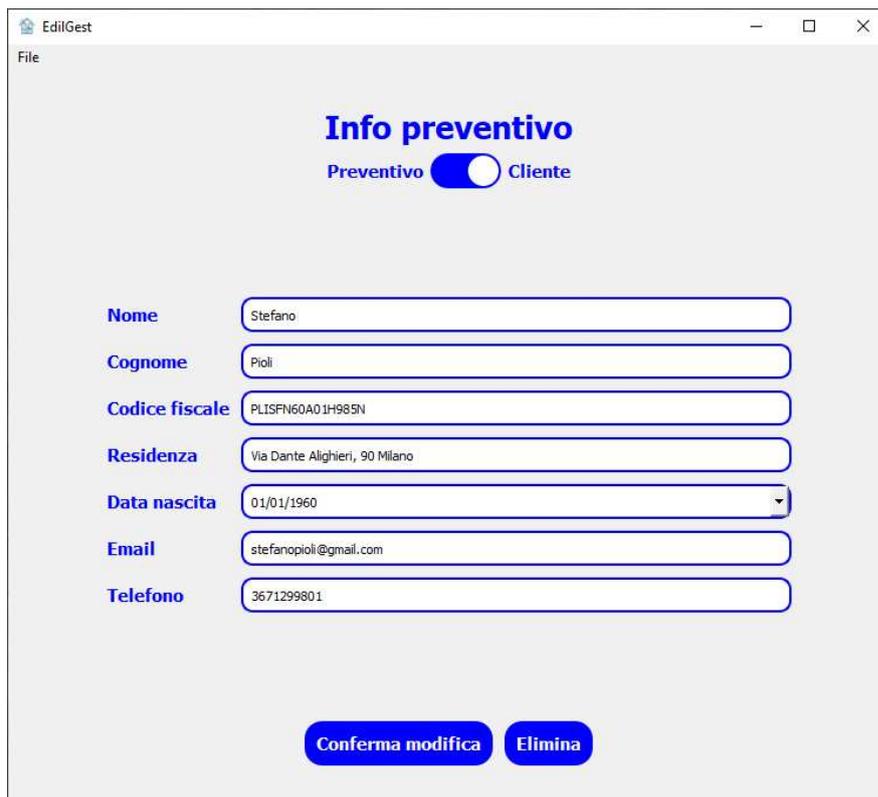
Figura 5.13: Schermata per l'inserimento di un nuovo preventivo (sezione "Cliente")

The screenshot shows a web form titled "Info preventivo" (Info preventive) in the EdilGest application. At the top, there is a toggle switch for "Preventivo" (selected) and "Cliente". Below the toggle, there is a text input field for "PREVENTIVO DI LAVORO N." with the value "2". Below this, there is a table showing the items included in the estimate:

Questo preventivo comprende i seguenti lavori ed i rispettivi prezzi:	
Sostituzione impianto bagno	€1000
Sostituzione sanitari	€1000
Pavimentazione	€2000

Below the table, there is a text input field for "Totale complessivo [€]" with the value "4000.00". Below this, there is a checkbox labeled "Accettato" which is checked. At the bottom, there is a text input field for the address: "San Nicandro Garganico, il mer ott 9 2024", "Impresa edile", "La Porta Tommaso". At the bottom of the form are two blue buttons: "Conferma modifica" and "Elimina".

Figura 5.14: Schermata relativa ai dettagli del preventivo selezionato (sezione "Preventivo")

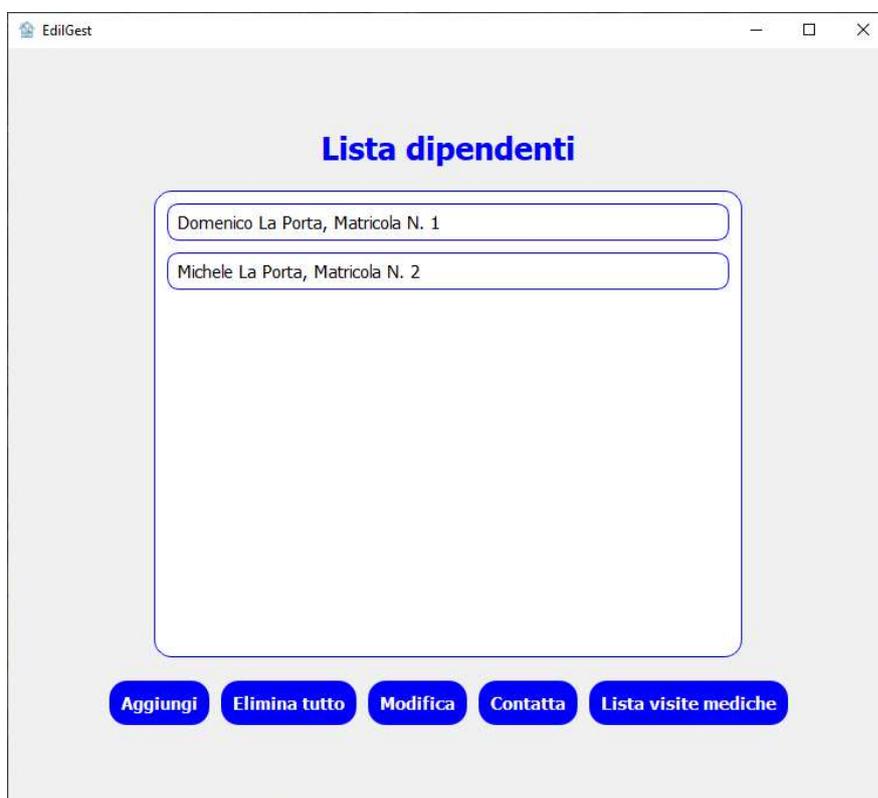


The screenshot shows a web application window titled 'EdilGest' with a 'File' menu. The main heading is 'Info preventivo'. Below it, there is a toggle switch for 'Preventivo' (which is selected) and 'Cliente'. The form contains the following fields:

Nome	Stefano
Cognome	Pioli
Codice fiscale	PLISFN60A01H985N
Residenza	Via Dante Alighieri, 90 Milano
Data nascita	01/01/1960
Email	stefanopioli@gmail.com
Telefono	3671299801

At the bottom of the form are two buttons: 'Conferma modifica' and 'Elimina'.

Figura 5.15: Schermata relativa ai dettagli del preventivo selezionato (sezione "Cliente")



The screenshot shows a web application window titled 'EdilGest' with a 'File' menu. The main heading is 'Lista dipendenti'. The list contains two entries:

Domenico La Porta, Matricola N. 1
Michele La Porta, Matricola N. 2

At the bottom of the list are five buttons: 'Aggiungi', 'Elimina tutto', 'Modifica', 'Contatta', and 'Lista visite mediche'.

Figura 5.16: Schermata relativa alla lista dei dipendenti

Cliccando sul pulsante "Aggiungi" della schermata "Lista dipendenti", mostrata in Figura 5.16 l'utente può visualizzare la pagina di inserimento di un nuovo dipendente, in cui sono presenti due sezioni:

- *sezione dati anagrafici*, visualizzata in Figura 5.17; qui l'utente può compilare i dati anagrafici;
- *sezione contratto*, visualizzata in Figura 5.18; qui l'utente può compilare i dati relativi al contratto del dipendente.

Si può passare da una sezione all'altra attraverso un click sullo switcher presente sotto la scritta "Completa i seguenti campi".

Una volta completati correttamente tutti i campi, l'utente può completare l'inserimento del dipendente cliccando sul pulsante "Conferma". Al termine dell'operazione, il sistema visualizza nuovamente la schermata "Lista dipendenti", mostrata in Figura 5.16, aggiornata.



The screenshot shows a web browser window titled "EdilGest". The main heading is "Completa i seguenti campi". Below the heading is a toggle switch with "Anagrafica" on the left and "Contratto" on the right; the "Anagrafica" side is selected. The form consists of the following fields:

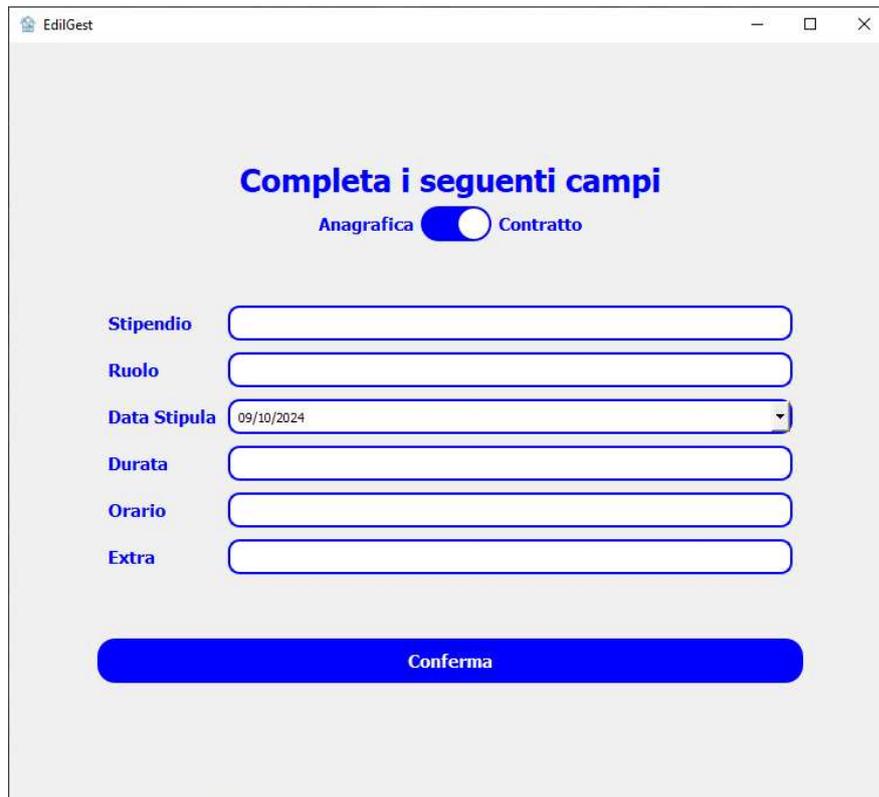
- Matricola:
- Nome:
- Cognome:
- Codice fiscale:
- Residenza:
- Data nascita:
- Email:
- Telefono:

At the bottom of the form is a large blue button labeled "Conferma".

Figura 5.17: Schermata per l'inserimento di un nuovo dipendente (sezione "Dati Anagrafici")

Dopo aver selezionato un dipendente nella schermata "Lista dipendenti", mostrata in Figura 5.16, l'utente può effettuare tre azioni:

- *visualizzare il dettaglio del dipendente selezionato*, attraverso il click sul pulsante "Modifica"; visualizza la schermata "Vista dipendente", presentata con due sezioni diverse, come vediamo nelle Figure 5.19 e 5.20
- *contattare il dipendente selezionato*, attraverso il click sul pulsante "Contatta"; apre una nuova e-mail da inviare, con il campo "Destinatario" precompilato con la mail del dipendente selezionato.



The screenshot shows a web browser window titled 'EdilGest'. The main heading is 'Completa i seguenti campi' in blue. Below it, there are two radio buttons: 'Anagrafica' (unselected) and 'Contratto' (selected). The form contains the following fields:

- Stipendio: text input field
- Ruolo: text input field
- Data Stipula: date input field with the value '09/10/2024' and a dropdown arrow
- Durata: text input field
- Orario: text input field
- Extra: text input field

At the bottom of the form is a large blue button labeled 'Conferma'.

Figura 5.18: Schermata per l'inserimento di un nuovo dipendente (sezione "Contratto")

- *visualizzare la lista delle visite mediche effettuate dal dipendente selezionato*, attraverso il click sul pulsante "Lista visite mediche"; apre la schermata "Lista visite mediche", mostrata in Figura 5.29, con presenti soltanto le visite mediche effettuate dal dipendente selezionato (inoltre, sono presenti soltanto i pulsanti "Modifica" e "Contatta dipendente").

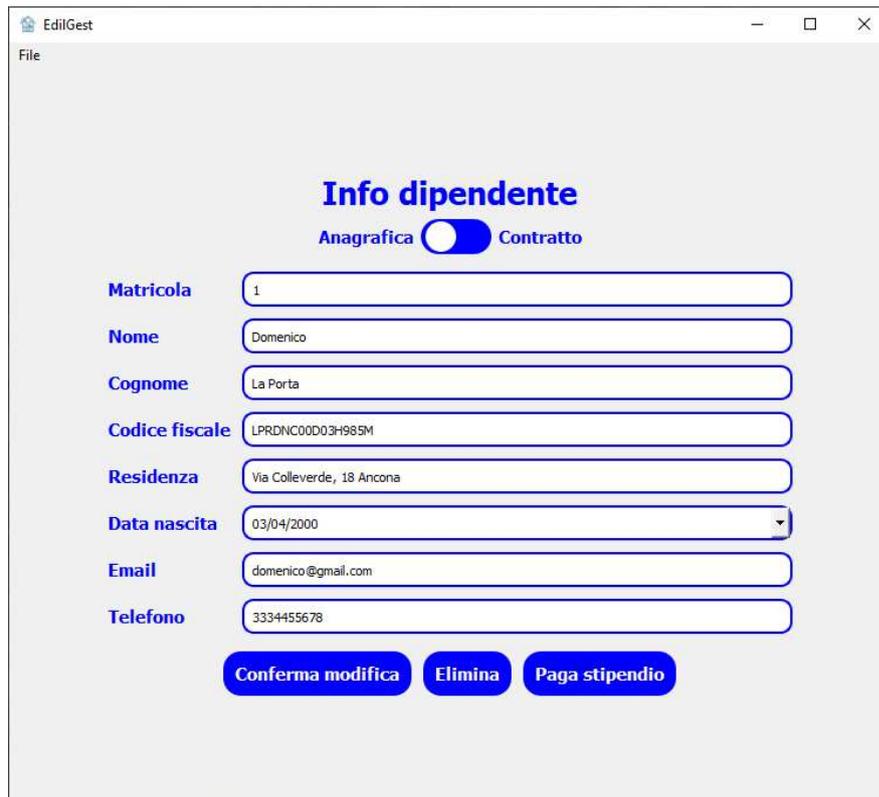
Nella schermata "Vista dipendente", mostrata in Figura 5.19, l'utente può confermare le modifiche effettuate, attraverso il click sul pulsante "Conferma modifica", eliminare il dipendente selezionato, cliccando sul pulsante "Elimina", pagare lo stipendio del dipendente selezionato, attraverso il click sul pulsante "Paga stipendio", oppure scaricare il PDF del contratto del dipendente selezionato, cliccando sull'opzione "File" del menù in alto e selezionando "Salva come PDF".

5.6 Sezione "Clienti"

Cliccando sull'opzione "Lista clienti" del menù a tendina associato al pulsante "Clienti" della Home Page, in Figura 5.3, precisamente nel menù in alto, l'utente visualizza la schermata "Lista clienti", mostrata in Figura 5.21; qui vede una lista contenente i codici fiscali, i nomi e i cognomi di tutti i clienti presenti nel sistema.

Cliccando sull'opzione "Cerca per nome" del menù a tendina associato al pulsante "Filtri", apparirà una schermata pop-up, come mostrato in Figura 5.22, dove l'utente può digitare il nome desiderato; al click dell'opzione "Ok", il sistema visualizza la lista di tutti i clienti con il nome inserito.

Cliccando sull'opzione "Cerca per cognome" del menù a tendina associato al pulsante "Filtri", apparirà una schermata pop-up, come mostrato in Figura 5.23, dove l'utente può

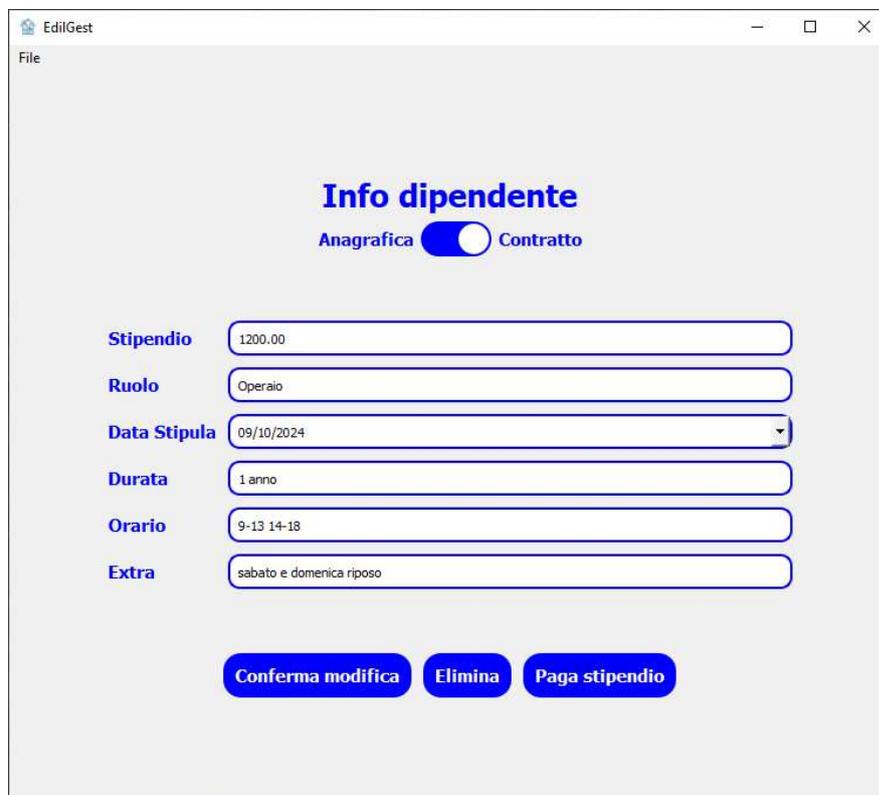


The screenshot shows a web application window titled 'EdilGest' with a 'File' menu. The main heading is 'Info dipendente'. Below it, there is a toggle switch for 'Anagrafica' (which is selected) and 'Contratto'. The form contains the following fields:

Matricola	1
Nome	Domenico
Cognome	La Porta
Codice fiscale	LPRDNC00D03H985M
Residenza	Via Colleverde, 18 Ancona
Data nascita	03/04/2000
Email	domenico@gmail.com
Telefono	3334455678

At the bottom of the form are three buttons: 'Conferma modifica', 'Elimina', and 'Paga stipendio'.

Figura 5.19: Schermata relativa ai dettagli del dipendente selezionato (sezione "Dati Anagrafici")



The screenshot shows the same web application window, but the toggle switch is now set to 'Contratto'. The form contains the following fields:

Stipendio	1200.00
Ruolo	Operaio
Data Stipula	09/10/2024
Durata	1 anno
Orario	9-13 14-18
Extra	sabato e domenica riposo

At the bottom of the form are three buttons: 'Conferma modifica', 'Elimina', and 'Paga stipendio'.

Figura 5.20: Schermata relativa ai dettagli del dipendente selezionato (sezione "Contratto")

digitare il cognome desiderato; al click dell'opzione "Ok", il sistema visualizza la lista di tutti i clienti con il cognome inserito.

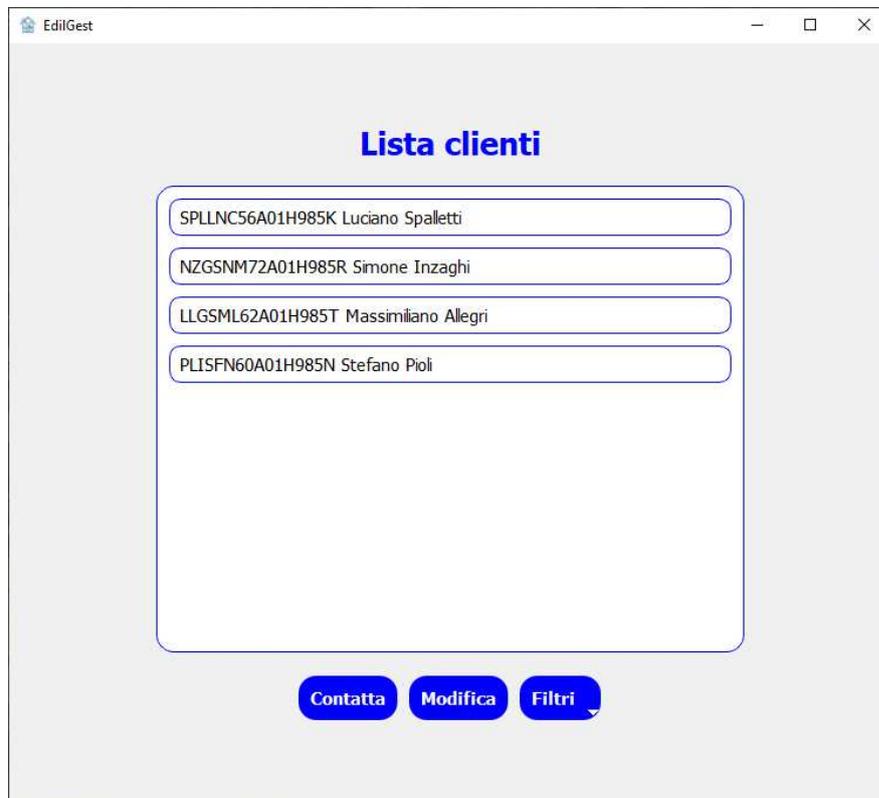


Figura 5.21: Schermata relativa alla lista dei clienti

Dopo aver selezionato un cliente nella schermata "Lista clienti", mostrata in Figura 5.21, l'utente può effettuare due azioni:

- *visualizzare il dettaglio del cliente selezionato*, attraverso il click sul pulsante "Modifica"; visualizza la schermata "Vista cliente", mostrata in Figura 5.24;
- *contattare il cliente selezionato*, attraverso il click sul pulsante "Contatta"; apre una nuova e-mail da inviare con il campo "Destinatario" precompilato con la mail del cliente selezionato.

Nella schermata "Vista cliente", mostrata in Figura 5.24, l'utente può confermare le modifiche effettuate, attraverso il click sul pulsante "Conferma modifica", oppure eliminare il preventivo selezionato, cliccando sul pulsante "Elimina".

L'eliminazione non va a buon fine se il cliente selezionato è impegnato in un lavoro o in un preventivo; mostriamo la schermata di eliminazione non riuscita nella Figura 5.25.

5.7 Sezione "Corsi di sicurezza"

Cliccando sull'opzione "Corsi di sicurezza sul lavoro" del menù a tendina associato al pulsante "Sicurezza" della Home Page, in Figura 5.3, l'utente visualizza la schermata "Lista corsi di sicurezza"; qui vede una lista contenente gli ID e le date di tutti i corsi di sicurezza presenti nel sistema.

Cliccando sul pulsante "Elimina tutto", l'utente può eliminare, previa conferma, per evitare click accidentali, tutti i corsi di sicurezza presenti nel sistema.

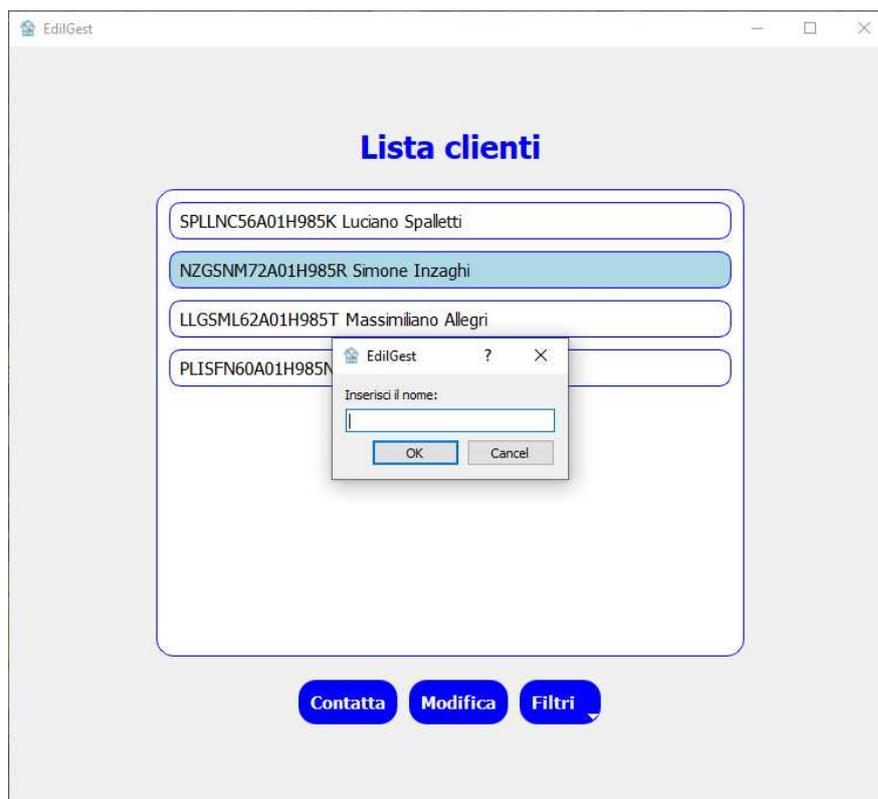


Figura 5.22: Schermata di pop-up che permette il filtraggio dei clienti per nome

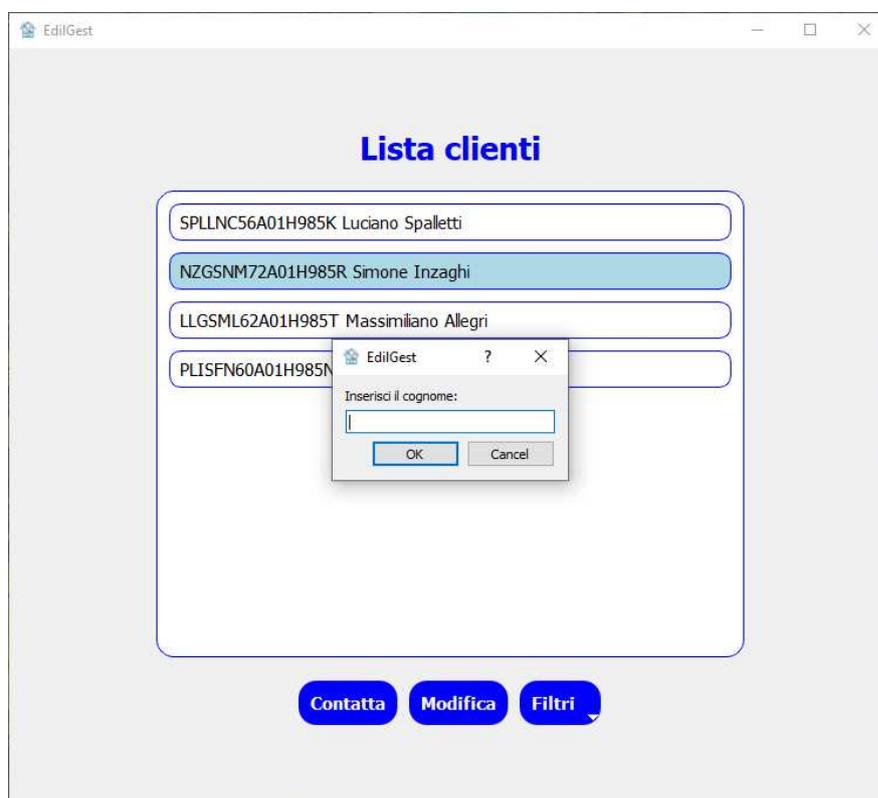
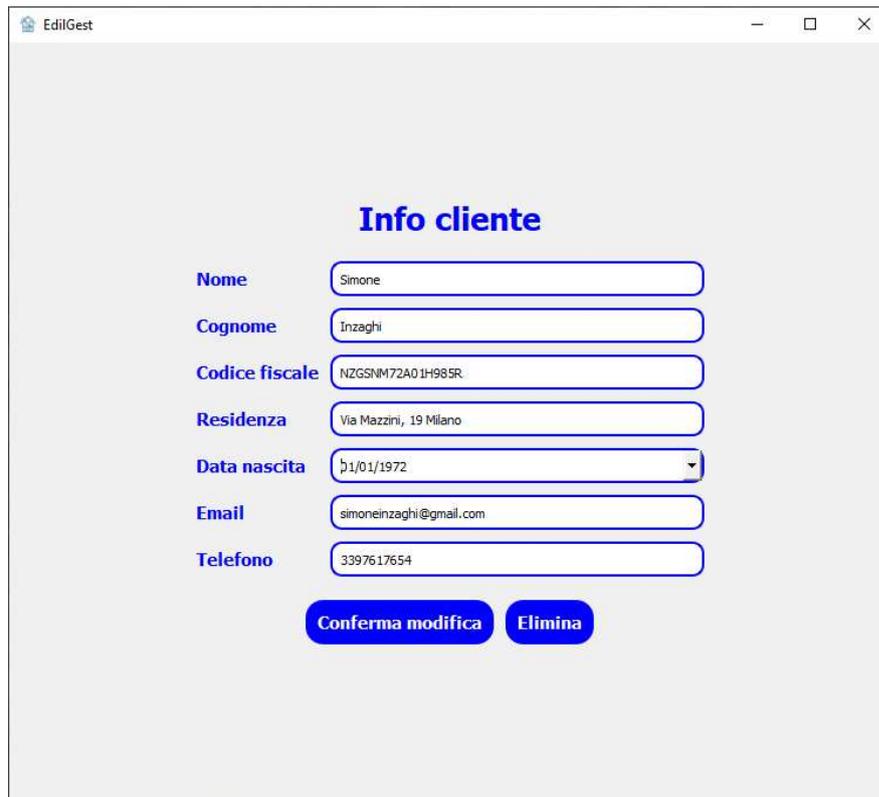


Figura 5.23: Schermata di pop-up che permette il filtraggio dei clienti per cognome



EdilGest

Info cliente

Nome

Cognome

Codice fiscale

Residenza

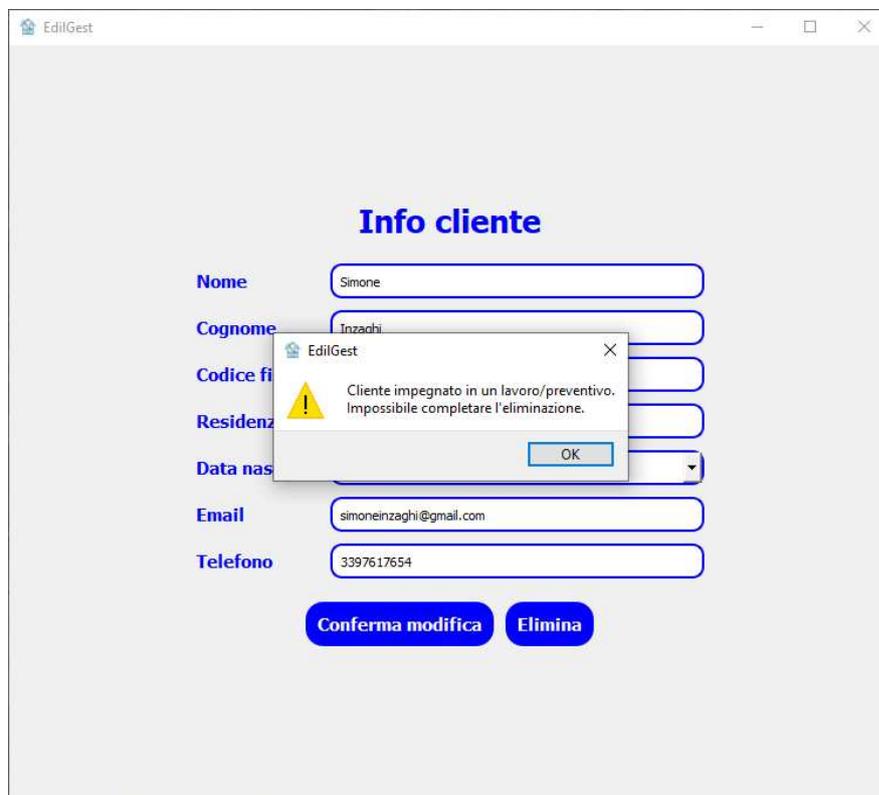
Data nascita

Email

Telefono

Conferma modifica **Elimina**

Figura 5.24: Schermata relativa ai dettagli del cliente selezionato



EdilGest

Info cliente

Nome

Cognome

Codice fiscale

Residenza

Data nascita

Email

Telefono

Conferma modifica **Elimina**

EdilGest

⚠ Cliente impegnato in un lavoro/preventivo.
Impossibile completare l'eliminazione.

OK

Figura 5.25: Avviso "Cliente impegnato"



Figura 5.26: Schermata relativa alla lista dei corsi di sicurezza sul lavoro

Cliccando sul pulsante "Aggiungi" della schermata "Lista corsi di sicurezza", mostrata in Figura 5.26 l'utente può visualizzare la schermata di inserimento di un nuovo corso di sicurezza, mostrata in Figura 5.27.

Una volta completati correttamente tutti i campi, l'utente può completare l'inserimento del corso cliccando sul pulsante "Conferma". Al termine dell'operazione, il sistema visualizza nuovamente la schermata "Lista corsi di sicurezza", mostrata in Figura 5.26, aggiornata.

Dopo aver selezionato un corso nella schermata "Lista corsi di sicurezza", mostrata in Figura 5.26, l'utente può effettuare una sola azione:

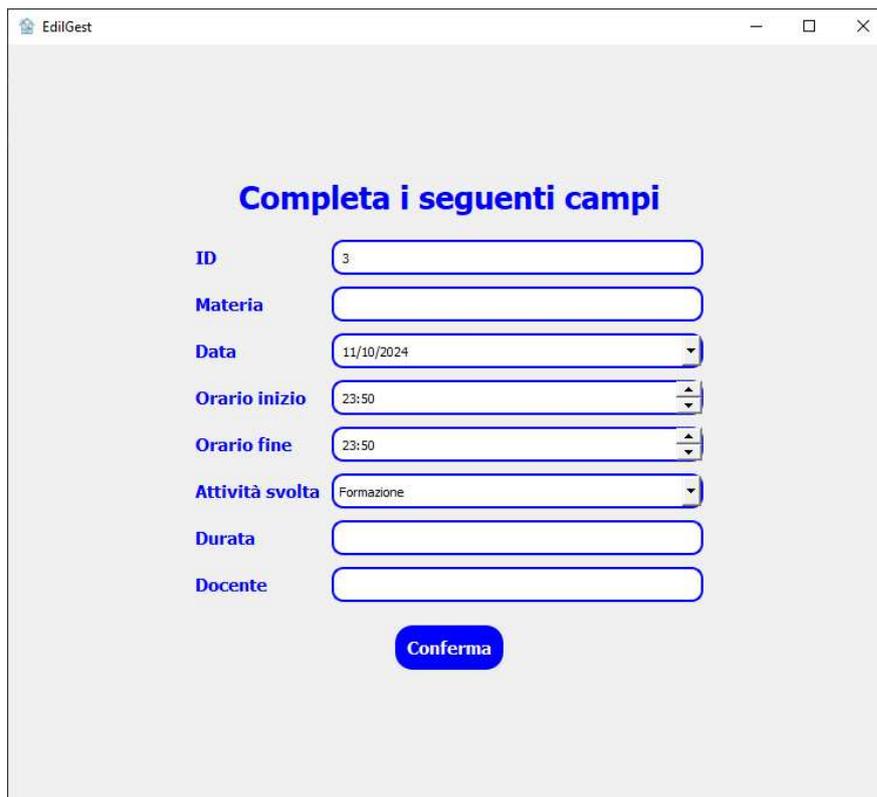
- *visualizzare il dettaglio del corso selezionato*, attraverso il click sul pulsante "Modifica"; visualizza la schermata "Vista corso di sicurezza", mostrata in Figura 5.28.

Nella schermata "Vista corso di sicurezza", mostrata in Figura 5.28, l'utente può confermare le modifiche effettuate, attraverso il click sul pulsante "Conferma modifica", oppure eliminare il corso selezionato, cliccando sul pulsante "Elimina".

5.8 Sezione "Visite mediche"

Cliccando sull'opzione "Visite mediche" del menù a tendina associato al pulsante "Sicurezza" della Home Page, in Figura 5.3, l'utente visualizza la schermata "Lista visite mediche"; qui vede una lista contenente gli ID, le date e i dipendenti di tutte le visite mediche presenti nel sistema.

Cliccando sul pulsante "Elimina tutto", l'utente può eliminare, previa conferma, per evitare click accidentali, tutte le visite mediche presenti nel sistema. I rispettivi dipendenti non verranno eliminati.

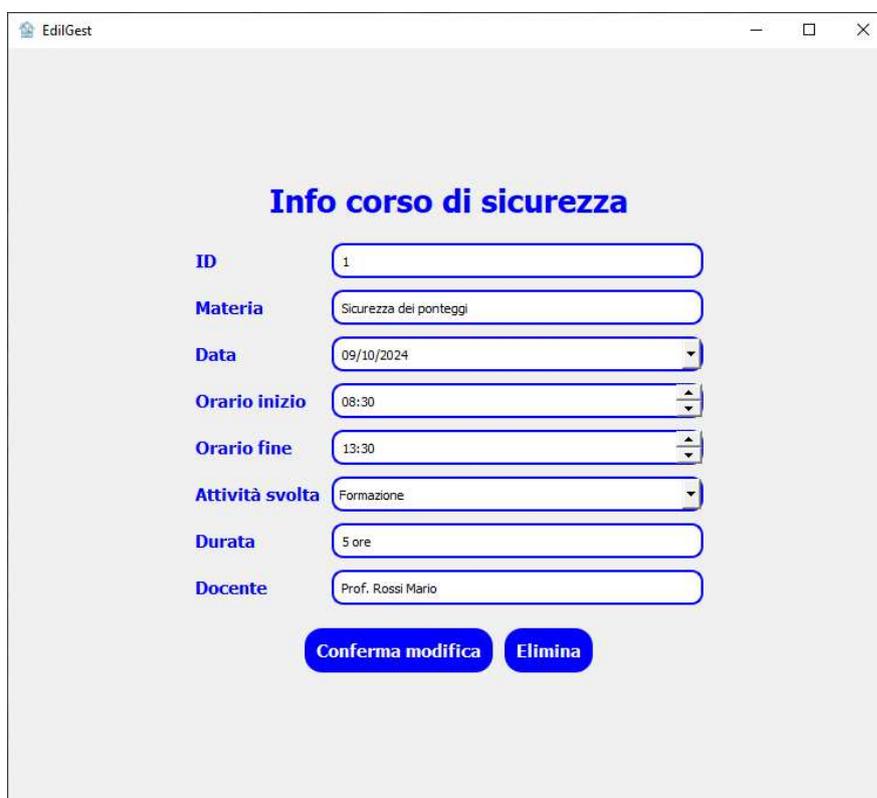


The screenshot shows a web form titled "Completa i seguenti campi" (Complete the following fields) in a window titled "EdilGest". The form contains the following fields:

ID	<input type="text" value="3"/>
Materia	<input type="text"/>
Data	<input type="text" value="11/10/2024"/>
Orario inizio	<input type="text" value="23:50"/>
Orario fine	<input type="text" value="23:50"/>
Attività svolta	<input type="text" value="Formazione"/>
Durata	<input type="text"/>
Docente	<input type="text"/>

At the bottom of the form is a blue button labeled "Conferma".

Figura 5.27: Schermata per l'inserimento di un nuovo corso di sicurezza sul lavoro



The screenshot shows a web form titled "Info corso di sicurezza" (Safety course info) in a window titled "EdilGest". The form contains the following fields:

ID	<input type="text" value="1"/>
Materia	<input type="text" value="Sicurezza dei ponteggi"/>
Data	<input type="text" value="09/10/2024"/>
Orario inizio	<input type="text" value="08:30"/>
Orario fine	<input type="text" value="13:30"/>
Attività svolta	<input type="text" value="Formazione"/>
Durata	<input type="text" value="5 ore"/>
Docente	<input type="text" value="Prof. Rossi Mario"/>

At the bottom of the form are two blue buttons: "Conferma modifica" and "Elimina".

Figura 5.28: Schermata relativa ai dettagli del corso di sicurezza sul lavoro selezionato

Cliccando sull'opzione "Cerca per dipendente" del menù a tendina associato al pulsante "Filtri", apparirà una schermata pop-up, come mostrato in Figura 5.30, dove l'utente può selezionare il dipendente di cui vuole visualizzare le visite mediche; al click dell'opzione "Ok", il sistema visualizza la lista di tutte le visite mediche associate al dipendente selezionato.



Figura 5.29: Schermata relativa alla lista delle visite mediche

Cliccando sul pulsante "Aggiungi" della schermata "Lista visite mediche", mostrata in Figura 5.29, l'utente può visualizzare la pagina di inserimento di una nuova visita medica, mostrata in Figura 5.31.

Una volta completati correttamente tutti i campi, l'utente può completare l'inserimento della visita medica cliccando sul pulsante "Conferma". Al termine dell'operazione, il sistema visualizza nuovamente la schermata "Lista visite mediche", mostrata in Figura 5.29, aggiornata.

Dopo aver selezionato una visita medica nella schermata "Lista visite mediche", mostrata in Figura 5.29, l'utente può effettuare due azioni:

- *visualizzare il dettaglio della visita medica selezionata*, attraverso il click sul pulsante "Modifica"; visualizza la schermata "Vista visita medica", mostrata in Figura 5.32;
- *contattare il dipendente della visita medica selezionata*, attraverso il click sul pulsante "Contatta dipendente"; apre una nuova e-mail da inviare con il campo "Destinatario" precompilato con la mail del dipendente della visita medica selezionata.

Nella schermata "Vista visita medica", mostrata in Figura 5.32, l'utente può confermare le modifiche effettuate, attraverso il click sul pulsante "Conferma modifica", oppure eliminare la visita medica selezionata, cliccando sul pulsante "Elimina".



Figura 5.30: Schermata di pop-up che permette il filtraggio delle visite mediche per dipendente

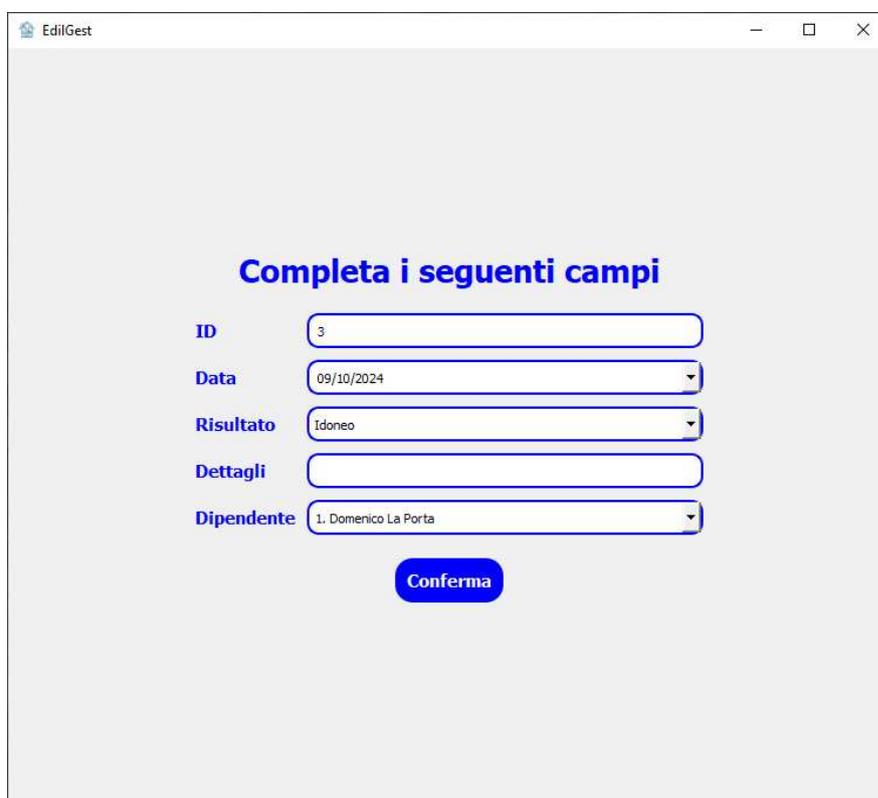


Figura 5.31: Schermata per l'inserimento di una nuova visita medica

The screenshot shows a web application window titled 'EdilGest'. The main content area is titled 'Info visita medica' in blue. Below the title, there are five rows of form fields, each with a label on the left and a control on the right:

- ID**: A text input field containing the value '1'.
- Data**: A date picker dropdown menu showing '09/10/2024'.
- Risultato**: A dropdown menu showing 'Idoneo'.
- Dettagli**: A text input field containing 'Il dipendente risulta idoneo'.
- Dipendente**: A dropdown menu showing '1. Domenico La Porta'.

At the bottom of the form, there are two blue buttons: 'Conferma modifica' and 'Elimina'.

Figura 5.32: Schermata relativa ai dettagli della visita medica selezionata

5.9 Sezione "Fornitori"

Cliccando sull'opzione "Lista fornitori" del menù a tendina associato al pulsante "Fornitori" della Home Page, in Figura 5.3, precisamente nel menù in alto, l'utente visualizza la schermata "Lista fornitori", mostrata in Figura 5.33; qui vede una lista contenente i nomi e i titolari di tutti i clienti presenti nel sistema.

Cliccando sul pulsante "Elimina tutto", l'utente può eliminare, previa conferma per evitare click accidentali, tutti i fornitori presenti nel sistema.

Cliccando sull'opzione "Cerca per nome" del menù a tendina associato al pulsante "Filtri", apparirà una schermata pop-up, come mostrato in Figura 5.34, dove l'utente può digitare il nome desiderato; al click dell'opzione "Ok", il sistema visualizza il fornitore con il nome inserito.

Cliccando sull'opzione "Cerca per tipologia" del menù a tendina associato al pulsante "Filtri", apparirà una schermata di pop-up, come mostrato in Figura 5.35, dove l'utente può selezionare la tipologia desiderata; al click dell'opzione "Ok", il sistema visualizza la lista di tutti i fornitori della tipologia selezionata.

Cliccando sul pulsante "Aggiungi" della schermata "Lista fornitori", mostrata in Figura 5.33, l'utente può visualizzare la schermata di inserimento di un nuovo fornitore, mostrata in Figura 5.36.

Una volta completati correttamente tutti i campi, l'utente può completare l'inserimento del fornitore cliccando sul pulsante "Conferma". Al termine dell'operazione, il sistema visualizza nuovamente la schermata "Lista fornitori", mostrata in Figura 5.33, aggiornata.

Dopo aver selezionato un fornitore nella schermata "Lista fornitori", mostrata in Figura 5.33, l'utente può effettuare due azioni:

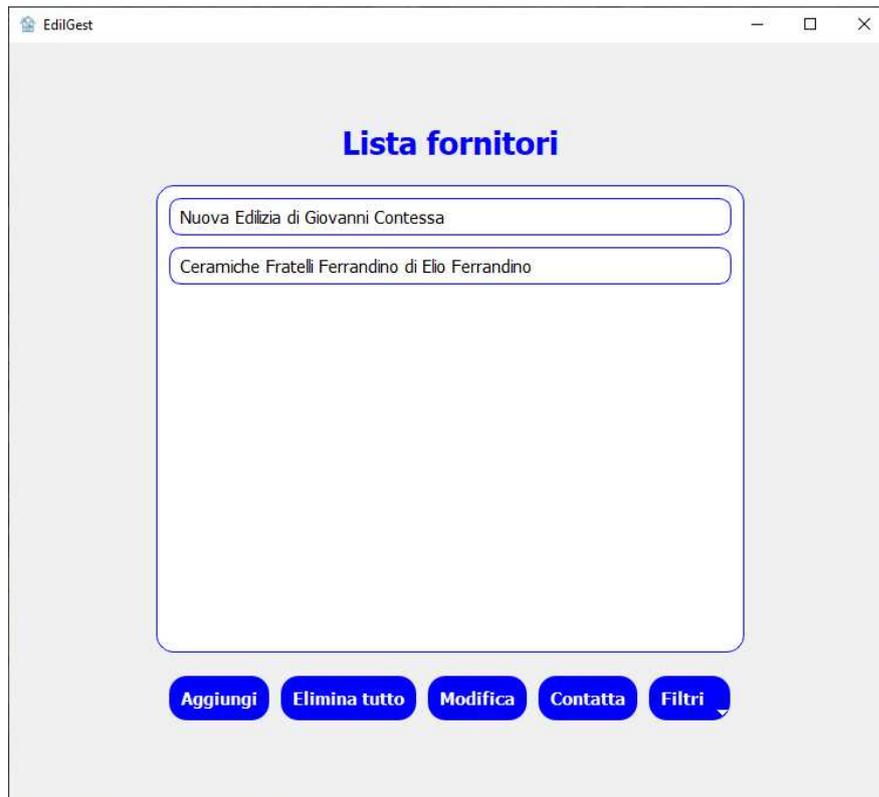


Figura 5.33: Schermata relativa alla lista dei fornitori

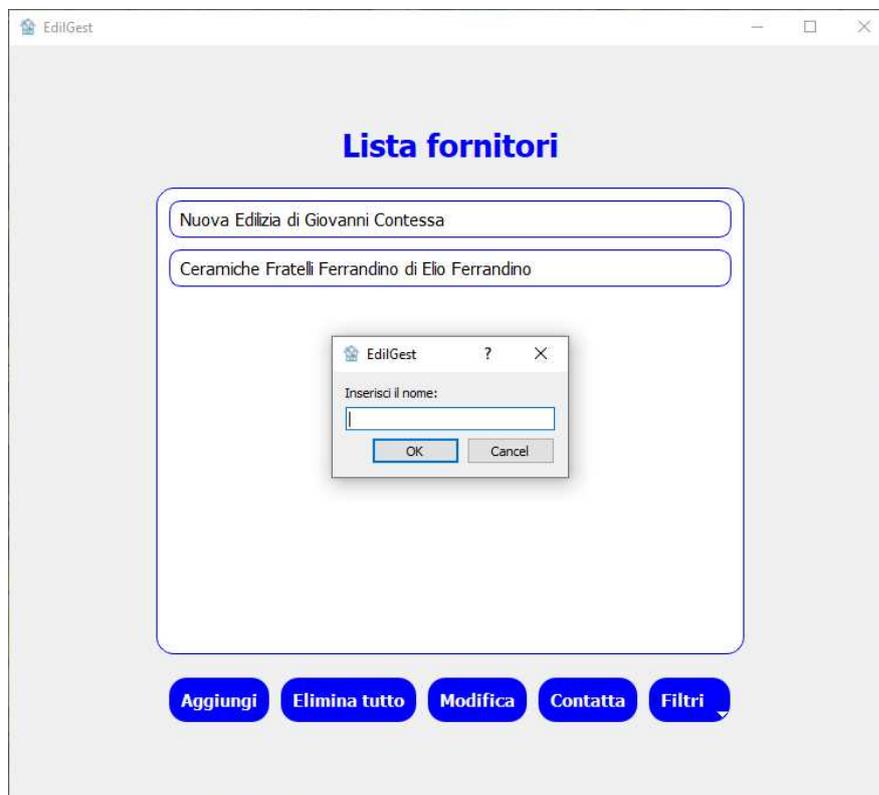


Figura 5.34: Schermata di pop-up che permette la ricerca di un fornitore per nome

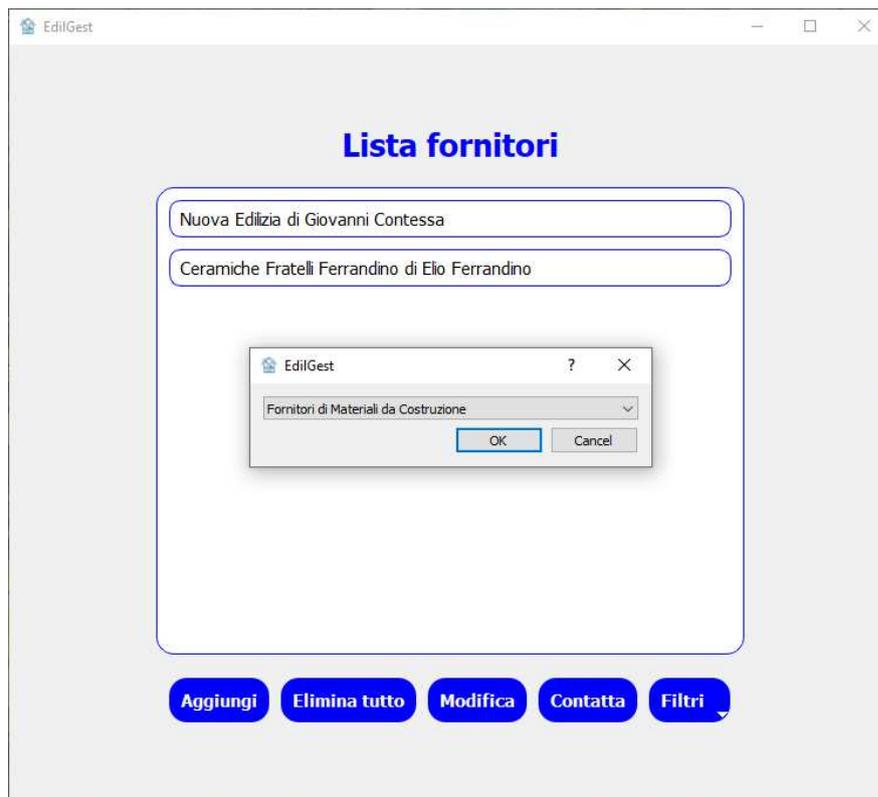


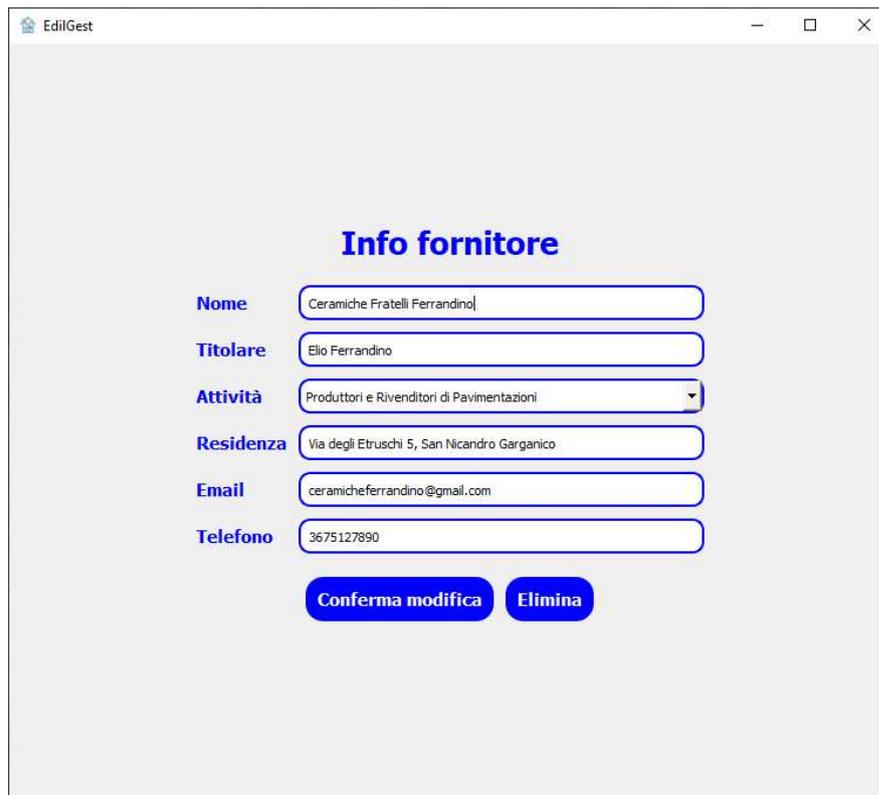
Figura 5.35: Schermata di pop-up che permette il filtraggio dei fornitori per tipologia

The screenshot shows a web application window titled 'EdilGest'. The main content area is titled 'Completa i seguenti campi'. It contains a form with the following fields: 'Nome' (text input), 'Titolare' (text input), 'Attività' (dropdown menu with 'Fornitori di Materiali da Costruzione' selected), 'Residenza' (text input), 'Email' (text input), and 'Telefono' (text input). A blue button labeled 'Conferma' is positioned below the form.

Figura 5.36: Schermata di inserimento di un nuovo fornitore

- *visualizzare il dettaglio del fornitore selezionato*, attraverso il click sul pulsante "Modifica"; visualizza la schermata "Vista fornitore", mostrata in Figura 5.37;
- *contattare il fornitore selezionato*, attraverso il click sul pulsante "Contatta"; apre una nuova e-mail da inviare con il campo "Destinatario" precompilato con la mail del fornitore selezionato.

Nella schermata "Vista fornitore", mostrata in Figura 5.37, l'utente può confermare le modifiche effettuate, attraverso il click sul pulsante "Conferma modifica", oppure eliminare il fornitore selezionato, cliccando sul pulsante "Elimina".



The screenshot shows a web browser window titled "EdilGest" with a standard window control bar. The main content area displays a form titled "Info fornitore" in blue text. The form consists of several input fields, each with a label to its left: "Nome" (Ceramiche Fratelli Ferrandino), "Titolare" (Elio Ferrandino), "Attività" (a dropdown menu showing "Produttori e Rivenditori di Pavimentazioni"), "Residenza" (Via degli Etruschi 5, San Nicandro Garganico), "Email" (ceramicheferrandino@gmail.com), and "Telefono" (3675127890). Below the input fields are two blue buttons: "Conferma modifica" and "Elimina".

Figura 5.37: Schermata relativa ai dettagli del fornitore selezionato

5.10 Sezione "Materie prime"

Cliccando sul pulsante "Magazzino" della Home Page, in Figura 5.3, l'utente visualizza la schermata "Magazzino", mostrata in Figura 5.38; qui vede una lista contenente i nomi e le quantità di tutte le materie prime presenti nel magazzino.

Dopo aver selezionato un prodotto nella schermata "Magazzino", mostrata in Figura 5.38, l'utente può effettuare una sola azione:

- *consuma materia prima selezionata*, attraverso il click sul pulsante "Consuma materie prime"; visualizza una schermata pop-up, come vediamo in Figura 5.39, dove può inserire la quantità di prodotto da consumare; al click dell'opzione "Ok", il sistema ridurrà la quantità del prodotto selezionato, secondo quanto indicato dall'utente.

Cliccando sul pulsante "Acquista materie prime" della schermata "Magazzino", mostrata in Figura 5.38, l'utente può visualizzare la schermata di acquisto di nuove scorte di materie

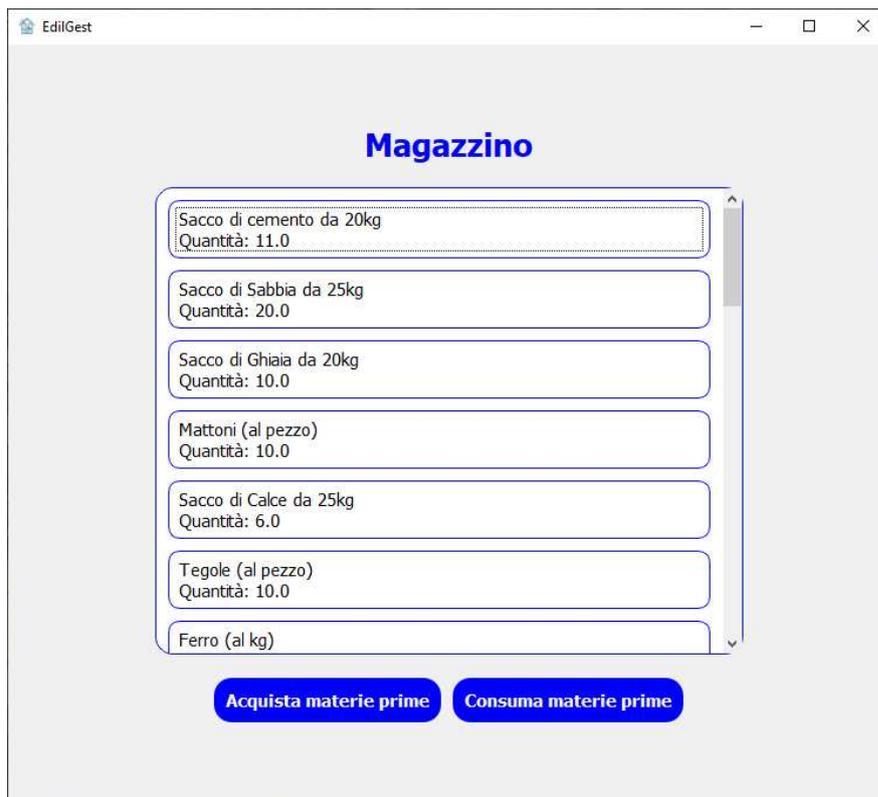


Figura 5.38: Schermata relativa alle scorte di materie prime presenti in magazzino

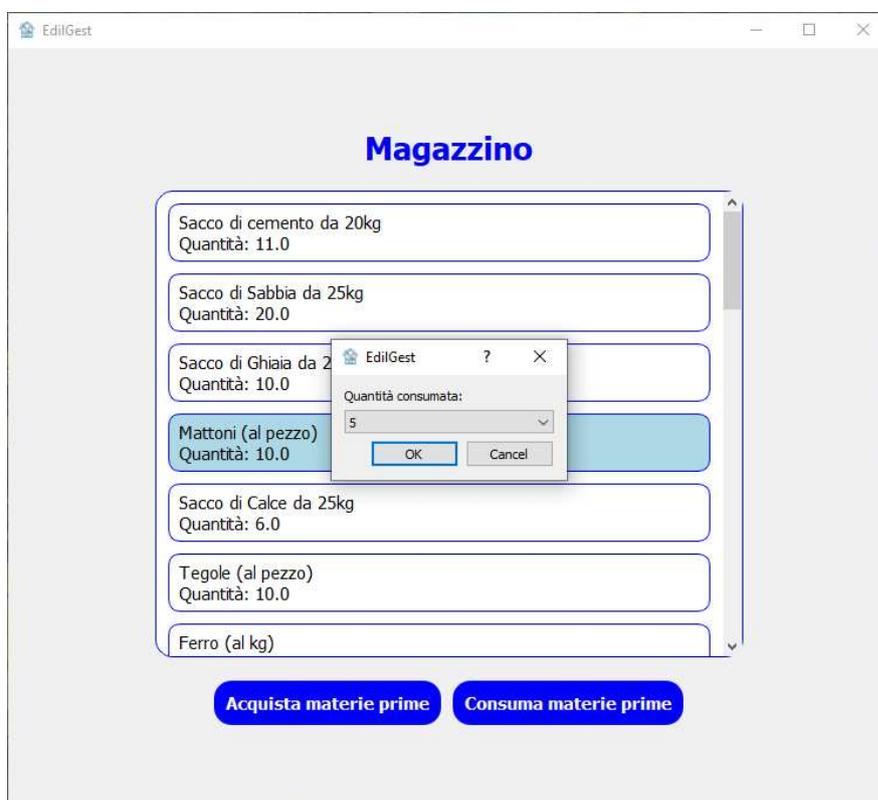


Figura 5.39: Schermata di pop-up che permette il consumo della materia prima selezionata

prime, mostrata in Figura 5.40, che contiene una lista dei prodotti in vendita, da una parte, e la lista della spesa, dall'altra.

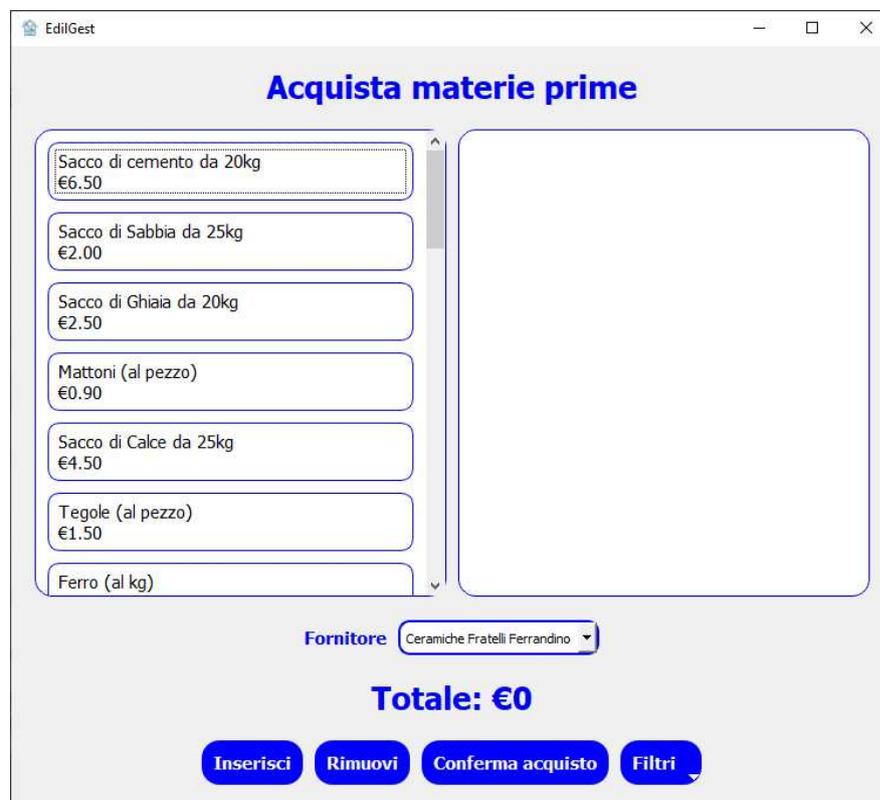


Figura 5.40: Schermata relativa all'acquisto di nuove scorte di materie prime (lista della spesa vuota)

Cliccando sull'opzione "Cerca prodotto per nome" del menù a tendina associato al pulsante "Filtri", apparirà una schermata di pop-up, come mostrato in Figura 5.41, dove l'utente può inserire il nome del prodotto che desidera acquistare; al click dell'opzione "Ok", il sistema visualizza il prodotto desiderato nella sezione relativa ai prodotti in vendita.

Dopo aver selezionato un prodotto nella sezione relativa alle materie prime in vendita, l'utente può effettuare una sola azione:

- *inserire il prodotto selezionato nella lista della spesa*, attraverso il click sul pulsante "Inserisci"; visualizza una schermata di pop-up in cui seleziona la quantità di scorte da acquistare del prodotto selezionato, come vediamo in Figura 5.42; al click dell'opzione "Ok", il sistema inserisce il prodotto desiderato, con la quantità selezionata, nella lista della spesa e aggiorna il costo totale.

Dopo aver selezionato un prodotto nella lista della spesa, come vediamo in Figura 5.43, l'utente può effettuare una sola azione:

- *rimuovere il prodotto selezionato dalla lista della spesa*, attraverso il click sul pulsante "Rimuovi"; rimuove il prodotto selezionato dalla lista della spesa e aggiorna il totale.

Dopo aver riempito la lista della spesa e selezionato il fornitore da cui si acquista, cliccando sul pulsante "Conferma acquisto" della schermata "Acquista materie prime", mostrata in Figura 5.43, l'utente può confermare l'acquisto delle materie prime presenti nella lista della spesa. Il sistema aggiorna le scorte di materie prime presenti in magazzino e inserisce l'uscita a partire dal costo totale della spesa.



Figura 5.41: Schermata di pop-up che permette la ricerca per nome di una specifica materia prima

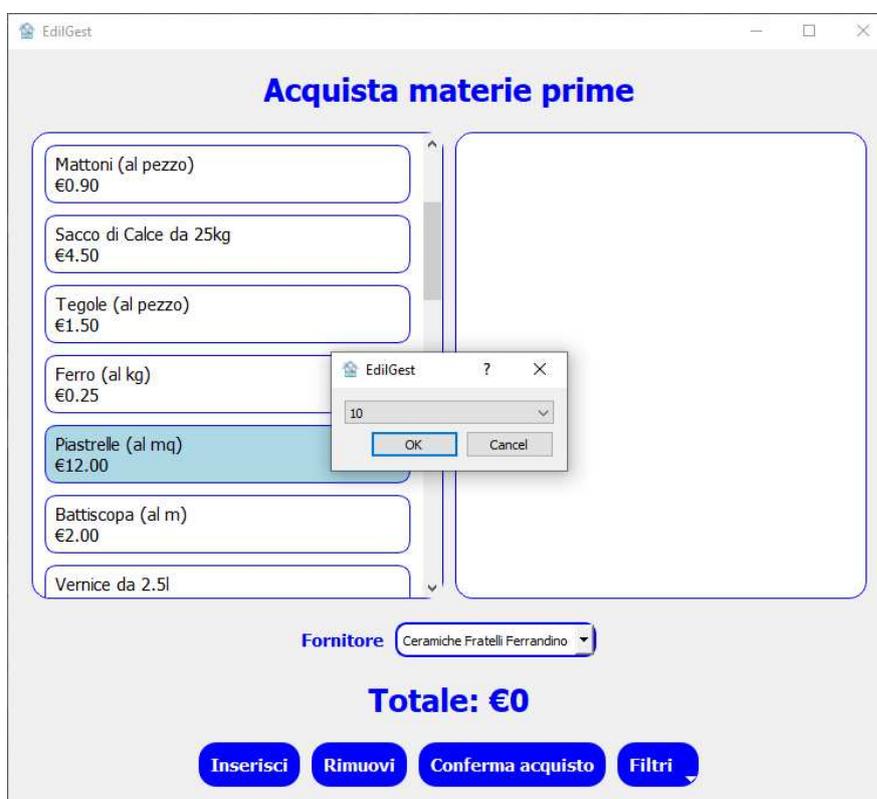


Figura 5.42: Schermata di pop-up che permette l'inserimento nella lista della spesa della materia prima selezionata



Figura 5.43: Schermata relativa all'acquisto di nuove scorte di materie prime (lista della spesa non vuota)

5.11 Sezione "Locali"

Cliccando sul pulsante "Locali" della Home Page, in Figura 5.3, l'utente visualizza la schermata "Lista locali"; qui vede una lista contenente le tipologie e l'indirizzo di tutti i locali presenti nel sistema.

Cliccando sull'opzione "Cerca per tipo" del menù a tendina associato al pulsante "Filtri", apparirà una schermata pop-up, come mostrato in Figura 5.45, dove l'utente può selezionare la tipologia di locale desiderata; al click dell'opzione "Ok", il sistema visualizza la lista di tutti i locali della tipologia selezionata.

Cliccando sul pulsante "Aggiungi" della schermata "Lista locali", mostrata in Figura 5.44 l'utente può visualizzare la pagina di inserimento di un nuovo locale, mostrata in Figura 5.46.

Una volta completati correttamente tutti i campi, l'utente può completare l'inserimento del locale cliccando sul pulsante "Conferma". Al termine dell'operazione, il sistema visualizza nuovamente la schermata "Lista locali", mostrata in Figura 5.44, aggiornata.

Dopo aver selezionato un locale nella schermata "Lista locali", mostrata in Figura 5.47, l'utente può effettuare due azioni:

- *visualizzare il dettaglio del locale selezionato*, attraverso il click sul pulsante "Modifica"; visualizza la schermata "Vista locale", mostrata in Figura 5.47;
- *visualizzare la lista dei locali presenti nel locale selezionato*, attraverso il click sul pulsante "Lista veicoli"; apre la schermata "Lista veicoli", mostrata in Figura 5.51, dove vengono visualizzati soltanto i veicoli presenti nel locale selezionato (inoltre, è presente soltanto il pulsante "Modifica").

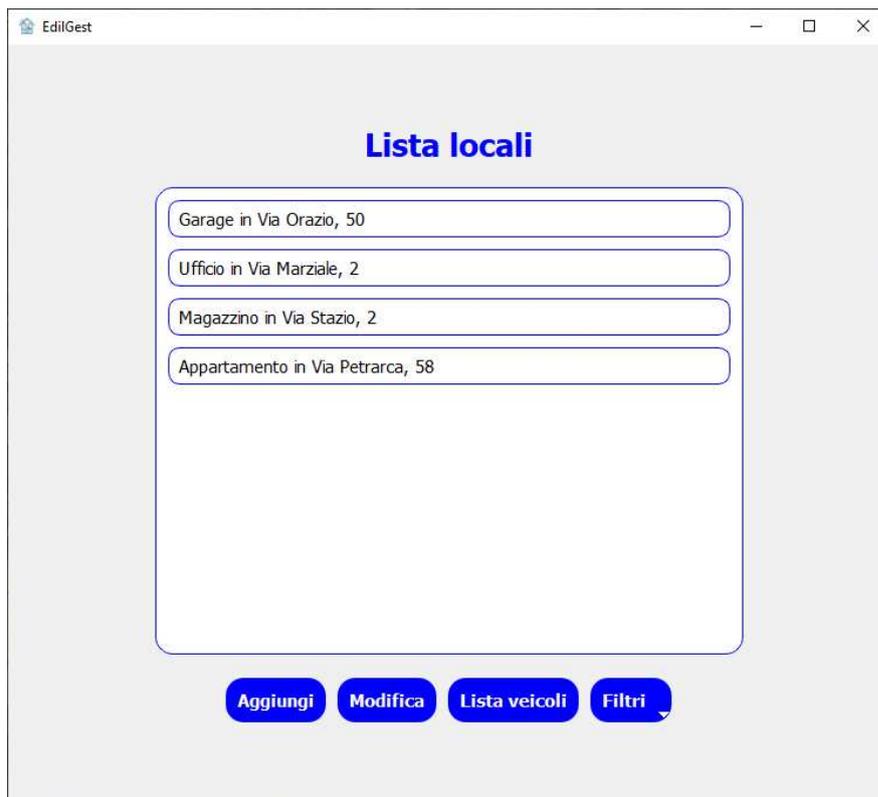


Figura 5.44: Schermata relativa alla lista dei locali

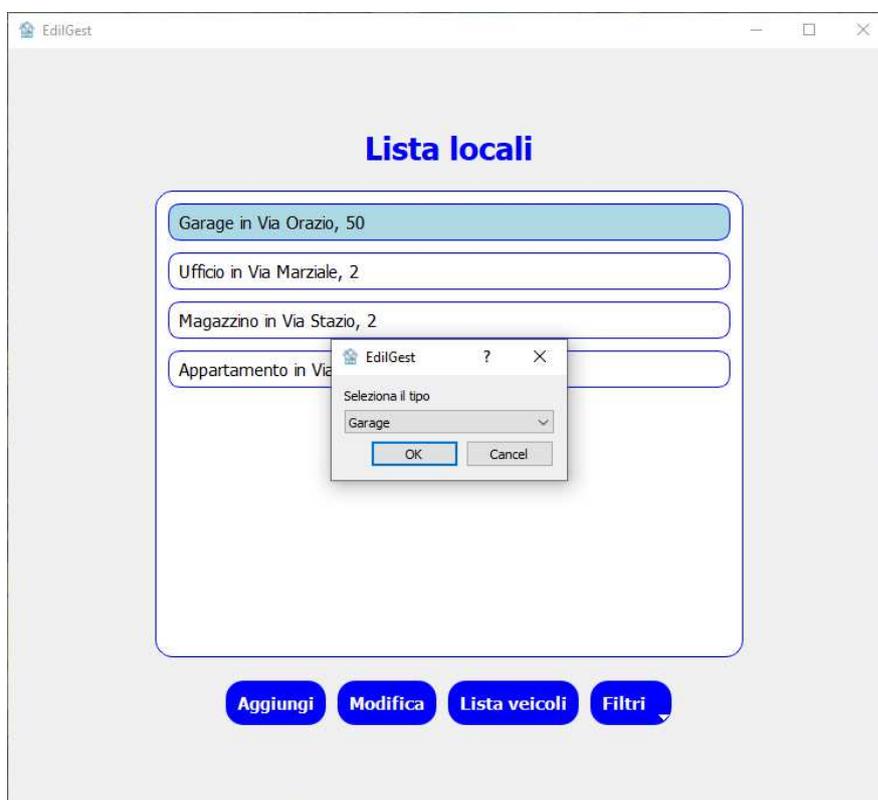
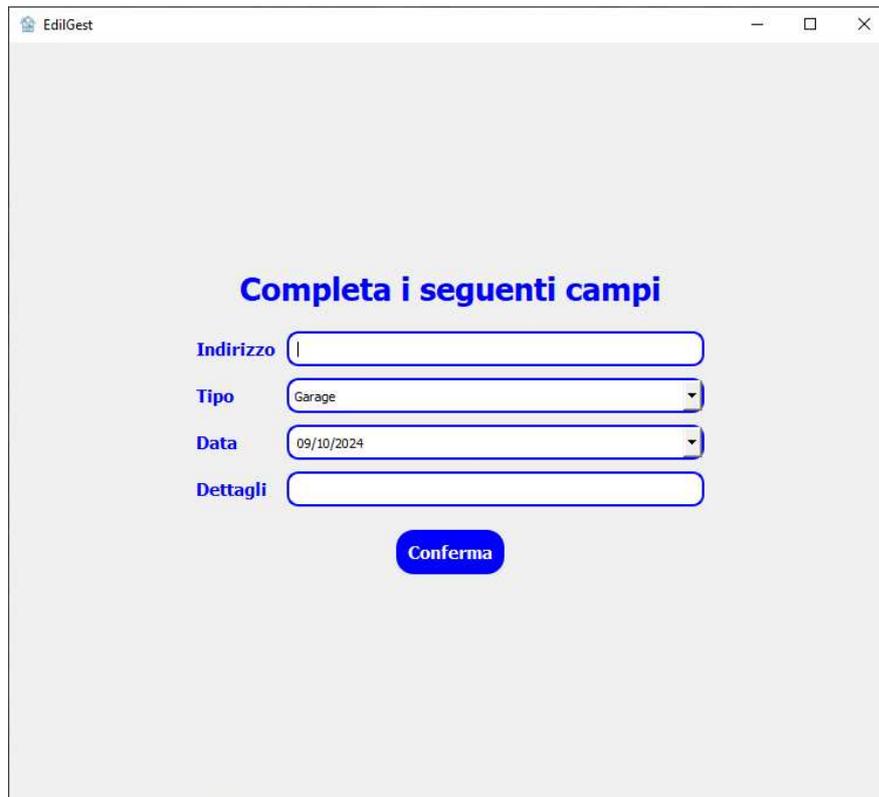


Figura 5.45: Schermata di pop-up che permette il filtraggio dei locali per tipologia



The screenshot shows a web browser window titled "EdilGest". The main heading is "Completa i seguenti campi" in blue. Below it are four input fields: "Indirizzo" (text input), "Tipo" (dropdown menu with "Garage" selected), "Data" (dropdown menu with "09/10/2024" selected), and "Dettagli" (text input). A blue "Conferma" button is centered below the fields.

Figura 5.46: Schermata per l'inserimento di un nuovo locale

Nella schermata "Vista locale", mostrata in Figura 5.47, l'utente può confermare le modifiche effettuate, attraverso il click sul pulsante "Conferma modifica", eliminare il locale selezionato, cliccando sul pulsante "Elimina", inserire un'uscita, attraverso il click su una delle due opzioni del menù a tendina associato al pulsante "Pagamento in uscita", come mostrato in Figura 5.49, oppure inserire un'entrata, attraverso il click su una delle due opzioni del menù a tendina associato al pulsante "Pagamento in entrata", come mostrato in Figura 5.50.

L'eliminazione non va a buon fine se il locale selezionato è occupato da almeno un veicolo; mostriamo la schermata di eliminazione non riuscita nella Figura 5.48.

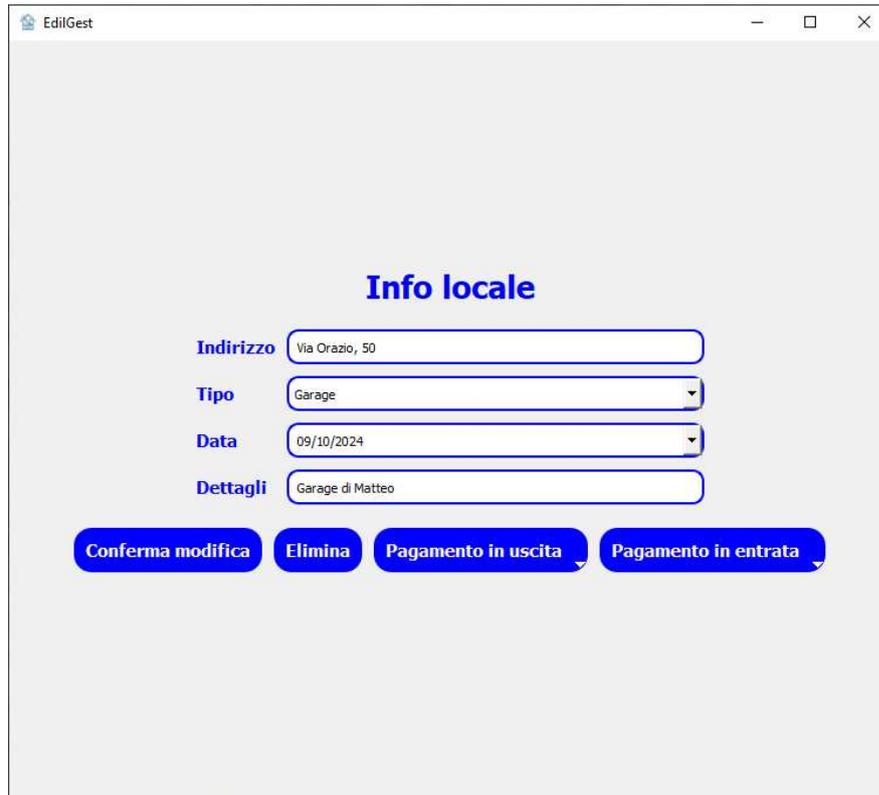
5.12 Sezione "Veicoli"

Cliccando sull'opzione "Lista veicoli" del menù a tendina associato al pulsante "Veicoli" della Home Page, in Figura 5.3, precisamente nel menù in alto, l'utente visualizza la schermata "Lista veicoli", mostrata in Figura 5.51; qui vede una lista contenente le tipologie e le targhe di tutti i veicoli presenti nel sistema.

Cliccando sul pulsante "Elimina tutto", l'utente può eliminare, previa conferma, per evitare click accidentali, tutti i corsi di sicurezza presenti nel sistema.

Cliccando sull'opzione "Cerca per tipologia" del menù a tendina associato al pulsante "Filtri", apparirà una schermata di pop-up, come mostrato in Figura 5.52, dove l'utente può selezionare la tipologia di veicolo desiderata; al click dell'opzione "Ok", il sistema visualizza la lista di tutti i veicoli della tipologia selezionata.

Cliccando sull'opzione "Cerca per locale" del menù a tendina associato al pulsante "Filtri", apparirà una schermata di pop-up, come mostrato in Figura 5.53, dove l'utente può selezionare il locale desiderato; al click dell'opzione "Ok", il sistema visualizza la lista di tutti i veicoli presenti nel locale selezionato.



EdilGest

Info locale

Indirizzo

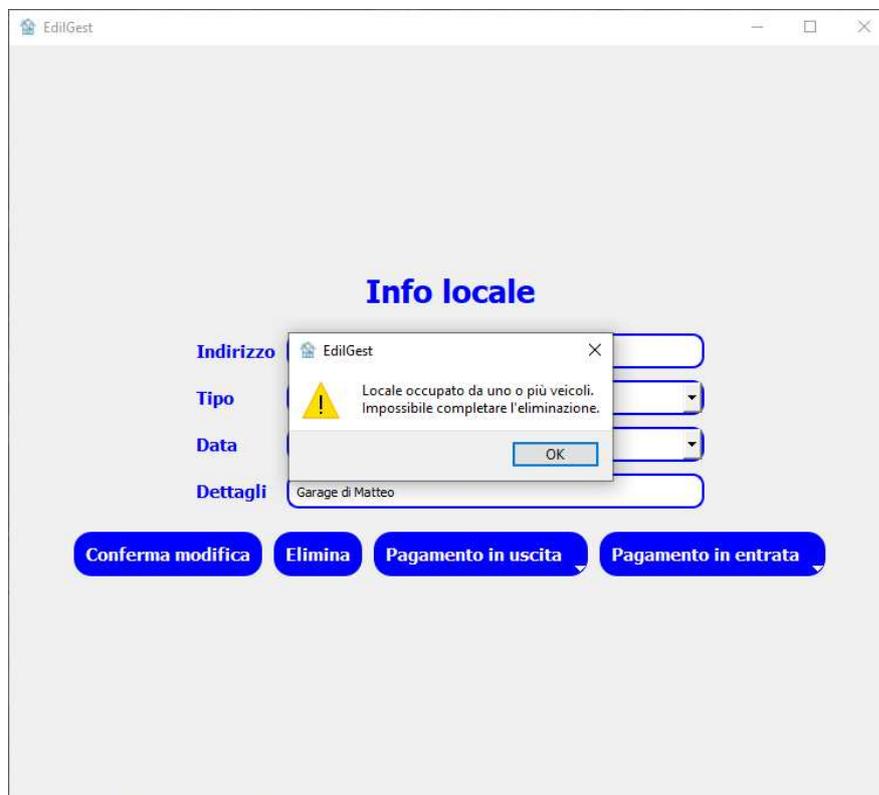
Tipo

Data

Dettagli

Conferma modifica **Elimina** **Pagamento in uscita** **Pagamento in entrata**

Figura 5.47: Schermata relativa ai dettagli del locale selezionato



EdilGest

Info locale

Indirizzo

Tipo

Data

Dettagli

Conferma modifica **Elimina** **Pagamento in uscita** **Pagamento in entrata**

EdilGest

Locale occupato da uno o più veicoli.
Impossibile completare l'eliminazione.

OK

Figura 5.48: Avviso "Locale occupato"



The screenshot shows a web application window titled "EdilGest". The main content area is titled "Info locale" in blue. Below the title, there are four input fields: "Indirizzo" with the value "Via Orazio, 50", "Tipo" with a dropdown menu showing "Garage", "Data" with a dropdown menu showing "09/10/2024", and "Dettagli" with the value "Garage di Matteo". Below these fields, there are four blue buttons: "Conferma modifica", "Elimina", "Pagamento in uscita", and "Pagamento in entrata". The "Pagamento in uscita" button has a dropdown menu open, showing three options: "Acquisto", "Affitto", and "Bollette".

Figura 5.49: Menù che permette di selezionare il tipo di pagamento in uscita che riguarda il locale selezionato

Cliccando sul pulsante "Aggiungi" della schermata "Lista veicoli", mostrata in Figura 5.51, l'utente può visualizzare la pagina di inserimento di un nuovo veicolo, mostrata in Figura 5.54.

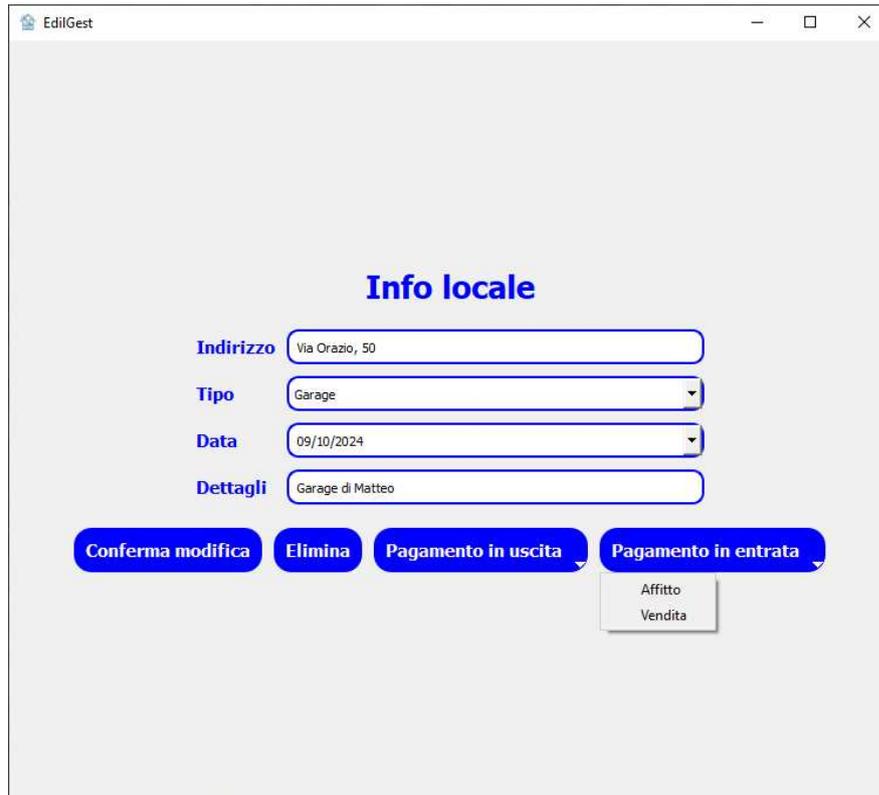
Tale pagina viene aperta soltanto se sono presenti locali dove poter posizionare il nuovo veicolo; in caso contrario, il sistema segnala l'errore, come vediamo in Figura 5.55, e impedisce la visualizzazione della schermata di inserimento.

Una volta completati correttamente tutti i campi, l'utente può completare l'inserimento del veicolo cliccando sul pulsante "Conferma". Al termine dell'operazione, il sistema visualizza nuovamente la schermata "Lista veicoli", mostrata in Figura 5.51, aggiornata.

Dopo aver selezionato un veicolo nella schermata "Lista veicoli", mostrata in Figura 5.51, l'utente può effettuare una sola azione:

- *visualizzare il dettaglio del veicolo selezionato*, attraverso il click sul pulsante "Modifica"; visualizza la schermata "Vista veicolo", mostrata in Figura 5.56.

Nella schermata "Vista veicolo", mostrata in Figura 5.56, l'utente può confermare le modifiche effettuate, attraverso il click sul pulsante "Conferma modifica", eliminare il veicolo selezionato, cliccando sul pulsante "Elimina", inserire un'uscita, attraverso il click su una delle tre opzioni del menù a tendina associato al pulsante "Pagamento in uscita", come mostrato in Figura 5.57, oppure inserire un'entrata, attraverso il click su una delle due opzioni del menù a tendina associato al pulsante "Pagamento in entrata", come mostrato in Figura 5.58.



EdilGest

Info locale

Indirizzo

Tipo

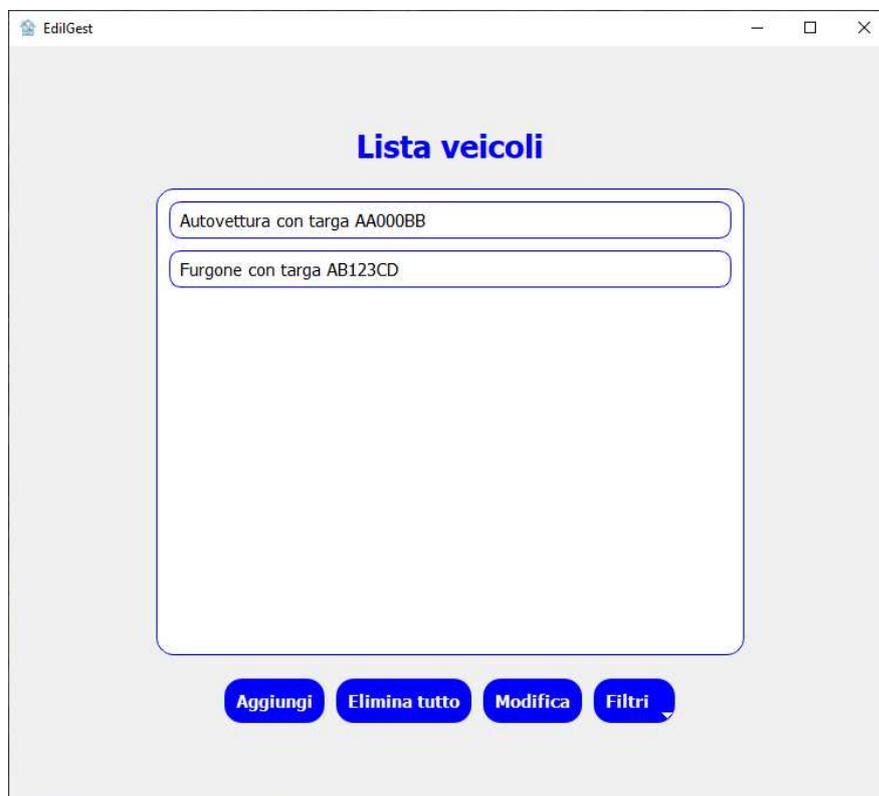
Data

Dettagli

Conferma modifica **Elimina** **Pagamento in uscita** **Pagamento in entrata**

- Affitto
- Vendita

Figura 5.50: Menù che permette di selezionare il tipo di pagamento in entrata che riguarda il locale selezionato



EdilGest

Lista veicoli

Autovettura con targa AA000BB

Furgone con targa AB123CD

Aggiungi **Elimina tutto** **Modifica** **Filtri**

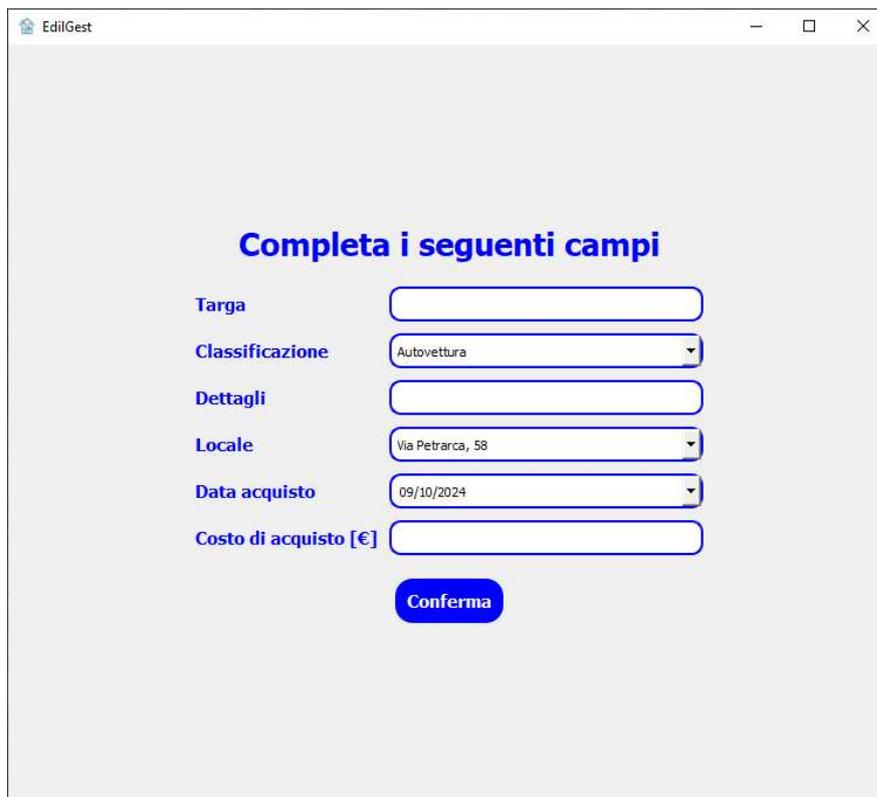
Figura 5.51: Schermata relativa alla lista dei veicoli



Figura 5.52: Schermata di pop-up che permette il filtraggio dei veicoli per tipologia



Figura 5.53: Schermata di pop-up che permette il filtraggio dei veicoli per locale



The screenshot shows a web application window titled 'EdilGest'. The main heading is 'Completa i seguenti campi' (Complete the following fields). Below this, there are several input fields and dropdown menus:

- Targa**: A text input field.
- Classificazione**: A dropdown menu with 'Autovettura' selected.
- Dettagli**: A text input field.
- Locale**: A dropdown menu with 'Via Petrarca, 58' selected.
- Data acquisto**: A date dropdown menu with '09/10/2024' selected.
- Costo di acquisto [€]**: A text input field.

At the bottom of the form is a blue button labeled 'Conferma' (Confirm).

Figura 5.54: Schermata per l'inserimento di un nuovo veicolo

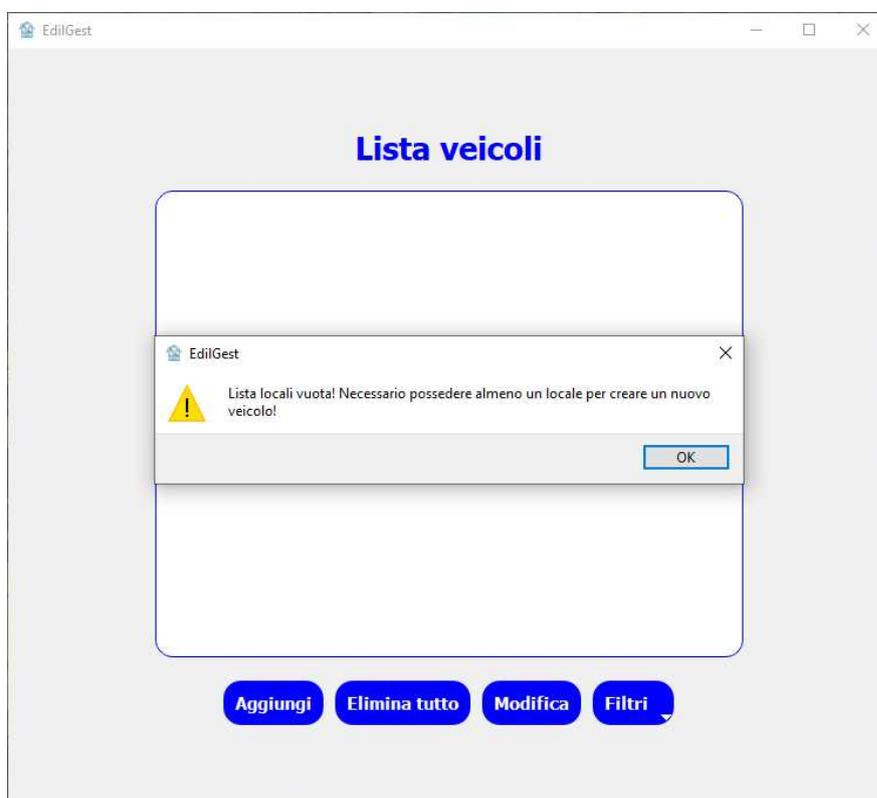


Figura 5.55: Avviso "Lista locali vuota"



The screenshot shows a web application window titled 'EdilGest'. The main heading is 'Info veicolo'. Below it, there are several input fields and dropdown menus:

- Targa:** Text input field containing 'AA000BB'.
- Classificazione:** Dropdown menu with 'Autovettura' selected.
- Dettagli:** Text input field containing 'Macchina aziendale'.
- Locale:** Dropdown menu with 'Via Stazio, 2' selected.
- Data acquisto:** Dropdown menu with '09/10/2024' selected.
- Costo di acquisto [€]:** Text input field containing '10000.00'.

At the bottom, there are four blue buttons: 'Conferma modifica', 'Elimina', 'Pagamento in uscita', and 'Pagamento in entrata'.

Figura 5.56: Schermata relativa ai dettagli del veicolo selezionato



This screenshot is identical to the previous one, but the 'Pagamento in uscita' button is active, and its dropdown menu is open, showing three options:

- Acquisto
- Noleggio
- Costi di mantenimento

Figura 5.57: Menù che permette di selezionare il tipo di pagamento in uscita che riguarda il veicolo selezionato

The screenshot shows a web application window titled 'EdilGest'. The main content area is titled 'Info veicolo'. It contains several form fields:

- Targa:** Input field with value 'AA000BB'.
- Classificazione:** Dropdown menu with value 'Autovettura'.
- Dettagli:** Input field with value 'Macchina aziendale'.
- Locale:** Dropdown menu with value 'Via Stazio, 2'.
- Data acquisto:** Dropdown menu with value '09/10/2024'.
- Costo di acquisto [€]:** Input field with value '10000.00'.

Below the form are four blue buttons: 'Conferma modifica', 'Elimina', 'Pagamento in uscita', and 'Pagamento in entrata'. The 'Pagamento in entrata' button has a dropdown menu open, showing two options: 'Noleggio' and 'Vendita'.

Figura 5.58: Menù che permette di selezionare il tipo di pagamento in entrata che riguarda il veicolo selezionato

5.13 Sezione "Entrate"

Cliccando sull'opzione "Entrate" del menù a tendina associato al pulsante "Contabilità" della Home Page, in Figura 5.3, l'utente visualizza la schermata "Lista entrate", mostrata in Figura 5.59; qui vede una lista contenente gli importi e le provenienze di tutte le entrate presenti nel sistema.

Cliccando sul pulsante "Elimina tutto", l'utente può eliminare, previa conferma, per evitare click accidentali, tutte le entrate presenti nel sistema.

Cliccando sull'opzione "Cerca per provenienza" del menù a tendina associato al pulsante "Filtri", apparirà una schermata di pop-up, come mostrato in Figura 5.60, dove l'utente può selezionare la provenienza desiderata; al click dell'opzione "Ok", il sistema visualizza la lista di tutte le entrate associate alla provenienza selezionata.

Cliccando sull'opzione "Cerca per data" del menù a tendina associato al pulsante "Filtri", apparirà una schermata di pop-up, come mostrato in Figura 5.61, dove l'utente può selezionare il periodo temporale desiderato; al click dell'opzione "Ok", il sistema visualizza la lista di tutte le entrate presenti nel sistema che sono state effettuate nel periodo indicato.

Cliccando sul pulsante "Aggiungi" della schermata "Lista entrate", mostrata in Figura 5.59, l'utente può visualizzare la pagina di inserimento di una nuova entrata, mostrata in Figura 5.62.

Una volta completati correttamente tutti i campi, l'utente può completare l'inserimento dell'entrata cliccando sul pulsante "Conferma". Al termine dell'operazione, il sistema visualizza nuovamente la schermata "Lista entrate", mostrata in Figura 5.59, aggiornata.

Dopo aver selezionato un'entrata nella schermata "Lista entrate", mostrata in Figura 5.59, l'utente può effettuare una sola azione:



Figura 5.59: Schermata relativa alla lista delle entrate



Figura 5.60: Schermata di pop-up che permette il filtraggio delle entrate per provenienza

Lista entrate

Entrata N. 1	Importo: €10000.00	Provenienza: Lavori
Entrata N. 2	Importo: €500.00	Provenienza: Vendite
Entrata N. 3	Importo: €10000.00	Provenienza: Finanziarie
Entrata N. 4	Importo: €600.00	Provenienza: Sovvenzioni e incentivi

Totale: €21100.00

Aggiungi **Elimina tutto** **Modifica** **Filtri**

EdilGest ? X

Da: 09/10/2024

A: 10/10/2024

OK **Cancel**

Figura 5.61: Schermata di pop-up che permette il filtraggio delle entrate per data

Completa i seguenti campi

Id

Data

Importo [€]

Causale

Dettagli

Provenienza

Conferma

Figura 5.62: Schermata per inserimento di una nuova entrata

- visualizzare il dettaglio dell'entrata selezionata, attraverso il click sul pulsante "Modifica"; visualizza la schermata "Vista entrata", mostrata in Figura 5.63.

Nella schermata "Vista entrata", mostrata in Figura 5.63, l'utente può confermare le modifiche effettuate, attraverso il click sul pulsante "Conferma modifica", oppure eliminare l'entrata selezionata, cliccando sul pulsante "Elimina".



The screenshot shows a web application window titled "EdilGest" with a close button. The main heading is "Info entrata". The form contains the following fields:

- Id**: Input field with value "1".
- Data**: Dropdown menu with value "09/10/2024".
- Importo [€]**: Input field with value "10000.00".
- Causale**: Text area containing the text: "Pagamento di €10000.0 ricevuto in data 09/10/2024 da parte del cliente: Simone Inzaghi, codice fiscale NZGSM72A01H985R per lavoro effettuato: Rifacimento bagno in Via Mazzini 19".
- Dettagli**: Input field (empty).
- Destinazione**: Dropdown menu with value "Lavori".

At the bottom of the form, there are two buttons: "Conferma modifica" and "Elimina".

Figura 5.63: Schermata relativa ai dettagli dell'entrata selezionata

5.14 Sezione "Uscite"

Cliccando sull'opzione "Uscite" del menù a tendina associato al pulsante "Contabilità" della Home Page, in Figura 5.3, l'utente visualizza la schermata "Lista uscite", mostrata in Figura 5.64; qui vede una lista contenente gli importi e le destinazioni di tutte le uscite presenti nel sistema.

Cliccando sul pulsante "Elimina tutto", l'utente può eliminare, previa conferma, per evitare click accidentali, tutte le uscite presenti nel sistema.

Cliccando sull'opzione "Cerca per destinazione" del menù a tendina associato al pulsante "Filtri", apparirà una schermata di pop-up, come mostrato in Figura 5.65, dove l'utente può selezionare la destinazione desiderata; al click dell'opzione "Ok", il sistema visualizza la lista di tutte le uscite associate alla destinazione selezionata.

Cliccando sull'opzione "Cerca per data" del menù a tendina associato al pulsante "Filtri", apparirà una schermata di pop-up, come mostrato in Figura 5.66, dove l'utente può selezionare il periodo temporale desiderato; al click dell'opzione "Ok", il sistema visualizza la lista di tutte le uscite presenti nel sistema che sono state effettuate nel periodo indicato.



Figura 5.64: Schermata relativa alla lista delle uscite



Figura 5.65: Schermata di pop-up che permette il filtraggio delle uscite per destinazione



Figura 5.66: Schermata di pop-up che permette il filtraggio delle uscite per data

Cliccando sul pulsante "Aggiungi" della schermata "Lista uscite", mostrata in Figura 5.64, l'utente può visualizzare la pagina di inserimento di una nuova uscita, mostrata in Figura 5.67.

Una volta completati correttamente tutti i campi, l'utente può completare l'inserimento dell'uscita cliccando sul pulsante "Conferma". Al termine dell'operazione, il sistema visualizza nuovamente la schermata "Lista uscite", mostrata in Figura 5.64, aggiornata.

Dopo aver selezionato un'uscita nella schermata "Lista uscite", mostrata in Figura 5.64, l'utente può effettuare una sola azione:

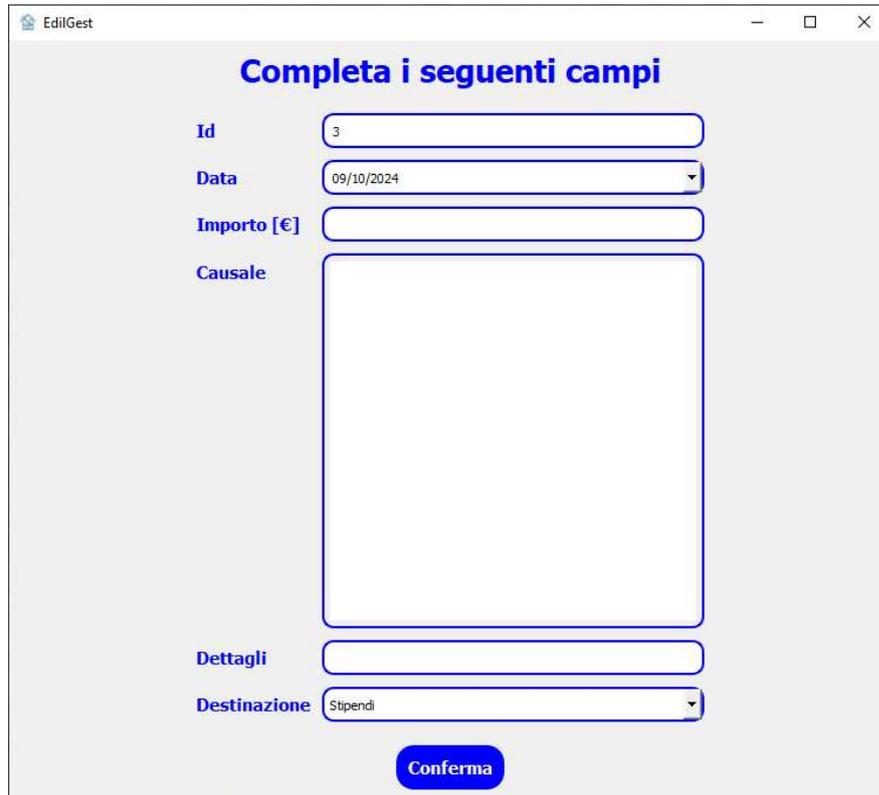
- *visualizzare il dettaglio dell'uscita selezionata*, attraverso il click sul pulsante "Modifica"; visualizza la schermata "Vista uscita", mostrata in Figura 5.68.

Nella schermata "Vista uscita", mostrata in Figura 5.68, l'utente può confermare le modifiche effettuate, attraverso il click sul pulsante "Conferma modifica", oppure eliminare l'uscita selezionata, cliccando sul pulsante "Elimina".

5.15 Sezione "Bilanci"

Cliccando sull'opzione "Bilanci" del menù a tendina associato al pulsante "Contabilità" della Home Page, in Figura 5.3, l'utente visualizza la schermata "Lista bilanci", mostrata in Figura 5.69; qui vede una lista contenente gli anni di tutti i bilanci presenti nel sistema.

Cliccando sul pulsante "Aggiungi" della schermata "Lista bilanci", mostrata in Figura 5.69, apparirà una schermata di pop-up, come mostrato in Figura 5.70, dove l'utente può selezionare di quale anno calcolare il bilancio; al click dell'opzione "Ok", il sistema calcola il bilancio dell'anno selezionato e lo inserisce nel sistema.



The screenshot shows a web application window titled 'EdilGest' with a close button. The main heading is 'Completa i seguenti campi'. The form contains the following fields:

- Id**: A text input field containing the number '3'.
- Data**: A date dropdown menu showing '09/10/2024'.
- Importo [€]**: An empty text input field.
- Causale**: A large, empty text area for entering the cause of the expense.
- Dettagli**: An empty text input field.
- Destinazione**: A dropdown menu with 'Stipendi' selected.

At the bottom center, there is a blue button labeled 'Conferma'.

Figura 5.67: Schermata per l'inserimento di una nuova uscita



The screenshot shows a web application window titled 'EdilGest' with a close button. The main heading is 'Info uscita'. The form contains the following fields:

- Id**: A text input field containing the number '1'.
- Data**: A date dropdown menu showing '09/10/2024'.
- Importo [€]**: A text input field containing '540.00'.
- Causale**: A text area containing the text: 'Pagamento di €540.0 in data 09/10/2024 per acquisto materie prime presso la ditta: Ceramiche Fratelli Ferrandino'.
- Dettagli**: An empty text input field.
- Destinazione**: A dropdown menu with 'Materiali e Forniture' selected.

At the bottom center, there are two blue buttons: 'Conferma modifica' and 'Elimina'.

Figura 5.68: Schermata relativa ai dettagli dell'uscita selezionata

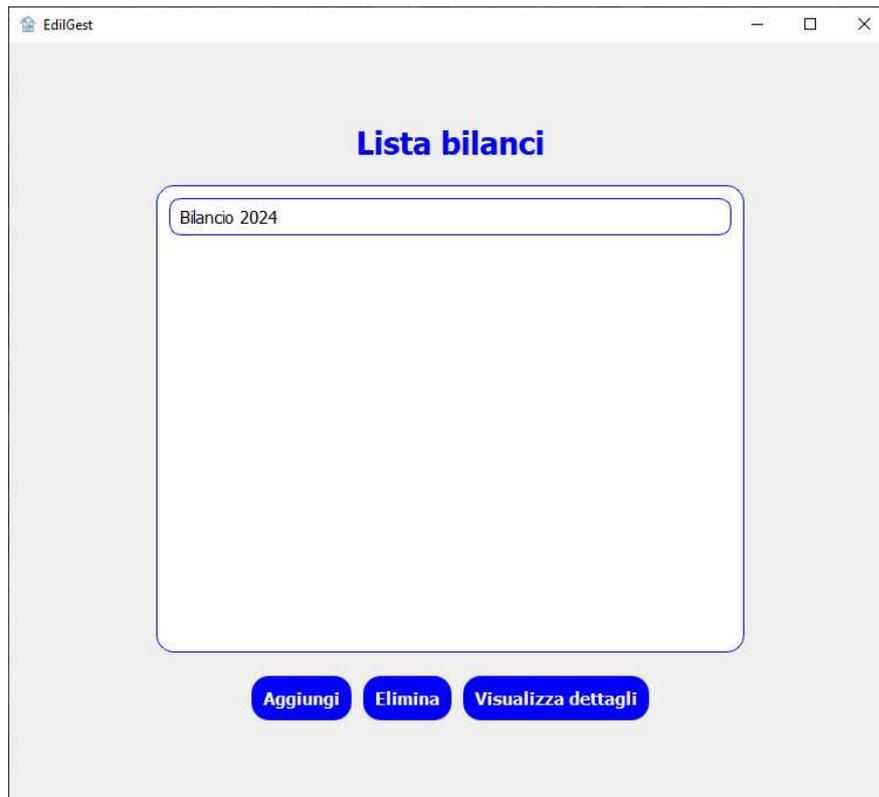


Figura 5.69: Schermata relativa alla lista dei bilanci

Al termine dell'operazione, il sistema visualizza nuovamente la schermata "Lista bilanci", mostrata in Figura 5.69, aggiornata.

Se nel sistema è già presente un bilancio relativo all'anno selezionato, l'operazione di inserimento fallisce e l'utente visualizza un messaggio di errore, come vediamo in Figura 5.71.

Dopo aver selezionato un bilancio nella schermata "Lista bilanci", mostrata in Figura 5.69, l'utente può effettuare due azioni:

- *visualizzare il dettaglio del bilancio selezionato*, attraverso il click sul pulsante "Visualizza dettagli"; visualizza una schermata pop-up contenente i dettagli del bilancio selezionato, come vediamo nella Figura 5.72;
- *eliminare il bilancio selezionato*, attraverso il click sul pulsante "Elimina"; effettua l'eliminazione del bilancio selezionato.

5.16 Sezione "Statistiche"

Cliccando sull'opzione "Vista statistiche" del menù a tendina associato al pulsante "Statistiche" della Home Page, in Figura 5.3, precisamente nel menù in alto, l'utente visualizza la schermata "Statistiche", mostrata in Figura 5.73, dove vede cinque pulsanti:

- *Dipendenti*: cliccando su questo pulsante, l'utente visualizza un grafico che ha in ascissa i nomi dei dipendenti e in ordinata i rispettivi stipendi, come vediamo in Figura 5.74;
- *Lavori*: cliccando su questo pulsante, l'utente può selezionare due opzioni:

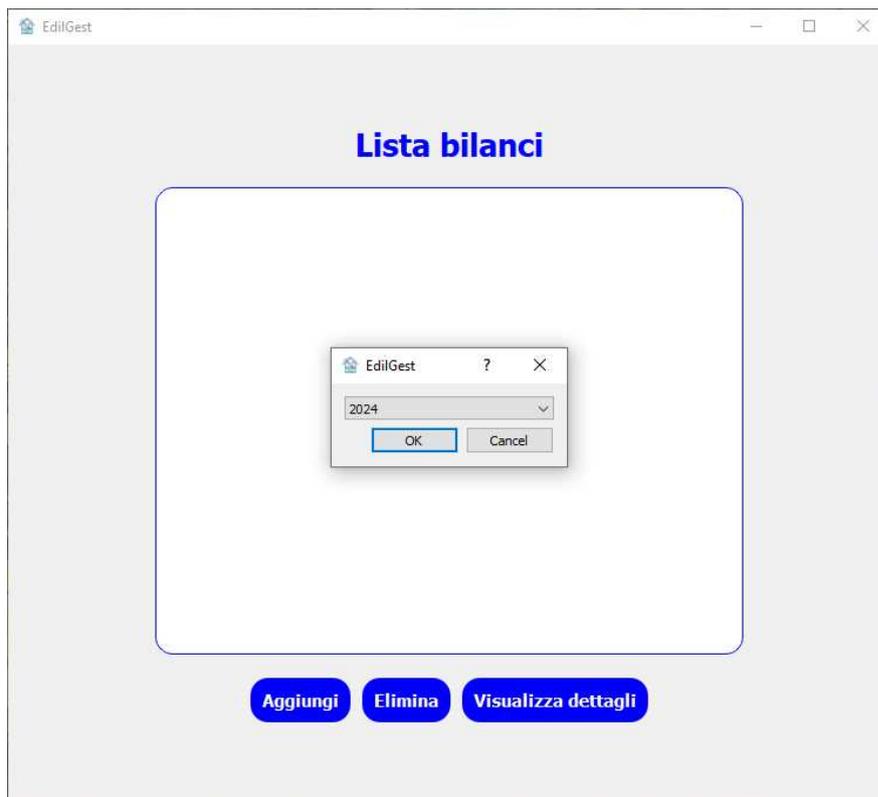


Figura 5.70: Schermata di pop-up per l'inserimento di un nuovo bilancio

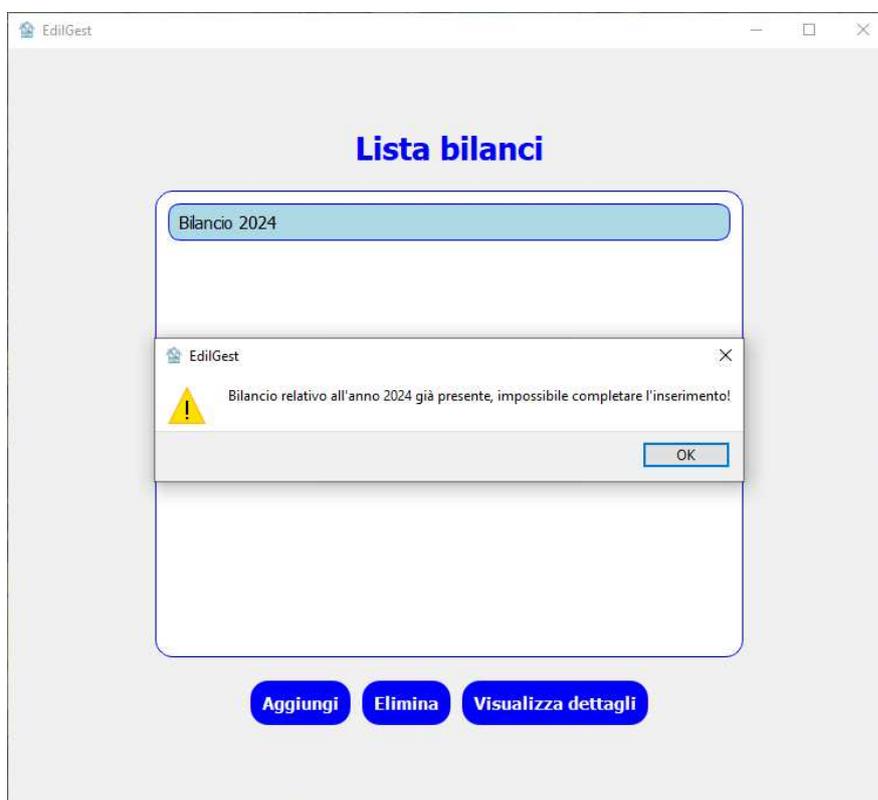


Figura 5.71: Avviso "Bilancio già presente per l'anno selezionato"

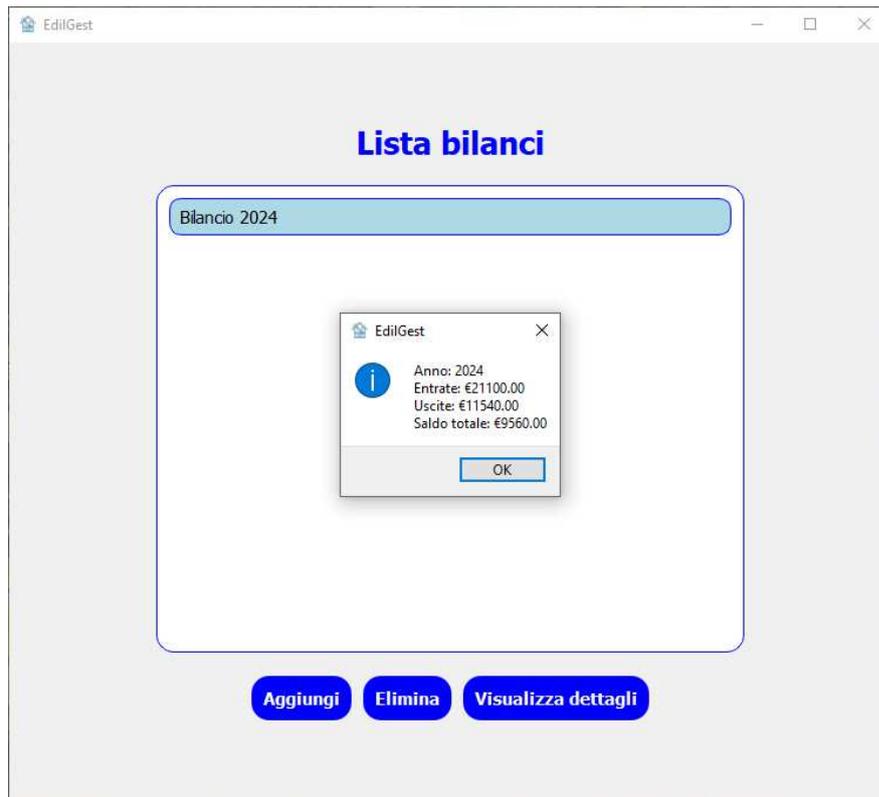


Figura 5.72: Schermata di pop-up relativa ai dettagli del bilancio selezionato

- *Importi*: selezionando questa opzione, l'utente visualizza un grafico che ha in ascissa le denominazioni dei lavori e in ordinata i rispettivi importi, come vediamo in Figura 5.75;
- *Pagamenti*: selezionando questa opzione, l'utente visualizza una schermata pop-up che mostra la percentuale di lavori pagati dal cliente (su tutti i lavori presenti nel sistema), come vediamo in Figura 5.76;
- *Costi/Ricavi*: cliccando su questo pulsante, dopo aver inserito la data iniziale e quella finale, l'utente visualizza la schermata "Statistiche costi e ricavi", mostrata in Figura 5.77, dove vede un riepilogo di tutte le entrate e le uscite effettuate nel periodo indicato;
- *Bilanci*: cliccando su questo pulsante, l'utente visualizza un grafico che ha in ascissa gli anni dei bilanci effettuati e in ordinata i rispettivi saldi complessivi, come vediamo in Figura 5.78;
- *Preventivi*: cliccando su questo pulsante, l'utente visualizza una schermata pop-up che mostra la percentuale di preventivi accettati dal cliente (su tutti i preventivi presenti nel sistema), come vediamo in Figura 5.79.

5.17 Video Tutorial

Cliccando sull'opzione "Visualizza tutorial" del menù a tendina associato al pulsante "Tutorial" della Home Page, in Figura 5.3, precisamente nel menù in alto, l'utente visualizza la schermata "Tutorial", mostrata in Figura 5.80; qui vede il video tutorial e una lista di collegamenti contenente tutte le categorie del software, ognuna associata al rispettivo istante

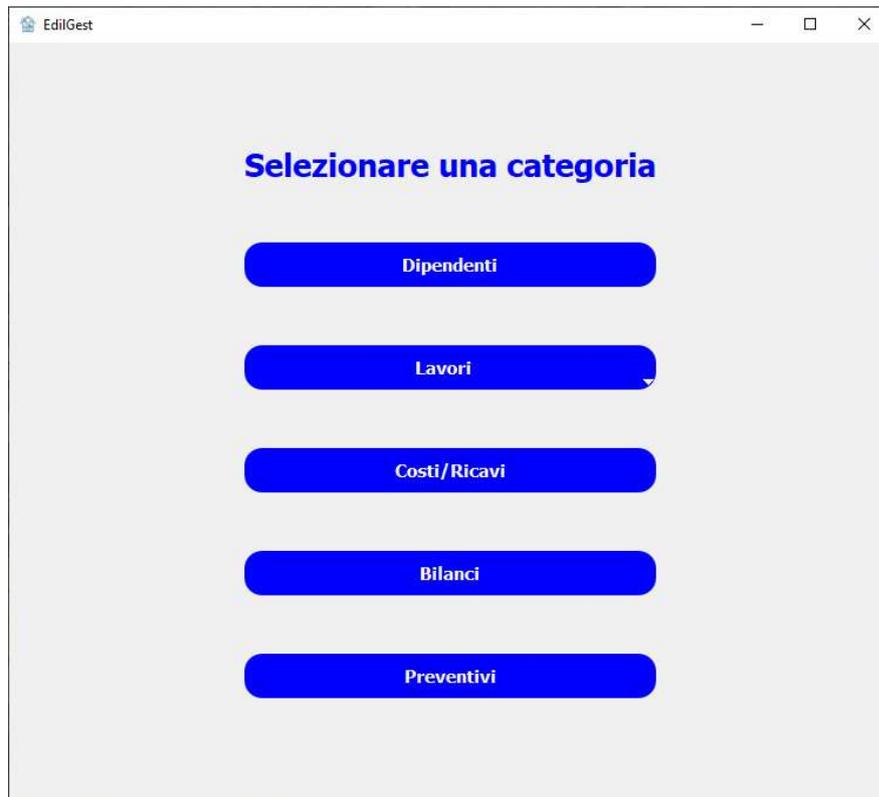


Figura 5.73: Schermata relativa alle statistiche

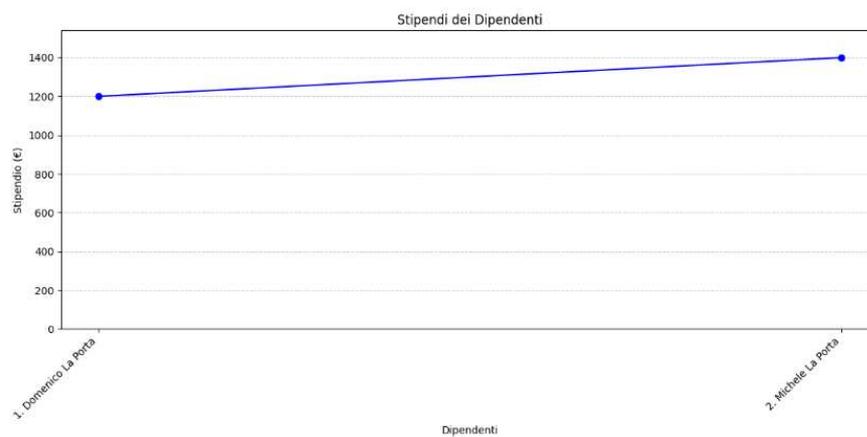


Figura 5.74: Grafico dipendenti-stipendi

del video in cui se ne parla; cliccando su uno di questi collegamenti, il video passa all'istante in cui si parla della categoria selezionata.

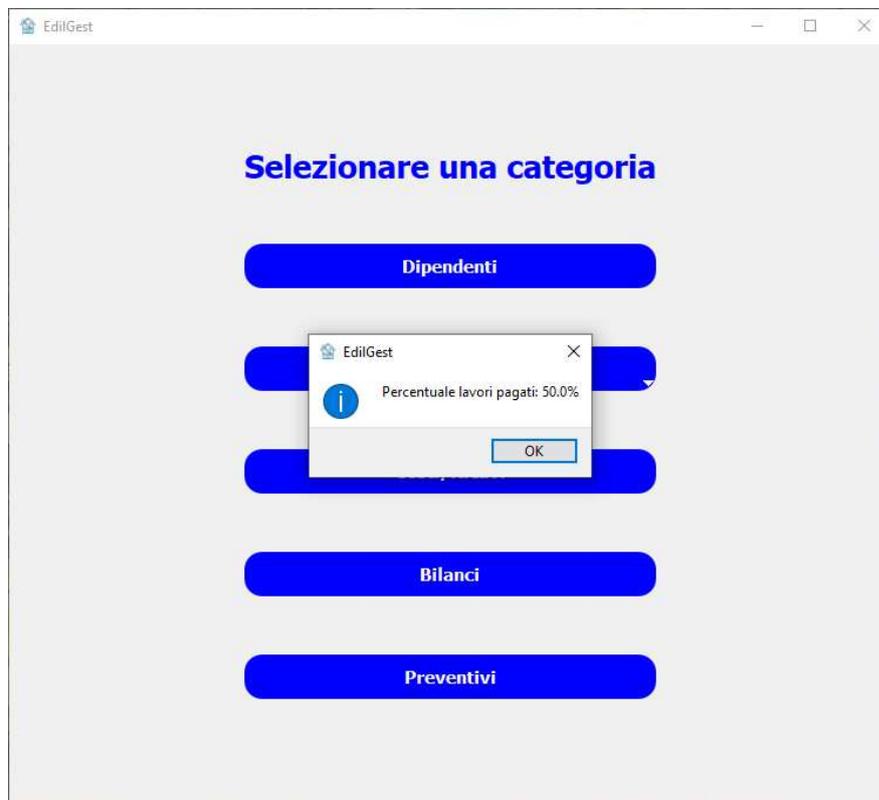


Figura 5.75: Schermata di pop-up che mostra la percentuale dei lavori pagati

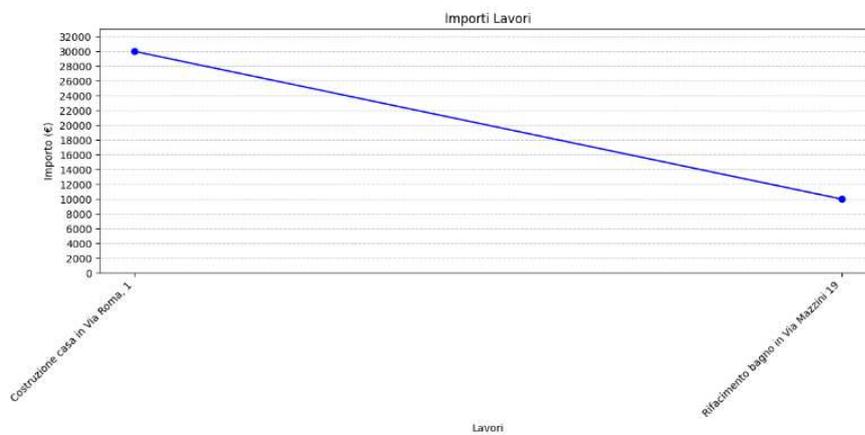


Figura 5.76: Grafico lavori-importi



Figura 5.77: Schermata relativa alla lista delle entrate e delle uscite nel periodo di tempo selezionato dall'utente

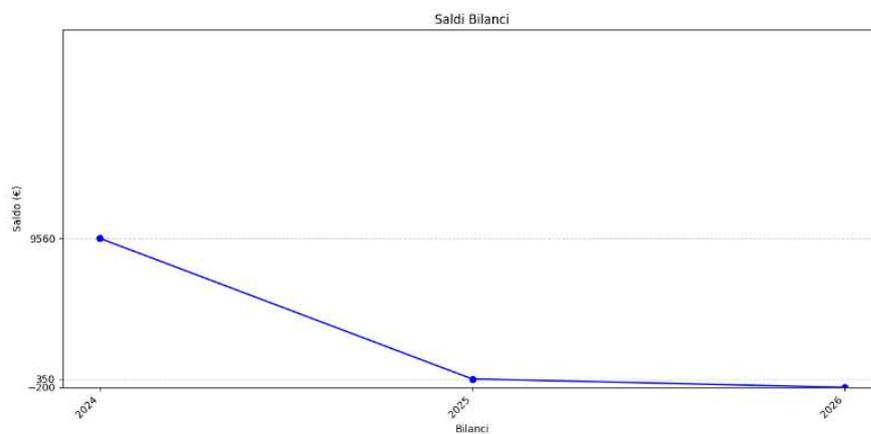


Figura 5.78: Grafico bilanci-saldi complessivi

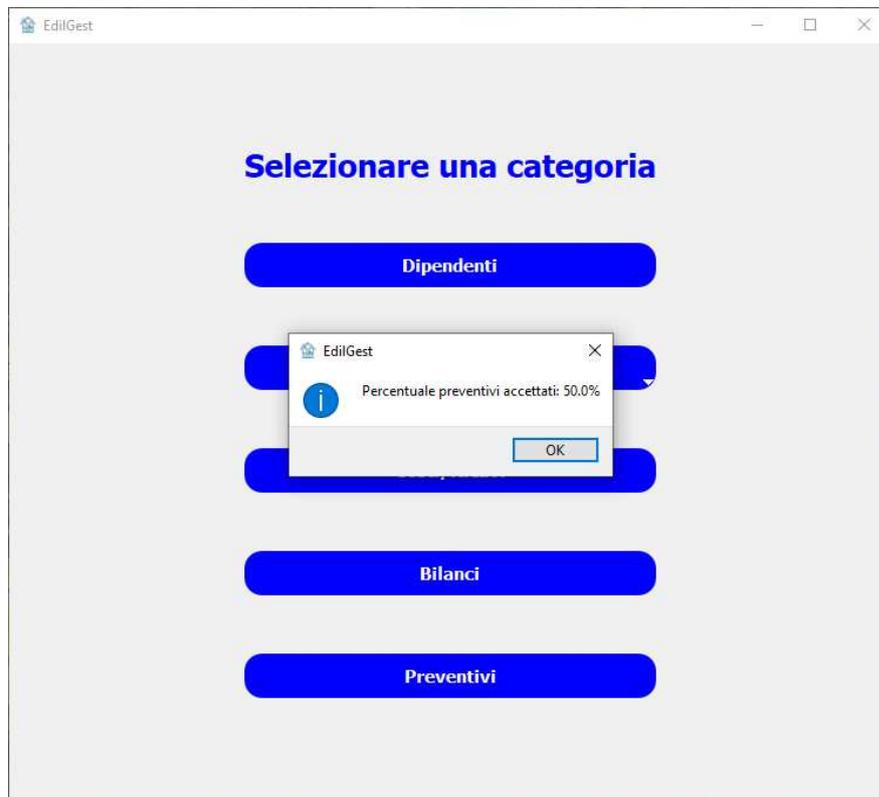


Figura 5.79: Schermata pop-up che mostra la percentuale dei preventivi accettati



Video tutorial per l'utilizzo di EdilGest

- 0.05 - autenticazione e cambio credenziali
- 0.30 - gestione dipendenti
- 1.30 - gestione lavori
- 2.25 - gestione preventivi
- 3.30 - gestione magazzino
- 4.30 - gestione corsi sicurezza sul lavoro
- 5.30 - gestione visite mediche
- 6.30 - gestione locali
- 7.30 - gestione entrate
- 8.30 - gestione uscite
- 9.30 - gestione bilanci
- 10.30 - gestione clienti
- 11.30 - gestione versoli
- 12.30 - gestione fornitori
- 13.30 - statistiche



Figura 5.80: Schermata Web che mostra il video tutorial di come utilizzare il software

Nel corso di questa tesi abbiamo affrontato il processo di sviluppo di un software per la gestione dell'“Impresa Edile La Porta Tommaso” in tutte e quattro le sue fasi fondamentali.

Inizialmente, abbiamo esplorato il mondo dell'impresa in generale, mettendo, poi, l'attenzione sul settore edile e su come le aziende italiane investano nel proprio settore informatico. Successivamente, ci siamo interfacciati con l'impresa committente attraverso un'intervista svolta su più livelli, al fine di avere una visione olistica sui requisiti del sistema, che ci ha portato, dopo un'attenta analisi, a definire i possibili scenari di utilizzo del software in modo chiaro e inequivocabile.

Una volta terminata questa fase, ci siamo dedicati alla progettazione del database, partendo dalla scelta del modello da utilizzare, ovvero quello relazionale, per poi iniziare il lavoro vero e proprio che ha compreso la specifica delle entità fondamentali del sistema, la definizione del diagramma E/R e la traduzione verso il modello relazionale, senza tralasciare l'attenta analisi svolta sui vincoli da rispettare.

A questo punto, prima di iniziare a scrivere codice, è stata fondamentale la fase di progettazione, svolta prima dal punto di vista strutturale, attraverso il diagramma dell'architettura e quello delle classi, e poi da quello comportamentale, utilizzando i diagrammi di attività e quelli di sequenza. Questa fase, come già affermato, è stata tanto importante in questo contesto – quello dell'ingegneria del software – quanto lo sarebbe stata in altri settori, come quello edile; pensare di passare subito alla fase di implementazione senza aver prima progettato tutto, sarebbe stato controproducente, oltre che sbagliato.

Soltanto dopo aver definito una buona architettura e analizzato il sistema dal punto di vista comportamentale, siamo passati alla fase di implementazione, evidenziando le tecnologie utilizzate e mostrando alcuni esempi presi direttamente dal codice sorgente del programma, seguita da quella di testing, in cui, attraverso la libreria PyUnit, abbiamo verificato il corretto funzionamento di alcune funzionalità implementate.

Infine, abbiamo dotato il prodotto finale di un manuale utente, un documento essenziale e da non dimenticare nella fase di rilascio del software, utile al cliente per capire come muoversi all'interno di “EdilGest”.

L'aspetto interessante dello sviluppo di un software è che si tratta di un lavoro sempre in evoluzione e che, difficilmente, termina completamente; il modello proposto, ovvero quello definito dalle fasi di analisi e specifica dei requisiti, progettazione, implementazione e testing, non si conclude mai del tutto, in quanto si è sempre pronti a ripartire daccapo, ad esempio, nel caso in cui il cliente voglia aggiungere nuove funzionalità oppure vengano alla luce potenziali bug da sistemare. Altro aspetto importante è la manutenzione del sistema; per gli stessi motivi per cui è necessario effettuare periodicamente la manutenzione in altri

settori dell'ingegneria – si pensi sempre al settore edile – anche nel nostro ambito bisogna, inevitabilmente, prendere in considerazione questo lavoro che, ormai, è diventato quasi una quinta fase che si aggiunge alle quattro del modello visto in precedenza.

Il rischio a cui si va incontro nel momento in cui si tralasciano gli sviluppi futuri nell'ambito della realizzazione di un software è che questo diventi legacy, ovvero obsoleto, retrogrado, e che venga utilizzato soltanto perché non si intende o non si può rimpiazzarlo con un altro prodotto più all'avanguardia. Il nostro obiettivo, da portare avanti anche con l'aiuto degli stakeholder, è quello di proporre periodicamente delle novità sotto ogni punto di vista, in modo che il sistema possa continuare ad essere utilizzato, non per mancanza di alternative, ma perché risulta essere moderno e all'avanguardia.

Possibili nuovi scenari per lo sviluppo futuro sono, quindi:

- *Introduzione di nuovi livelli di autorizzazione*: si può pensare di introdurre diversi livelli di autorizzazione, in modo che anche, ad esempio, i clienti oppure i fornitori possano essere degli attori e interagire con il software e con l'impresa edile.
- *Introduzione di nuove funzionalità*: si può pensare, anche come conseguenza del caso precedente, di espandere le funzionalità del sistema, ad esempio, aggiungendo uno scenario in cui l'utente dell'impresa interagisce, attraverso un servizio di messaggistica interno al sistema, con i clienti e/o i fornitori.
- *Gestione del rischio*: dobbiamo essere pronti, anche come conseguenza del primo scenario, ad evolvere il nostro software in modo che sia sempre più sicuro e che sia in grado di gestire eventuali minacce esterne, quali, ad esempio:
 - *Spoofing*: prendere l'identità di qualcun altro; violazione dell'autenticazione.
 - *Tampering*: manomettere le informazioni; violazione dell'integrità.
 - *Repudiation*: ripudiare qualcosa che si è fatto (ad esempio, un utente dice di aver accettato un ordine, ma, in un secondo momento, dice che non l'ha mai fatto); violazione della responsabilità.
 - *Information Disclosure*: sono state scoperte informazioni che non dovevano essere scoperte; violazione della confidenzialità.
 - *Denial of Service*: violazione della disponibilità.
 - *Elevation of Privilege*: riuscire ad ottenere privilegi più alti di quelli che uno dovrebbe avere; violazione dell'autorizzazione.
- *Aggiornamento delle interfacce grafiche*: un modo per mantenere all'avanguardia il software è di aggiornare continuamente le interfacce grafiche, in modo che queste restino al passo con i tempi, sempre, però, ricordandosi di rispettare il requisito non funzionale della buona User Experience.
- *Creazione di un'applicazione mobile*: si può pensare di introdurre un'applicazione mobile per consentire all'utente di gestire i principali processi aziendali direttamente dal proprio smartphone.

Seguendo queste proposte, il nostro sistema continuerà ad essere al passo con i tempi, sicuro e sarà in grado, sempre di più, di aiutare l'"Impresa Edile La Porta Tommaso" nella gestione dei propri processi in maniera efficiente.

- ATZENI, P., CERI, S., FRATERNALI, P., PARABOSCHI, S. e TORLONE, R. (2018), *Basi di dati*, McGraw-Hill.
- BOSCAINI, M. (2019), *Imparare a programmare con Python*, Apogeo.
- CERI, S., MANDRIOLI, D. e SBATELLA, L. (2004), *Informatica: arte e mestiere*, McGraw-Hill.
- CERVELLI, R. (2013), *Database e linguaggio SQL*, Lulu.com.
- CURTIN, D. P., FOLEY, K., SEN, K. e MORIN, C. (2016), *Informatica di base*, McGraw-Hill.
- DEITEL, P. e DEITEL, H. (2021), *Introduzione a Python*, Pearson.
- FOWLER, M. (2018), *UML distilled*, Pearson.
- LAMBERT, K. A. (2018), *Programmazione in Python*, Maggioli Editore.
- LARMAN, C. (2020), *Applicare UML e i pattern*, Pearson.
- MCGRAW, G. (2006), *Software Security: Building Security In*, Addison-Wesley Professional.
- SEBASTIAN, S. (2023), *Ingegneria del software: Processi di sviluppo del software*, Edizioni Sapienza.
- SOMMERVILLE, I. (2017), *Ingegneria del software*, Pearson.
- TOSORATTI, P. (1998), *Introduzione all'informatica*, Casa Editrice Ambrosiana.

- **About SQLite** – <https://www.sqlite.org/about.html>
- **Esercitazione Pyplot** – <https://matplotlib.org/stable/tutorials/pyplot.html>
- **GitHub Docs** – <https://docs.github.com>
- **Guida Python - PyQt** – <https://www.html.it/pag/71918/pyqt/>
- **iCRIBS** – www.contenuti.icribis.com/osservatorio/2023/impres-edili
- **OrganizationsUsingPython** – <https://wiki.python.org/moin/OrganizationsUsingPython>
- **Partitaiva** – www.partitaiva.it/impres-italia/
- **pycodicefiscale** – <https://github.com/ema/pycodicefiscale>
- **PyCharm Documentation** – <https://www.jetbrains.com/help/pycharm>
- **PyQt5: appunti** – <https://bancaldo.wordpress.com/2019/03/13/pyqt5-appunti/>
- **re — Regular expression operations** – <https://docs.python.org/3/library/re.html>
- **sqlite3** – <https://docs.python.org/3/library/sqlite3.html>
- **webbrowser – Un semplice gestore per browser web** – <https://docs.python.it/html/lib/module-webbrowser.html>

Siamo giunti alla conclusione di questo lavoro di tesi che chiude questo percorso di studi iniziato nel 2019 da un ragazzo che non sapeva nemmeno cosa fosse l'informatica, che, dopo il primo anno, dopo la pandemia, dopo aver avuto difficoltà a superare anche i primi esami, stava iniziando a pensare che forse non era la strada giusta per lui. Tuttavia, anche con l'aiuto di chi gli è stato vicino, oggi quel ragazzo si ritrova a festeggiare il raggiungimento di un traguardo che solo due anni fa sembrava irraggiungibile.

Il mio primo ringraziamento va a Dio; durante questo percorso di studi ho avuto modo di avvicinarmi alla fede e di conoscere il Signore personalmente. Questo mi ha portato, tra le tantissime cose positive, ad avere una spinta in più anche nell'ambito universitario; in Lui ho trovato la forza e il coraggio di andare avanti e di superare anche i momenti più difficili; ho sperimentato la Sua vicinanza e la Sua guida in ogni situazione e ho visto la Sua mano potente condurmi al meglio nel mio cammino. Nella Bibbia è scritto in Filippesi 4:6 "Non angustiatevi di nulla, ma in ogni cosa fate conoscere le vostre richieste a Dio in preghiere e suppliche, accompagnate da ringraziamenti"; durante questo percorso ho pregato tanto Dio ed Egli mi ha risposto e, in virtù di questo, non voglio dimenticare anche la seconda parte del versetto appena citato, ringraziandoLo, appunto, per ogni cosa che Egli ha fatto per me.

Voglio ringraziare anche il Prof. Ursino; sin dal primo giorno di lezione mi ha fatto appassionare tanto all'Ingegneria del Software; nonostante le difficoltà a superare l'esame, ha saputo affrontare la situazione particolare che si era creata e, venendomi incontro, mi ha aiutato nella consegna del progetto finale, nonché il mio primo software sviluppato. Nel corso del lavoro di tesi, poi, è stato sin da subito molto disponibile e gentile nei miei confronti, nonostante mi trovassi in una situazione nuova e non sapevo bene come muovermi; è stato sempre puntuale sia nelle risposte alle mie domande sia nelle correzioni, rendendo questa esperienza piacevole, leggera, senza affanni e preoccupazioni, ma allo stesso tempo molto professionale. Ho apprezzato molto il suo modo di rapportarsi con me; solo il fatto di essere addirittura chiamato per nome (e che bel nome!) mi ha fatto sentire molto coinvolto e vicino; quando mi parlava, sembrava quasi si rivolgesse non a uno studente, ma ad un figlio e questa è una cosa che ho apprezzato tanto.

Voglio ringraziare i miei genitori, mamma e papà, che mi hanno sempre sostenuto, incoraggiato, mi sono sempre stati vicini e sono stati pronti tanto a festeggiare per gli esami superati, quanto a tirarmi su nei momenti di difficoltà. Non mi hanno fatto mai mancare niente, sia dal punto di vista materiale che, soprattutto, morale, concedendomi di vivere un'esperienza fantastica sotto ogni punto di vista. Vivere lontano dalla propria famiglia non è semplice, sia per noi figli, ma anche per i genitori, e questo l'ho capito nel corso di questi anni. Nonostante la lontananza, però, mamma e papà hanno sempre avuto a cuore di sentirmi,

giorno dopo giorno, attraverso le videochiamate quotidiane e, quando c'è stata l'opportunità, anche venendomi a trovare, addirittura a sorpresa una volta, qui ad Ancona. Quando si va a vivere da soli, si capiscono tante cose e si inizia a dare valore a molti aspetti su cui prima non ci si soffermava abbastanza; è stato proprio questo quello che è successo: ho iniziato a dare valore ai momenti trascorsi in famiglia, che prima erano scontati, ma ora sono diventati molto meno frequenti; ho apprezzato ancora di più i sacrifici fatti da mamma e papà nel portare avanti una famiglia, sotto ogni punto di vista; in questa mia esperienza da fuori sede ho imparato talmente tanto ad empatizzare con i sacrifici fatti dai miei genitori che ho iniziato ad utilizzare frasi del tipo: "Ho appena pulito e già è sporco!" oppure "La luce accesa consuma corrente!". Posso dire che ho iniziato a dare tanto valore a cose che prima, probabilmente, davvo per scontate, ma che, in realtà, non lo erano per niente. Ed è per questo che ci tengo ancora a ringraziare mamma e papà per tutto quello che hanno fatto per me, non soltanto in questi anni, ma in tutta la mia vita.

Voglio ringraziare i miei fratelli Michele e Marco; ognuno dei due, in una maniera diversa, visti gli anni di differenza, mi ha aiutato in questi anni. Con Michele ho avuto il piacere di vivere nella stessa casa per tre anni; trascorrere le giornate lontano dal proprio paese e dai propri genitori non è semplice, ma la presenza costante di un fratello, così come l'ho avuta io, ha reso tutto molto più leggero e semplice, come se, alla fine, casa nostra non l'avessimo mai lasciata. Marco è il fratellino piccolo a cui vorresti insegnare tutto quello che impari, sperando di trasmettergli gli stessi interessi; per ora non ci sono riuscito: infatti ha sempre storto il naso quando volevo fargli testare i programmi che sviluppavo; però vedere la sua spensieratezza, quella di un bambino, vedere anche la sua gioia, sincera, quando tornavo a trovarlo o quando lui veniva a visitarmi qui ad Ancona, ma anche il suo dispiacere quando ripartivo, mi ha aiutato veramente tanto.

Voglio ringraziare tutto il resto della mia famiglia, i miei nonni Mimì, Rita, Michele e Maria, i miei zii Matteo, Lucia, Patrizia, Lucia, Vincenzo, le mie cugine Valentina, Rita, Ciretta; tutti, in un modo diverso, sono stati importanti in questo percorso.

Voglio ringraziare Maria Chiara, una delle persone per me più importanti e che più ha avuto un'impronta in questi anni, dimostrandomi, in tante occasioni, la sua vicinanza e il suo affetto. Voglio ringraziare Miriana, un'altra presenza costante in questo mio percorso, che mai ha fatto mancare in me il suo sostegno in ogni circostanza, sempre pronta in qualsiasi momento ad essere presente. Voglio ringraziare Andrea, il fratello maggiore che non ho mai avuto, un esempio per me, una persona che mi ha dato tanto, forse anche più di quello che crede. Voglio ringraziare Angela, colei che, dopo averla conosciuta nel mio paese d'origine quando ero bambino, ho incontrato dopo più di dieci anni qui ad Ancona e che, da quel momento in poi, ha sempre ricoperto un ruolo molto importante.

Voglio ringraziare Giuseppe, la prima persona che mi ha rivolto la parola all'università, il mio collega nello sviluppo di GreenMarket, ma, soprattutto, un amico con cui ho condiviso quotidianamente tutto il mio percorso universitario. Voglio ringraziare Emanuele, la prima macchina da guerra, dal punto di vista della programmazione, conosciuta; ricordo con piacere CovidForecast, progetto di Programmazione ad Oggetti, ed è vedendolo lavorare in quel contesto che ho iniziato ad apprezzare lo sviluppo di programmi, la creazione di interfacce grafiche la programmazione in generale. Voglio ringraziare Paolo, un altro programmatore fuori dal comune che, a vederlo lavorare su FlatMate, ho avuto solo da prendere esempio. Voglio ringraziare Emilio, con cui non ho mai lavorato insieme, ma ho avuto modo di vedere quanto fosse in gamba e professionale sotto ogni punto di vista. A prescindere, però, dalle competenze, loro sono stati gli amici con cui ho condiviso ogni momento nell'ambito universitario, sin dalle prime esercitazioni di Algebra Lineare e Geometria del lontano settembre 2019 e li ringrazio per essere sempre stati presenti in ogni contesto.

Voglio ringraziare Giovanni, il mio amico, ormai fratello, con cui ci conosciamo da 21

anni. Abbiamo condiviso insieme gli anni dall'asilo fino alle scuole superiori; poi, purtroppo, le strade si sono separate; nonostante i 300km di distanza, però, siamo sempre stati vicini, anche durante il percorso universitario e, ogni volta che ci rincontravamo, era come se non ci fossimo mai lasciati.

Voglio ringraziare tutto il mio gruppo di amici del mio paese di origine; anche se siamo andati a finire in zone diverse dell'Italia, siamo sempre rimasti in contatto; inoltre, non dimentico i caffè presi e le chiacchierate fatte quando ci rincontravamo, momenti preziosi che ho sempre apprezzato tanto.

Voglio ringraziare il gruppo giovani della chiesa di San Nicandro Garganico; ognuno in una maniera diversa, sono stati, a partire dal 2021, anno in cui mi sono legato maggiormente a loro, una presenza importantissima; sapere che in ogni occasione avrei trovato il loro appoggio e il loro sostegno mi ha dato una grande mano, e per questo voglio ringraziarli.

E per ultimo, non posso dimenticare di ringraziare l'Impresa Edile La Porta Tommaso che, nelle figure di mio padre e di mio nonno, mi ha dato anche l'opportunità, come si vede nelle Figure 5.81, 5.82, 5.83, 5.84 e 5.85, di sporcarmi le mani durante la fase di specifica dei requisiti; ciò mi ha aiutato nella realizzazione del software, in quanto mi ha fatto comprendere ancora meglio i processi aziendali :D.

Concludo con un grazie complessivo a chiunque abbia avuto un ruolo in questo mio percorso. Grazie a tutti voi.



Figura 5.81: Foto in cantiere da solo



Figura 5.82: Foto in cantiere con nonno Mimì, papà e mio fratello Michele



Figura 5.83: Foto in cantiere con papà e i miei due fratelli Michele e Marco



Figura 5.84: Foto sul cestello con mio fratello Michele



Figura 5.85: Foto in cantiere con mio fratello Michele