

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 realizzazione di un sistema per la gestione delle
attività di un'infermiera freelance**

**Design and implementation of a system for the management of the
activities of a freelance nurse**

Relatore

Prof. Domenico Ursino

Candidato

Lorenzo Longarini

ANNO ACCADEMICO 2020-2021

*Mi domando - disse - se le stelle sono illuminate
perchè ognuno un giorno possa trovare la sua*

Antoine De Saint-Exupery, "Il piccolo principe"

Sommario

Il software ha assunto sempre di più una grande rilevanza nel corso degli anni, in quanto aziende e privati necessitano di ottimizzare e velocizzare determinati processi lavorativi. In questa tesi vengono descritte la progettazione e l'implementazione di un software gestionale per un'infermiera libera professionista. In particolare è stata realizzata una fase di raccolta dei requisiti, seguita da un'analisi degli stessi. Attraverso questi ultimi è stato possibile articolare il progetto nelle varie classi con le rispettive funzionalità, mentre il comportamento del programma è stato analizzato attraverso i diagrammi dei casi d'uso, di sequenza e di attività. Infine, attraverso l'utilizzo del framework Django, abbiamo implementato, e successivamente testato, quanto è stato visto.

Keyword: Ingegneria del software, Diagrammi UML, Specifica e Analisi dei Requisiti, Django, Python

Introduzione	1
1 Definizione del contesto di riferimento	3
1.1 Intervista	3
1.2 Descrizione del sistema in linguaggio naturale	5
1.3 Glossario dei termini	6
2 Specifica e analisi dei requisiti	8
2.1 Requisiti funzionali	8
2.1.1 Gestione Agenda	9
2.1.2 Gestione Autenticazione	10
2.1.3 Gestione FAQ	10
2.1.4 Gestione Fatture	10
2.1.5 Gestione Prenotazioni	11
2.1.6 Gestione Prodotti	11
2.1.7 Gestione Tariffe	11
2.2 Requisiti non funzionali	11
3 Diagrammi delle classi	13
3.1 Mappa dell'architettura	13
3.2 Gestionale_Infermiera	14
3.2.1 ControllerHome	14
3.3 Agenda	14
3.3.1 Appuntamento	15
3.3.2 Meta	16
3.3.3 AgendaAdmin	16
3.3.4 EventCalendar	16
3.3.5 AgendaConfig	17
3.4 Tariffa	17
3.4.1 Tariffa	18
3.4.2 Meta	18
3.4.3 TariffaAdmin	19
3.4.4 TariffaConfig	19
3.5 Fattura	20
3.5.1 Fattura	20

3.5.2	Meta	21
3.5.3	TariffaFatt	22
3.5.4	FatturaAdmin	22
3.5.5	FatturaInline	22
3.5.6	ControllerFattura	23
3.5.7	FatturaConfig	23
3.6	Prodotto	24
3.6.1	Prodotto	24
3.6.2	Meta	25
3.6.3	ProdottoAdmin	25
3.6.4	ProdottoConfig	26
3.7	Prenotazione	26
3.7.1	Prenotazione	27
3.7.2	ControllerPrenotazione	27
3.7.3	PrenotazioneConfig	27
3.8	FAQ	28
3.8.1	FAQ	29
3.8.2	FAQAdmin	29
3.8.3	ControllerFAQ	29
3.8.4	FAQConfig	30
4	Diagrammi del comportamento	31
4.1	Diagramma dei casi d'uso	31
4.1.1	Gestione dell'agenda	32
4.1.2	Gestione delle tariffe	35
4.1.3	Gestione delle fatture	38
4.1.4	Gestione dei prodotti	41
4.1.5	Gestione delle FAQ	44
4.1.6	Gestione delle prenotazioni	47
4.1.7	Gestione dell'autenticazione	48
4.2	Matrice di mapping dei requisiti	50
4.3	Diagrammi di sequenza	51
4.3.1	Gestione dell'agenda	52
4.3.2	Gestione delle tariffe	53
4.3.3	Gestione delle fatture	54
4.3.4	Gestione dei prodotti	55
4.3.5	Gestione delle FAQ	57
4.3.6	Gestione delle prenotazioni	58
4.3.7	Gestione dell'autenticazione	59
4.4	Diagramma di attività	60
4.4.1	Gestione dell'agenda	60
4.4.2	Gestione delle tariffe	61
4.4.3	Gestione delle fatture	62
4.4.4	Gestione dei prodotti	63
4.4.5	Gestione delle FAQ	64
4.4.6	Gestione delle prenotazioni	66
4.4.7	Gestione dell'autenticazione	66

5	Definizione dei mockup	67
5.1	Area Amministratore	67
5.1.1	Login	67
5.1.2	Home	68
5.1.3	Agenda	68
5.1.4	FAQ	70
5.1.5	Fattura	72
5.1.6	Prodotto	74
5.1.7	Tariffa	76
5.1.8	Modifica Password	77
5.1.9	Logout	78
5.2	Area Cliente	78
5.2.1	Home page	79
5.2.2	Contattaci	80
5.2.3	Chi siamo	82
5.2.4	FAQ	82
6	Test del sistema	84
6.1	Framework e linguaggio utilizzati	84
6.2	Unit testing	84
6.2.1	Test relativo all'agenda	85
6.2.2	Test relativo alle FAQ	86
6.2.3	Test relativo alle fatture	86
6.2.4	Test relativo alle prenotazioni	88
6.2.5	Test relativo al prodotto	88
6.2.6	Test relativo alle tariffe	89
	Conclusioni e sviluppi futuri	91
	Bibliografia	93
	Ringraziamenti	95

Elenco delle figure

2.1	Analisi dei requisiti funzionali e non funzionali, schema generale	8
2.2	Analisi dei requisiti funzionali relativi al nostro sistema	9
2.3	Analisi dei requisiti non funzionali	12
3.1	Mappa dell'architettura con le relative interazioni tra classi	13
3.2	Package Gestionale_Infermiera e relative classi	14
3.3	Package Agenda e relative classi	15
3.4	Package Tariffa e relative classi	18
3.5	Package Fattura e relative classi	20
3.6	Package Prodotto e relative classi	24
3.7	Package Prenotazione e relative classi	26
3.8	Package FAQ e relative classi	28
4.1	Elenco dei casi d'uso relativi al sistema	31
4.2	Casi d'uso relativi alla gestione dell'agenda	32
4.3	Casi d'uso relativi alla gestione del tariffario	35
4.4	Casi d'uso relativi alla gestione delle fatture	38
4.5	Casi d'uso relativi alla gestione dei prodotti	42
4.6	Casi d'uso relativi alla gestione delle FAQ	45
4.7	Caso d'uso relativo alla gestione delle prenotazioni	48
4.8	Casi d'uso relativi alla gestione dell'autenticazione	49
4.9	Matrice di mapping dei requisiti	51
4.10	Diagramma di sequenza relativo alla gestione dell'agenda	53
4.11	Diagramma di sequenza relativo alla gestione delle tariffe	54
4.12	Diagramma di sequenza relativo alla gestione delle fatture	55
4.13	Diagramma di sequenza relativo alla gestione dei prodotti	56
4.14	Diagramma di sequenza relativo alla gestione delle FAQ per l'amministratore	58
4.15	Diagramma di sequenza relativo alla gestione delle FAQ per il cliente	58
4.16	Diagramma di sequenza relativo alla gestione delle prenotazioni	59
4.17	Diagramma di sequenza relativo alla gestione dell'autenticazione	60
4.18	Diagramma di attività relativo alla gestione dell'agenda	61
4.19	Diagramma di attività relativo alla gestione delle tariffe	62
4.20	Diagramma di attività relativo alla gestione delle fatture	63
4.21	Diagramma di attività relativo alla gestione dei prodotti	64
4.22	Diagramma di attività relativo alla gestione delle FAQ per un amministratore	65

4.23	Diagramma di attività relativo alla gestione delle FAQ per un cliente	65
4.24	Diagramma di attività relativo alla gestione delle prenotazioni	66
4.25	Diagramma di attività relativo alla gestione dell'autenticazione	66
5.1	Mockup della schermata di login	67
5.2	Mockup della schermata home page	68
5.3	Mockup della schermata relativa all'agenda: inserimento/modifica di un appuntamento	68
5.4	Mockup della schermata relativa all'agenda: visualizzazione di un appuntamento	69
5.5	Mockup della schermata relativa all'agenda: eliminazione di un appuntamento	69
5.6	Mockup della schermata relativa all'agenda: eliminazione di un appuntamento	69
5.7	Mockup della schermata relativa all'agenda: ricerca di un appuntamento . . .	70
5.8	Mockup della schermata relativa alle FAQ: inserimento/modifica di una FAQ	70
5.9	Mockup della schermata relativa alle FAQ: eliminazione di una FAQ	71
5.10	Mockup della schermata relativa alle FAQ: visualizzazione di tutte le FAQ . .	71
5.11	Mockup della schermata relativa alle FAQ: ricerca di una FAQ	71
5.12	Mockup della schermata relativa alle fatture: inserimento/modifica di una fattura	72
5.13	Mockup della schermata relativa alle fatture: inserimento/modifica di una fattura	72
5.14	Mockup della schermata relativa alle fatture: eliminazione di una fattura . . .	73
5.15	Mockup della schermata relativa alle fatture: ricerca di una fattura	73
5.16	Mockup della schermata relativa alle fatture: generazione del file PDF	73
5.17	Mockup della schermata relativa alle fatture: file PDF generato dalla fattura .	74
5.18	Mockup della schermata relativa ai prodotti: inserimento/modifica di un prodotto	74
5.19	Mockup della schermata relativa ai prodotti: visualizzazione di un prodotto .	75
5.20	Mockup della schermata relativa ai prodotti: ricerca di un prodotto	75
5.21	Mockup della schermata relativa ai prodotti: eliminazione di un prodotto . .	75
5.22	Mockup della schermata relativa alle tariffe: eliminazione di una tariffa	76
5.23	Mockup della schermata relativa alle tariffe: eliminazione di una tariffa	76
5.24	Mockup della schermata relativa alle tariffe: eliminazione di una tariffa	77
5.25	Mockup della schermata relativa alle tariffe: ricerca di una tariffa	77
5.26	Mockup della schermata relativa alla modifica della password	78
5.27	Mockup della schermata relativa al logout	78
5.28	Mockup relativo all'header	79
5.29	Mockup relativo al footer	79
5.30	Mockup relativo alla home page: slider iniziale	79
5.31	Mockup relativo alla home page: informazioni personali	80
5.32	Mockup relativo alla home page: informazioni riguardanti il lavoro	80
5.33	Mockup della schermata relativa a <i>Contattaci</i> : compilazione form	81
5.34	Mockup della schermata relativa a <i>Contattaci</i> : compilazione form	81
5.35	Mockup della schermata relativa a <i>Contattaci</i> : successo nell'invio della form .	81
5.36	Mockup della schermata relativa a <i>Chi siamo</i>	82
5.37	Mockup della schermata relativa a <i>Chi siamo</i>	82
5.38	Mockup della schermata relativa alle FAQ	83
6.1	Test di <i>Agenda</i> : creazione della classe e setup dei dati	85
6.2	Test di <i>Agenda</i> : etichetta relativa al nome	85
6.3	Test di <i>Agenda</i> : funzione <code>check_overlap</code>	85
6.4	Test di <i>Agenda</i> : funzione <code>clean</code>	85

6.5	Test di <i>Agenda</i> : funzione <code>get_absolute_url</code>	85
6.6	Test di <i>FAQ</i> : creazione della classe e setup dei dati	86
6.7	Test di <i>FAQ</i> : etichetta relativa al nome della FAQ	86
6.8	Test di <i>FAQ</i> : verifica del <code>max_length</code>	86
6.9	Test di <i>FAQ</i> : rotta <code>/faq/</code>	86
6.10	Test di <i>Fattura</i> : creazione della classe e setup dei dati	87
6.11	Test di <i>Fattura</i> : etichetta relativa al codice cliente	87
6.12	Test di <i>Fattura</i> : etichetta relativa alla quantità	87
6.13	Test di <i>Fattura</i> : verifica del <code>max_length</code>	87
6.14	Test di <i>Fattura</i> : funzione <code>validate_price</code>	87
6.15	Test di <i>Prenotazione</i> : etichetta relativa alla prescrizione	88
6.16	Test di <i>Prenotazione</i> : rotta <code>/contact/</code>	88
6.17	Test di <i>Prenotazione</i> : funzione <code>clean_data</code>	88
6.18	Test di <i>Prodotto</i> : creazione della classe e setup dei dati	88
6.19	Test di <i>Prodotto</i> : etichetta relativa alla marca	89
6.20	Test di <i>Prodotto</i> : funzione <code>validate_price</code>	89
6.21	Test di <i>Prodotto</i> : funzione <code>get_absolute_url</code>	89
6.22	Test di <i>Tariffa</i> : creazione della classe e setup dei dati	89
6.23	Test di <i>Tariffa</i> : etichetta relativa alla descrizione	89
6.24	Test di <i>Tariffa</i> : verifica del <code>max_length</code>	90
6.25	Test di <i>Tariffa</i> : funzione <code>get_absolute_url</code>	90

Elenco delle tabelle

1.1	Glossario dei termini relativi alla descrizione del sistema nel linguaggio naturale	7
2.1	Requisiti funzionali: Gestione Agenda	9
2.2	Requisiti funzionali: Gestione Autenticazione	10
2.3	Requisiti funzionali: Gestione FAQ	10
2.4	Requisiti Funzionali: Gestione Fatture	10
2.5	Requisiti funzionali: Gestione Prenotazioni	11
2.6	Requisiti funzionali: Gestione Prodotti	11
2.7	Requisiti funzionali: Gestione Tariffe	11
2.8	Dettaglio dei requisiti non funzionali relativi al nostro sistema	12
3.1	Gestionale_Infermiera: elementi della classe ControllerHome	14
3.2	Agenda: elementi della classe Appuntamento	15
3.3	Agenda: associazioni della classe Appuntamento	15
3.4	Agenda: elementi della classe Meta	16
3.5	Agenda: associazioni della classe Meta	16
3.6	Agenda: elementi della classe AgendaAdmin	16
3.7	Agenda: associazioni della classe AgendaAdmin	16
3.8	Agenda: elementi della classe EventCalendar	17
3.9	Agenda: associazioni della classe EventCalendar	17
3.10	Agenda: elementi della classe AgendaConfig	17
3.11	Agenda: associazioni della classe AgendaConfig	17
3.12	Tariffa: elementi della classe Tariffa	18
3.13	Tariffa: associazioni della classe Tariffa	18
3.14	Tariffa: elementi della classe Meta	19
3.15	Tariffa: associazioni della classe Meta	19
3.16	Tariffa: elementi della classe TariffaAdmin	19
3.17	Tariffa: associazioni della classe TariffaAdmin	19
3.18	Tariffa: elementi della classe TariffaConfig	19
3.19	Tariffa: associazioni della classe TariffaConfig	20
3.20	Fattura: elementi della classe Fattura	21
3.21	Fattura: associazioni della classe Fattura	21
3.22	Fattura: elementi della classe Meta	21
3.23	Fattura: associazioni della classe Meta	21
3.24	Fattura: elementi della classe TariffaFatt	22

3.25	Fattura: associazioni della classe TariffaFatt	22
3.26	Fattura: elementi della classe FatturaAdmin	22
3.27	Fattura: associazioni della classe FatturaConfig	22
3.28	Fattura: elementi della classe FatturaInline	23
3.29	Fattura: associazioni della classe FatturaInline	23
3.30	Fattura: elementi della classe ControllerFattura	23
3.31	Fattura: associazioni della classe ControllerFattura	23
3.32	Fattura: elementi della classe FatturaConfig	23
3.33	Fattura: associazioni della classe FatturaConfig	23
3.34	Prodotto: elementi della classe Prodotto	24
3.35	Prodotto: associazioni della classe Prodotto	25
3.36	Prodotto: elementi della classe Meta	25
3.37	Prodotto: associazioni della classe Meta	25
3.38	Prodotto: elementi della classe ProdottoAdmin	25
3.39	Prodotto: elementi della classe ProdottoAdmin	25
3.40	Prodotto: elementi della classe ProdottoConfig	26
3.41	Prodotto: associazioni della classe ProdottoConfig	26
3.42	Prenotazione: elementi della classe Prenotazione	27
3.43	Prenotazione: associazioni della classe Prenotazione	27
3.44	Prenotazione: elementi della classe ControllerPrenotazione	27
3.45	Prenotazione: associazioni della classe ControllerPrenotazione	28
3.46	Prenotazione: elementi della classe PrenotazioneConfig	28
3.47	Prenotazione: associazioni della classe PrenotazioneConfig	28
3.48	FAQ: elementi della classe FAQ	29
3.49	FAQ: associazioni della classe FAQ	29
3.50	FAQ: elementi della classe FAQAdmin	29
3.51	FAQ: associazioni della classe FAQAdmin	29
3.52	FAQ: elementi della classe ControllerFAQ	30
3.53	FAQ: associazioni della classe ControllerFaq	30
3.54	FAQ: elementi della classe FaqConfig	30
3.55	FAQ: associazioni della classe FaqConfig	30
4.1	Caso d'uso relativo alla visualizzazione di un appuntamento	33
4.2	Caso d'uso relativo all'inserimento di un appuntamento	33
4.3	Caso d'uso relativo alla modifica di un appuntamento	34
4.4	Caso d'uso relativo all'eliminazione di un appuntamento	34
4.5	Caso d'uso relativo alla ricerca di un appuntamento	35
4.6	Caso d'uso relativo alla visualizzazione di una tariffa	36
4.7	Caso d'uso relativo all'inserimento di una tariffa	36
4.8	Caso d'uso relativo alla modifica di una tariffa	37
4.9	Caso d'uso relativo all'eliminazione di una tariffa	37
4.10	Caso d'uso relativo alla ricerca di una tariffa	38
4.11	Caso d'uso relativo alla visualizzazione di una fattura	39
4.12	Caso d'uso relativo all'inserimento di una fattura	39
4.13	Caso d'uso relativo alla modifica di una fattura	40
4.14	Caso d'uso relativo all'eliminazione di una fattura	40
4.15	Caso d'uso relativo alla generazione di un file PDF relativo ad una fattura	41
4.16	Caso d'uso relativo alla ricerca di una fattura	41
4.17	Caso d'uso relativo alla visualizzazione di un prodotto	42
4.18	Caso d'uso relativo all'inserimento di un prodotto	43
4.19	Caso d'uso relativo alla modifica di un prodotto	43

4.20	Caso d'uso relativo all'eliminazione di un prodotto	44
4.21	Caso d'uso relativo alla ricerca di un prodotto	44
4.22	Caso d'uso relativo alla visualizzazione di una FAQ	45
4.23	Caso d'uso relativo all'inserimento di una FAQ	46
4.24	Caso d'uso relativo alla modifica di una FAQ	46
4.25	Caso d'uso relativo all'eliminazione di una FAQ	47
4.26	Caso d'uso relativo alla ricerca di una FAQ	47
4.27	Caso d'uso relativo alla prenotazione di un appuntamento	48
4.28	Caso d'uso relativo al login da parte dell'amministratore	49
4.29	Caso d'uso relativo al logout da parte dell'amministratore	50
4.30	Caso d'uso relativo alla prenotazione di un appuntamento	50

Il software ha un'importanza che non può essere trascurata; infatti, molti aspetti del mondo moderno non potrebbero funzionare senza di esso. Le infrastrutture e le aziende, la produzione e la distribuzione industriale, la finanza e l'intrattenimento: ogni aspetto si appoggia sul software.

In risposta alle esigenze di software crescenti, nasce l'ingegneria del software, che non comprende semplicemente il programma in sé ma anche la documentazione ad esso correlata.

L'ingegneria del software viene spesso criticata perché inadeguata al moderno sviluppo, ma è, in realtà, fondamentale oggi giorno nella creazione di programmi che hanno una crescente complessità: la metodologia analizzata è proprio ciò che contraddistingue un programma professionale dalla programmazione individuale. Per capirne l'importanza, riportiamo una frase di Ian Sommerville, noto ingegnere del software britannico, che dice: *"In generale, gli ingegneri del software adottano un approccio sistematico e organizzato al loro lavoro, poiché questo è spesso il modo più efficace per produrre software di alta qualità"*.

Questo è il concetto da cui è necessario partire: un ingegnere non scrive codice senza un approccio strutturato e definito, bensì definisce, modella ed elabora delle informazioni per raggiungere un obiettivo.

Avendo definito il metodo con il quale svolgeremo il nostro lavoro, è opportuno evidenziare la tendenza odierna nel possedere un software per gestire al meglio gli aspetti del proprio lavoro. Un cliente interessato si rivolge ad un professionista, ma la richiesta, spesso, non possiede una struttura ben definita e saremo noi, quindi, ad indirizzare verso un determinato schema ciò che a primo impatto potrebbe risultare astratto e privo di significato.

I dati raccolti subiscono un raffinamento, vengono modellati e strutturati per poter generare qualcosa di concreto e solido e che riesca a conciliare le richieste del cliente con la tecnologia utilizzata per soddisfarle.

Il presente elaborato non è dunque frutto di processi casuali, ma di un'analisi articolata e strutturata.

Esso è composto da sei capitoli strutturati come specificato di seguito:

- Nel Capitolo 1 riporteremo l'intervista effettuata al cliente e il conseguente raffinamento nel linguaggio naturale.
- Nel Capitolo 2 analizzeremo i requisiti, funzionali e non, che derivano dalle esigenze messe in evidenza nel capitolo precedente.
- Nel Capitolo 3 riporteremo le classi che costituiscono il nostro software, con i relativi metodi ed attributi che le caratterizzano e le relazioni che intercorrono tra di esse.

- Nel Capitolo 4 analizzeremo i diagrammi di comportamento, che definiscono cosa il nostro programma può fare in relazione all'interazione di un utente e come si articola la consecutio temporum delle varie operazioni.
- Nel Capitolo 5 riporteremo i mockup, ovvero la prima concretizzazione dei processi sviluppati fino a quel momento.
- Nel Capitolo 6 mostreremo il framework ed il linguaggio utilizzati nella parte implementativa e come il nostro software è stato testato attraverso degli esempi concreti di codice.

Definizione del contesto di riferimento

In questo capitolo iniziale l'intento è quello di analizzare le specifiche dettate dal cliente partendo da un'intervista mirata per poi raffinare la descrizione del contesto di riferimento attraverso una descrizione nel linguaggio naturale. Analizzeremo, attraverso le informazioni raccolte, quali possono essere le parole chiave ed i termini che ci aiuteranno a comprendere meglio le richieste, evidenziandole in un opportuno glossario. Questo processo iniziale è per noi fondamentale per sviluppare i requisiti del progetto.

1.1 Intervista

Il giorno 2 Aprile 2022 abbiamo deciso di contattare L., infermiera libera professionista, chiedendo un appuntamento privato. L'intento dell'intervista è stato quello di soffermarsi su informazioni di carattere generale, in un primo momento, per poi scendere successivamente ad un livello più specifico: in questo modo siamo riusciti ad individuare le problematiche che la progettazione avrebbe potuto sollevare.

Di seguito viene riportata la conversazione:

Buongiorno L.! Come saprai, ho intenzione di realizzare un software che possa aiutarti nel tuo lavoro, semplificando la gestione dei tuoi appuntamenti. Se ritieni di essere interessata ad un programma di questo tipo procederei ponendoti alcune domande che mi permettano di realizzare il lavoro in maniera eccellente.

Buongiorno! Ti ringrazio, un programma di questo tipo potrebbe senz'altro essermi utile; sono a tua disposizione per qualsiasi domanda.

Per prima cosa inizierei con il chiederti come si articola la tua giornata.

Non ho una giornata tipo: in base agli appuntamenti presi la strutturo di volta in volta. Può capitare di avere giornate più o meno libere, mentre altre prevedono poche pause. Per quanto mi è possibile cerco un equilibrio, così da non averne di eccessivamente stressanti.

E' possibile identificare quali sono le mansioni principali da te svolte, indipendentemente dagli orari lavorativi?

Certamente, generalmente ho due tipologie di clienti: pazienti che chiedono prestazioni a domicilio, e ambulatori privati, i quali, a corto di personale, si rivolgono a liberi professioni-

sti come me. Nel primo caso le mansioni dipendono dalle necessità dei singoli, come punture o somministrazione di antibiotici; nel secondo caso, data anche la situazione attuale, il lavoro consiste nell'effettuare molti tamponi.

E come vengono gestiti gli appuntamenti in genere?

Solitamente vengo contattata telefonicamente, ed in base agli impegni correnti gestisco l'appuntamento. Uso un'agenda per poter memorizzare con chi, dove e quando dovrò lavorare. Quando un paziente mi chiama, in generale devo conoscere: i dati anagrafici, l'indirizzo, il codice fiscale, e se il paziente dispone già del materiale necessario alla cura, oppure se devo procurarlo io; infine, ho necessità di sapere se ha una prescrizione del medico.

Come mai è necessario conoscere queste ultime due informazioni?

Per quanto riguarda il materiale, il discorso è prettamente economico: se il paziente non dispone del materiale, sarò io a procurarmelo per poi addebitare l'eventuale spesa. Per quanto riguarda la prescrizione del medico, se il paziente non ne è in possesso, dovrò porre delle domande riguardo l'anamnesi, ossia la 'storia clinica' del paziente. Quest'ultima è necessaria soprattutto per conoscere le eventuali allergie.

All'interno di quali fasce orarie e in quali giorni settimanali risulta operativa?

Non ho delle vere e proprie fasce orarie, lavoro molto a seconda delle esigenze dei clienti, potenzialmente anche di notte e durante le giornate festive. Tuttavia nella maggior parte dei casi, le fasce lavorative vanno dalle ore nove della mattina fino alle ore venti della sera, spesso con intervalli non lavorativi nel mezzo.

Per quanto riguarda la prenotazione, potrebbe interessarti la possibilità di ricevere prenotazioni anche online tramite posta elettronica?

Certamente, purchè questo risulti in qualche modo compatibile con le prenotazioni effettuate tramite telefono.

L'orario dedicato ad un paziente può dipendere anche dall'eventuale farmaco somministrato?

Sì, esattamente. Ovviamente c'è una certa flessibilità, ma non in maniera assoluta; se, ad esempio, un paziente necessita di un antibiotico oppure di un anticoagulante, è necessario somministrarlo ad un certo orario per poi mantenerlo per tutta la durata della somministrazione.

Molto bene; per quanto riguarda il tariffario: come viene gestito? Come procedi per la fatturazione?

Relativamente ai pazienti privati, ho un tariffario che dipende da cosa viene richiesto, a cui viene aggiunta la spesa degli eventuali materiali utilizzati, che addebito attraverso la ricevuta della spesa fatta. La fattura viene fatta una volta svolto il lavoro. Solitamente la fatturazione è cartacea, ma, se richiesta, spedisco tramite mail anche quella elettronica, dato che quest'ultima non risulta essere obbligatoria.

Per quanto riguarda le cliniche private, il compenso dipende dal contratto.

Per la fatturazione elettronica, ti appoggi a qualche software in particolare? Quali dati sono richiesti?

Utilizzo Excel per poter generare la fattura: i dati richiesti sono standard, eccezione fatta per i dati personali dei pazienti o dei privati presso cui lavoro.

Ritieni che una particolare parte del tuo processo lavorativo, soprattutto a livello economico, potrebbe essere ottimizzato? Ad esempio, per quanto la riguarda il possedimento dei materiali?

Credo di sì; solitamente faccio fatica a conteggiare bene le ore in cui ho lavorato, ed anche a ricordarmi il compenso richiesto per una determinata situazione.

Riguardo ai materiali, solitamente, tendo ad acquistare dei prodotti di base a prescindere dalle richieste dei pazienti (come disinfettante e guanti); avrei bisogno di avere una sorta di inventario, per evitare di acquistare materiale che già possiedo.

Ti ringrazio; per ora queste informazioni sono sufficienti per poter procedere con il lavoro. Buona giornata.

Ti ringrazio, buona giornata!

1.2 Descrizione del sistema in linguaggio naturale

Il progetto proposto consiste nella realizzazione di un software per la gestione delle attività di un'infermiera freelance.

Gli aspetti di interesse riguardano:

- la possibilità, da parte dei clienti, di prenotarsi tramite mail per un appuntamento;
- la capacità di gestire un'agenda per i vari appuntamenti;
- la capacità di generare una fattura inserendo i dati opportuni;
- la possibilità di gestire un piccolo inventario dei prodotti utilizzati;
- la possibilità di gestire un piccolo tariffario;
- la possibilità di gestire delle FAQ.

Il sito dovrà prevedere solamente due livelli di utenza: quello generico per il cliente, il quale non richiede registrazione, e quello dell'amministratore, in questo caso la nostra infermiera, che richiede un'autenticazione.

Se un paziente intende prenotarsi tramite email, dovrà fornire: nome, cognome, indirizzo (città, via e numero civico), codice fiscale, servizio richiesto. In base alla richiesta, sarà possibile indicare se si ha necessità di materiale specifico, o se il richiedente ne è già in possesso. Infine, sarà importante indicare se si è forniti della prescrizione del medico: sarà molto importante segnalare la mancanza di quest'ultima (poichè ne consegue l'anamnesi). Il paziente dovrà, infine, indicare una preferenza riguardo il giorno e l'orario dell'appuntamento, che verrà poi confermato dall'amministratore.

Per entrare nell'area riservata sarà necessaria un'autenticazione tramite username e password. L'area riservata sarà suddivisa in quattro zone principali: una per ogni argomento di interesse:

- La prima riguarderà l'agenda: sarà possibile visualizzare l'agenda con gli appuntamenti presi. Il software prevederà la possibilità di inserire, modificare ed eliminare un eventuale appuntamento. Ogni appuntamento verrà identificato in maniera univoca; inoltre dovrà essere caratterizzato dal giorno e dall'orario, oltre che da un'eventuale piccola descrizione e da un titolo.
- La seconda riguarderà la gestione delle fatture: in questo caso il software dovrà garantire la possibilità di visualizzare tutte le fatture che sono state emesse in passato. Per poter generare una fattura, l'amministratore dovrà compilare un form, con i dati necessari; a questo punto sarà data la possibilità di generare un file pdf con le informazioni inserite. Sarà possibile eliminare le fatture create. Ogni fattura sarà identificata da un codice univoco.
- La terza riguarda l'inventario dei prodotti: l'amministratore sarà in grado di inserire, modificare ed eliminare un eventuale prodotto. Ogni prodotto dovrà avere un nome, un'eventuale descrizione, la marca ed il prezzo. In questo modo, prodotti che hanno nome simile saranno identificati in maniera univoca.
- La quarta e ultima riguarda il tariffario; anche in questo caso dovrà essere data la possibilità di visualizzare le tariffe presenti. Ogni tariffa dovrà avere un nome, una descrizione e, ovviamente, un prezzo; dovrà essere garantita la possibilità di modificare una tariffa e di eliminarla.

Un ulteriore aspetto degno di analisi riguarda la grafica del software: dovrà essere semplice ed intuitivo effettuare una prenotazione da parte dei clienti (ossia un nuovo utente deve poter effettuare un basso numero di tentativi per individuare le aree di interesse), inoltre, si dovrà prevedere una sezione *faq* riguardante i dubbi più frequenti.

Per quanto riguarda la parte dell'amministratore, si richiede, anche in questo caso, semplicità; il nostro amministratore non è un esperto di informatica; il sito dovrà velocizzare i processi di cui si è parlato sopra; dunque si darà più importanza alla velocità di esecuzione (quindi le operazioni che l'amministratore vuole svolgere non dovranno essere nascoste, ma poste in evidenza), piuttosto che all'aspetto grafico in sé, che resterà per lo più basilico.

1.3 Glossario dei termini

Il glossario dei termini contiene tutte le parole che possono generare ambiguità con la relativa spiegazione.

Di seguito il glossario dei termini della nostra descrizione.

Termine	Descrizione	Sinonimi
Prenotazione	Atto mediante il quale un paziente può richiedere un appuntamento	
Cliente	Chiunque desidera effettuare una prenotazione tramite email o telefono	Paziente
Prescrizione	Documento fornito dal proprio medico di base. Consente al paziente di prenotare visite specialistiche, esami diagnostici e di poter assumere determinati farmaci	

Termine	Descrizione	Sinonimi
Servizio	Ciò che viene richiesto dal cliente (ad esempio, una puntura, la somministrazione di un farmaco, oppure un tampone)	Prestazione
Materiale specifico	Il materiale che viene richiesto per effettuare la specifica prestazione (potrebbero essere dei guanti, oppure un particolare medicinale)	Prodotto
Agenda	Raccolta degli appuntamenti dei pazienti. Viene gestita direttamente dall'amministratore	
Fattura	Documento nel quale viene riportata la quantità di denaro che il cliente deve corrispondere all'amministratore per i beni o servizi forniti	Ricevuta
Prodotto	Materiale che l'amministratore deve acquistare per poter svolgere il proprio lavoro (comprende il materiale specifico richiesto dal paziente, oltre che il materiale di base)	
Tariffario	Elenco dei prezzi relativi ai servizi svolti	
Amministratore	Colui che gestisce il software, nel nostro caso specifico l'infermiera	Admin
Anamnesi	Storia clinica del paziente. Consiste nella raccolta e nello studio critico dei sintomi e dei fatti di interesse medico riferiti dal paziente o dai suoi familiari	
Pdf	Portable Document Format: il formato di file utilizzato per le fatture.	
Codice univoco	Codice identificativo unico (può identificare, ad esempio, una fattura o un prodotto)	Id
Faq	Frequently Asked Questions: una serie di risposte stilate in risposta alle domande che vengono poste più frequentemente dagli utilizzatori del portale.	

Tabella 1.1: Glossario dei termini relativi alla descrizione del sistema nel linguaggio naturale

Specifica e analisi dei requisiti

In questo capitolo individueremo le specifiche del sistema attraverso l'analisi dei requisiti. I requisiti di un sistema corrispondono alla descrizione dei servizi che il sistema stesso dovrà fornire e, allo stesso tempo, mettono in evidenza i vincoli operativi. Analizzeremo, quindi, separatamente i requisiti funzionali e quelli non funzionali, mostrando accuratamente le caratteristiche di entrambi.

In primo luogo è opportuno separare i requisiti in funzionali e non funzionali, come mostrato in Figura 2.1:

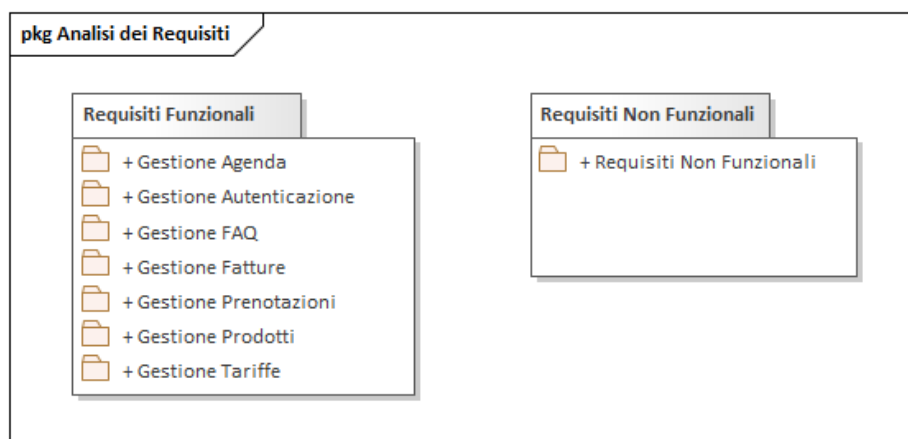


Figura 2.1: Analisi dei requisiti funzionali e non funzionali, schema generale

2.1 Requisiti funzionali

I requisiti funzionali dichiarano, nel linguaggio naturale, quali servizi il sistema dovrebbe fornire. Nel nostro caso i requisiti funzionali riguardano le operazioni che verranno svolte dall'amministratore o da un potenziale cliente.

I requisiti funzionali relativi al nostro sistema sono riportati in Figura 2.2. Come si evince dalla figura, essi si possono suddividere in sette macro-aree, che si occupano dell'agenda, dell'autenticazione, delle FAQ, delle fatture, delle prenotazioni, dei prodotti e delle tariffe, rispettivamente. Esamineremo ciascuna macro-area di requisiti nelle prossime sottosezioni.

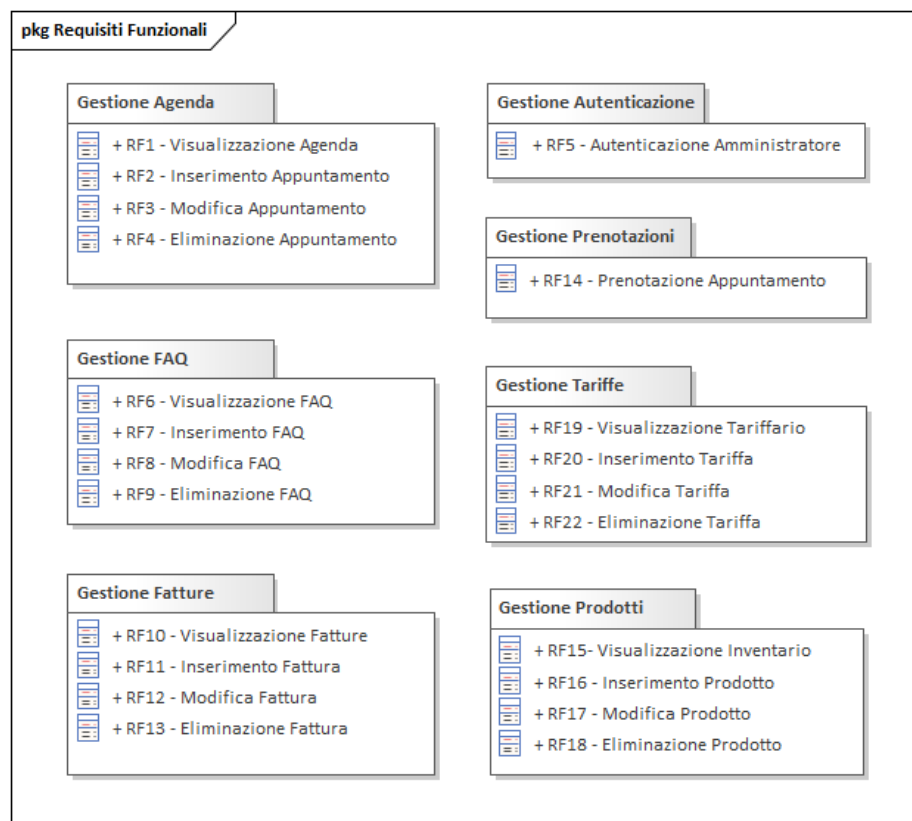


Figura 2.2: Analisi dei requisiti funzionali relativi al nostro sistema

2.1.1 Gestione Agenda

Nella tabella 2.1 vengono riportati i requisiti funzionali relativi alla gestione dell'agenda.

Requisito	Descrizione
RF1 - Visualizzazione Agenda	Il sistema dovrà consentire la visualizzazione di tutti gli appuntamenti registrati
RF2 - Inserimento Appuntamento	Il sistema dovrà consentire l'inserimento di un appuntamento
RF3 - Modifica Appuntamento	Il sistema dovrà gestire la modifica di un appuntamento
RF4 - Eliminazione Appuntamento	Il sistema dovrà consentire l'eliminazione di un appuntamento

Tabella 2.1: Requisiti funzionali: Gestione Agenda

2.1.2 Gestione Autenticazione

Nella tabella 2.2 vengono riportati i requisiti funzionali relativi alla gestione dell'autenticazione.

Requisito	Descrizione
RF5 - Autenticazione Amministratore	Il sistema dovrà consentire all'amministratore di autenticarsi

Tabella 2.2: Requisiti funzionali: Gestione Autenticazione

2.1.3 Gestione FAQ

Nella tabella 2.3 vengono riportati i requisiti funzionali relativi alla gestione delle FAQ.

Requisito	Descrizione
RF6 - Visualizzazione FAQ	Il sistema dovrà consentire la visualizzazione di tutte le FAQ registrate
RF7 - Inserimento FAQ	Il sistema dovrà consentire l'inserimento di una nuova FAQ
RF8 - Modifica FAQ	Il sistema dovrà gestire la modifica di una FAQ
RF9 - Eliminazione FAQ	Il sistema dovrà gestire l'eliminazione di una FAQ

Tabella 2.3: Requisiti funzionali: Gestione FAQ

2.1.4 Gestione Fatture

Nella tabella 2.4 vengono riportati i requisiti funzionali relativi alla gestione delle fatture.

Requisito	Descrizione
RF10 - Visualizzazione Fatture	Il sistema dovrà consentire la visualizzazione di tutte le fatture effettuate
RF11 - Inserimento Fattura	Il sistema dovrà consentire l'inserimento di una nuova fattura
RF12 - Modifica Fattura	Il sistema dovrà gestire la modifica di una fattura
RF13 - Eliminazione Fattura	Il sistema dovrà gestire l'eliminazione di una fattura

Tabella 2.4: Requisiti Funzionali: Gestione Fatture

2.1.5 Gestione Prenotazioni

Nella tabella 2.5 vengono riportati i requisiti funzionali relativi alla gestione delle prenotazioni.

Requisito	Descrizione
RF14 - Prenotazione appuntamento	Il sistema dovrà consentire al cliente di effettuare una prenotazione tramite email inserendo le informazioni richieste

Tabella 2.5: Requisiti funzionali: Gestione Prenotazioni

2.1.6 Gestione Prodotti

Nella tabella 2.6 vengono riportati i requisiti funzionali relativi alla gestione dei prodotti.

Requisito	Descrizione
RF15 - Visualizzazione Inventario	Il sistema dovrà consentire la visualizzazione di tutti i prodotti registrati
RF16 - Inserimento Prodotto	Il sistema dovrà consentire l'inserimento di un nuovo prodotto
RF17 - Modifica Prodotto	Il sistema dovrà gestire la modifica di un prodotto
RF18 - Eliminazione Prodotto	Il sistema dovrà gestire l'eliminazione di un prodotto

Tabella 2.6: Requisiti funzionali: Gestione Prodotti

2.1.7 Gestione Tariffe

Nella tabella 2.7 vengono riportati i requisiti funzionali relativi alla gestione delle tariffe.

Requisito	Descrizione
RF19 - Visualizzazione Tariffario	Il sistema dovrà consentire la visualizzazione di tutte le tariffe registrate
RF20 - Inserimento Tariffa	Il sistema dovrà consentire l'inserimento di una nuova tariffa
RF21 - Modifica Tariffa	Il sistema dovrà gestire la modifica di una tariffa
RF22 - Eliminazione Tariffa	Il sistema dovrà gestire l'eliminazione di una tariffa

Tabella 2.7: Requisiti funzionali: Gestione Tariffe

2.2 Requisiti non funzionali

I requisiti non funzionali rappresentano i vincoli, soprattutto di natura tecnologica, che il sistema deve garantire.

I requisiti non funzionali relativi al nostro sistema sono riportati in Figura 2.3.

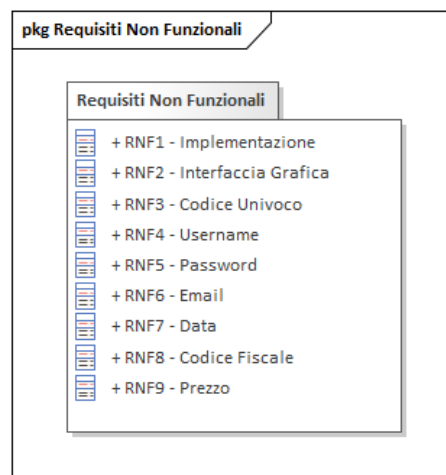


Figura 2.3: Analisi dei requisiti non funzionali

Nella tabella 2.8 vengono riportati i requisiti non funzionali relativi al nostro sistema.

Requisito	Descrizione
RNF1 - Implementazione	Il sistema dovrà essere scritto mediante l'utilizzo del framework Django, nel linguaggio di programmazione Python
RNF2 - Interfaccia grafica	Il sistema dovrà essere dotato di un'interfaccia grafica semplice
RNF3 - Codice univoco	Il sistema non dovrà consentire l'impiego di codici identificativi identici
RNF4 - Username	Il sistema dovrà permettere solo username con una lunghezza minima di 8 caratteri
RNF5 - Password	Il sistema dovrà permettere solo password con una lunghezza minima di 10 caratteri
RNF6 - Email	Il sistema non dovrà consentire email che non rispettino il formato corretto
RNF7 - Data	Il sistema non dovrà consentire di inserire date antecedenti la data odierna
RNF8 - Codice Fiscale	Il sistema dovrà permettere codici fiscali di esattamente sedici caratteri
RNF9 - Prezzo	Il sistema non dovrà consentire di inserire prezzi negativi

Tabella 2.8: Dettaglio dei requisiti non funzionali relativi al nostro sistema

3.2 Gestionale_Infermiera

Questo primo package costituisce la parte centrale del software poichè, in esso, sono definiti le impostazioni di progetto e i percorsi verso le pagine principali. Ogni applicazione all'interno del progetto deve necessariamente fare riferimento ad esso.

L'unica classe presente è quella che gestisce le viste relative all'utente, come mostrato in Figura 3.2.

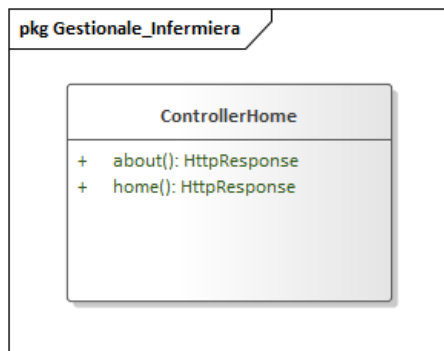


Figura 3.2: Package Gestionale_Infermiera e relative classi

3.2.1 ControllerHome

`ControllerHome` è l'unica classe creata in questo package ed è situata all'interno del file `views.py`. Come illustrato precedentemente, la classe gestisce le viste relative all'utente. Le sue componenti sono mostrate nella Tabella 3.1.

Elemento	Nome	Tipo
Funzione	<code>home()</code>	<code>HttpResponse</code>
Funzione	<code>about()</code>	<code>HttpResponse</code>

Tabella 3.1: Gestionale_Infermiera: elementi della classe `ControllerHome`

3.3 Agenda

Il package `Agenda` gestisce la visualizzazione, l'inserimento, la modifica e l'eliminazione di un appuntamento. Le classi che lo costituiscono sono mostrate in Figura 3.3.

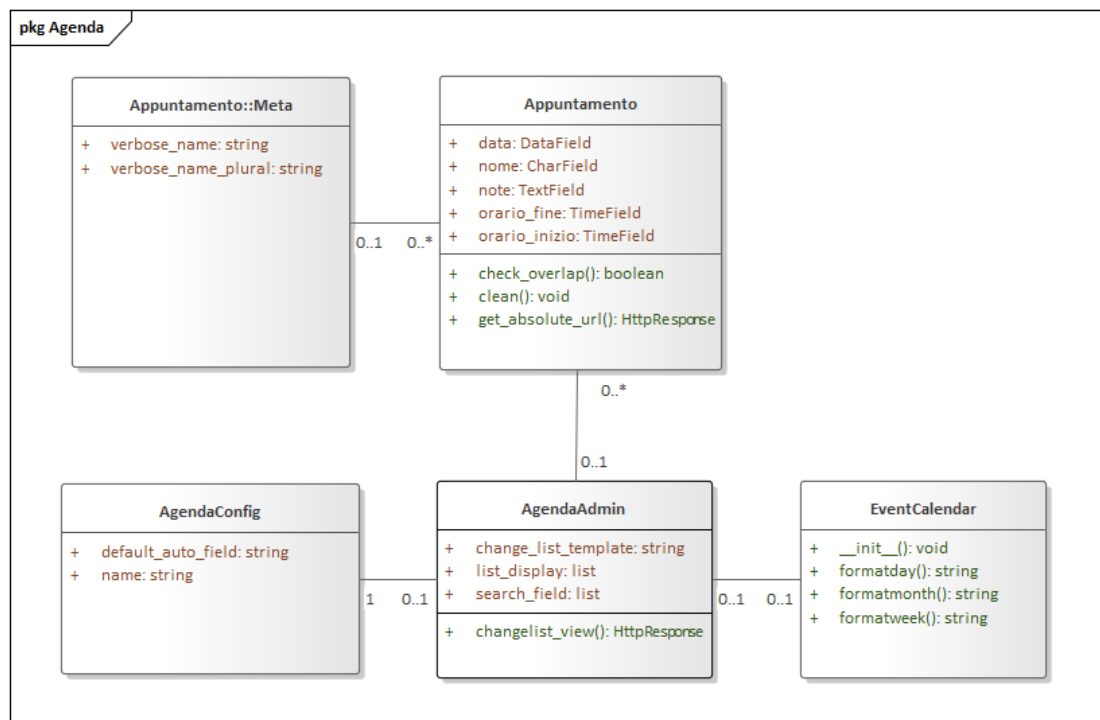


Figura 3.3: Package Agenda e relative classi

3.3.1 Appuntamento

La classe `Appuntamento` costituisce il modello; quindi le caratteristiche che dovrà possedere un generico appuntamento. Essa è situata nel file `models.py`. Gli elementi che la costituiscono sono mostrati in Tabella 3.2 mentre le associazioni sono mostrate in Tabella 3.3.

Elemento	Nome	Tipo
Attributo	<code>data</code>	<code>DataField</code>
Attributo	<code>nome</code>	<code>CharField</code>
Attributo	<code>note</code>	<code>TextField</code>
Attributo	<code>orario_fine</code>	<code>TimeField</code>
Attributo	<code>orario_inizio</code>	<code>TimeField</code>
Funzione	<code>check_overlap</code>	<code>boolean</code>
Funzione	<code>clean()</code>	<code>void</code>
Funzione	<code>get_absolute_url</code>	<code>HttpResponse</code>

Tabella 3.2: Agenda: elementi della classe `Appuntamento`

Associazione	Cardinalità
<code>Appuntamento - Meta</code>	<code>0..* - 0..1</code>
<code>Appuntamento - AgendaAdmin</code>	<code>0..* - 0..1</code>

Tabella 3.3: Agenda: associazioni della classe `Appuntamento`

3.3.2 Meta

La classe `Meta` consente di modificare alcuni comportamenti della classe `Appuntamento` e, per questo motivo, è definita all'interno di quest'ultima. Gli elementi della classe sono mostrati in Tabella 3.4 mentre le associazioni della classe sono mostrate in Tabella 3.5.

Elemento	Nome	Tipo
Attributo	<code>verbose_name</code>	<code>string</code>
Attributo	<code>verbose_name_plural</code>	<code>string</code>

Tabella 3.4: Agenda: elementi della classe `Meta`

Associazione	Nome
Meta - Appuntamento	0..1 - 0..*

Tabella 3.5: Agenda: associazioni della classe `Meta`

3.3.3 AgendaAdmin

La classe `AgendaAdmin` è definita in `admin.py` e modifica la vista standard definita da Django generando un calendario più semplice da utilizzare. Gli elementi della classe sono definiti in Tabella 3.6, mentre le associazioni della classe sono mostrate in Tabella 3.7.

Elemento	Nome	Tipo
Attributo	<code>change_list_template</code>	<code>string</code>
Attributo	<code>list_display</code>	<code>string</code>
Attributo	<code>search_field</code>	<code>list</code>
Funzione	<code>changelist_view</code>	<code>HttpResponse</code>

Tabella 3.6: Agenda: elementi della classe `AgendaAdmin`

Associazione	Nome
AgendaAdmin - Appuntamento	0..1 - 0..*

Tabella 3.7: Agenda: associazioni della classe `AgendaAdmin`

3.3.4 EventCalendar

La classe `EventCalendar` consente di personalizzare il calendario definito nella classe `AgendaAdmin` ed è situato nel file `utils.py`. Gli elementi sono stati inseriti nella tabella 3.8 mentre le associazioni della classe sono mostrate in Tabella 3.9.

Elemento	Nome	Tipo
Funzione	<code>__init__()</code>	void
Funzione	<code>formatday()</code>	string
Funzione	<code>formatmonth()</code>	string
Funzione	<code>formatweek()</code>	string

Tabella 3.8: Agenda: elementi della classe EventCalendar

Associazione	Nome
EventCalendar - AgendaAdmin	0..1 - 0..1

Tabella 3.9: Agenda: associazioni della classe EventCalendar

3.3.5 AgendaConfig

La classe `AgendaConfig` contiene gli elementi di configurazione dell'intero package; senza di essa il funzionamento di tutte le altre classi verrebbe meno. Gli elementi sono mostrati in Tabella 3.10. Le associazioni della classe sono mostrate in Tabella 3.11.

Elemento	Nome	Tipo
Attributo	<code>default_auto_field</code>	string
Attributo	<code>name</code>	string

Tabella 3.10: Agenda: elementi della classe AgendaConfig

Associazione	Nome
AgendaAdmin - Appuntamento	0..1 - 0..*

Tabella 3.11: Agenda: associazioni della classe AgendaConfig

3.4 Tariffa

Il package `Tariffa` consente di inserire, modificare ed eliminare una prestazione infermieristica. Le classi che costituiscono il package sono mostrate in Figura 3.4.

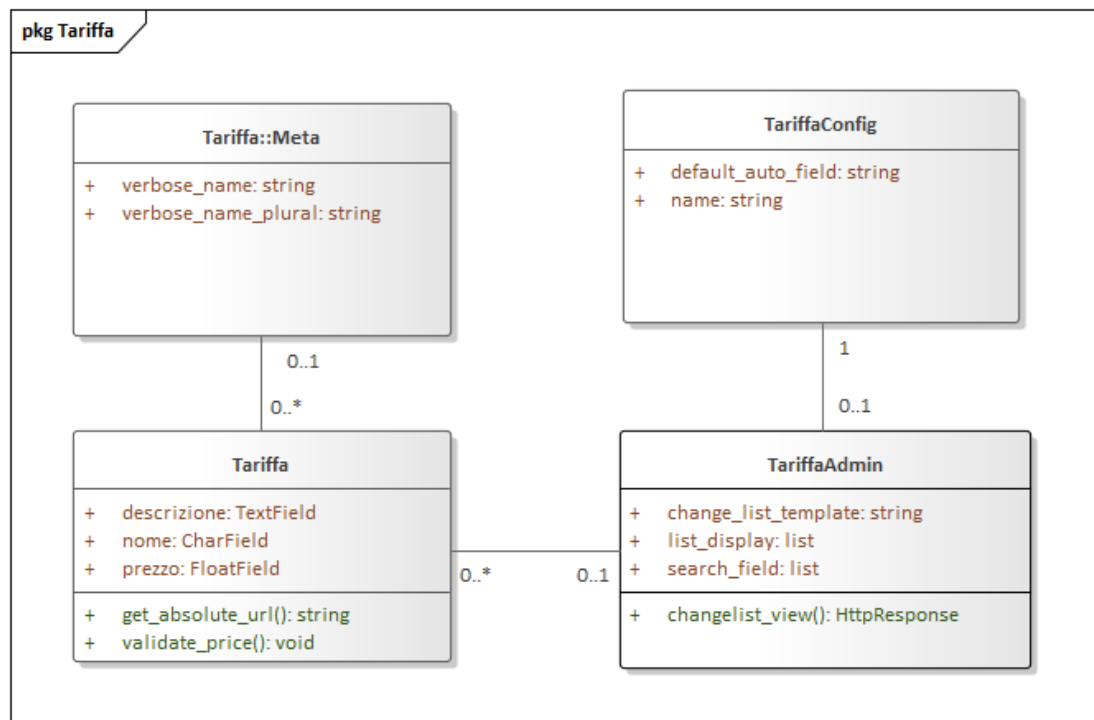


Figura 3.4: Package Tariffa e relative classi

3.4.1 Tariffa

La classe `Tariffa` costituisce il modello con cui l'amministratore gestirà le tariffe all'interno del software ed è situata nel file `model.py`. Gli elementi che la costituiscono sono mostrate nella Tabella 3.12 mentre le associazioni della classe sono mostrate in Tabella 3.13.

Elemento	Nome	Tipo
Attributo	descrizione	TextField
Attributo	nome	CharField
Attributo	prezzo	FloatField
Funzione	<code>get_absolute_url()</code>	string
Funzione	prezzo	void

Tabella 3.12: Tariffa: elementi della classe Tariffa

Associazione	Nome
Tariffa - TariffaAdmin	0..* - 0..1
Tariffa - Meta	0..* - 0..1

Tabella 3.13: Tariffa: associazioni della classe Tariffa

3.4.2 Meta

La classe `Meta` consente di modificare alcuni comportamenti della classe `Tariffa` e, per questo motivo, è definita all'interno di quest'ultima. Gli elementi della classe sono mostrati

in Tabella 3.14 mentre le associazioni della classe sono mostrate in Tabella 3.15.

Elemento	Nome	Tipo
Attributo	verbose_name	string
Attributo	verbose_name_plural	string

Tabella 3.14: Tariffa: elementi della classe Meta

Associazione	Nome
Meta - Tariffa	0..1 - 0..*

Tabella 3.15: Tariffa: associazioni della classe Meta

3.4.3 TariffaAdmin

La classe `TariffaAdmin` gestisce la vista relativa alle tariffe che verrà visualizzata dall'amministratore ed è situata nel file `admin.py`. Gli elementi della classe sono mostrati in Tabella 3.16. Le associazioni della classe sono mostrate in Tabella 3.17.

Elemento	Nome	Tipo
Attributo	change_list_template	string
Attributo	list_display	list
Attributo	search_field	list
Funzione	changelist_view()	HttpResponse

Tabella 3.16: Tariffa: elementi della classe `TariffaAdmin`

Associazione	Nome
<code>TariffaAdmin</code> - Tariffa	0..1 - 0..*
<code>TariffaAdmin</code> - <code>TariffaConfig</code>	0-1 - 1

Tabella 3.17: Tariffa: associazioni della classe `TariffaAdmin`

3.4.4 TariffaConfig

La classe `TariffaConfig` contiene gli elementi di configurazione dell'intero package, senza di essa il funzionamento di tutte le altre classi verrebbe meno. Gli elementi sono mostrati in Tabella 3.18 mentre le associazioni della classe sono mostrate in Tabella 3.19.

Elemento	Nome	Tipo
Attributo	default_auto_field	string
Attributo	name	string

Tabella 3.18: Tariffa: elementi della classe `TariffaConfig`

Associazione	Nome
TariffaConfig - TariffaAdmin	1 - 0..1

Tabella 3.19: Tariffa: associazioni della classe TariffaConfig

3.5 Fattura

Il package `Fattura` gestisce l’inserimento, la modifica e l’eliminazione di una fattura da parte dell’amministratore. A partire da una fattura, inoltre, è possibile generare un file PDF con gli elementi inseriti. Il package con le relative classi è mostrato in Figura 3.5.

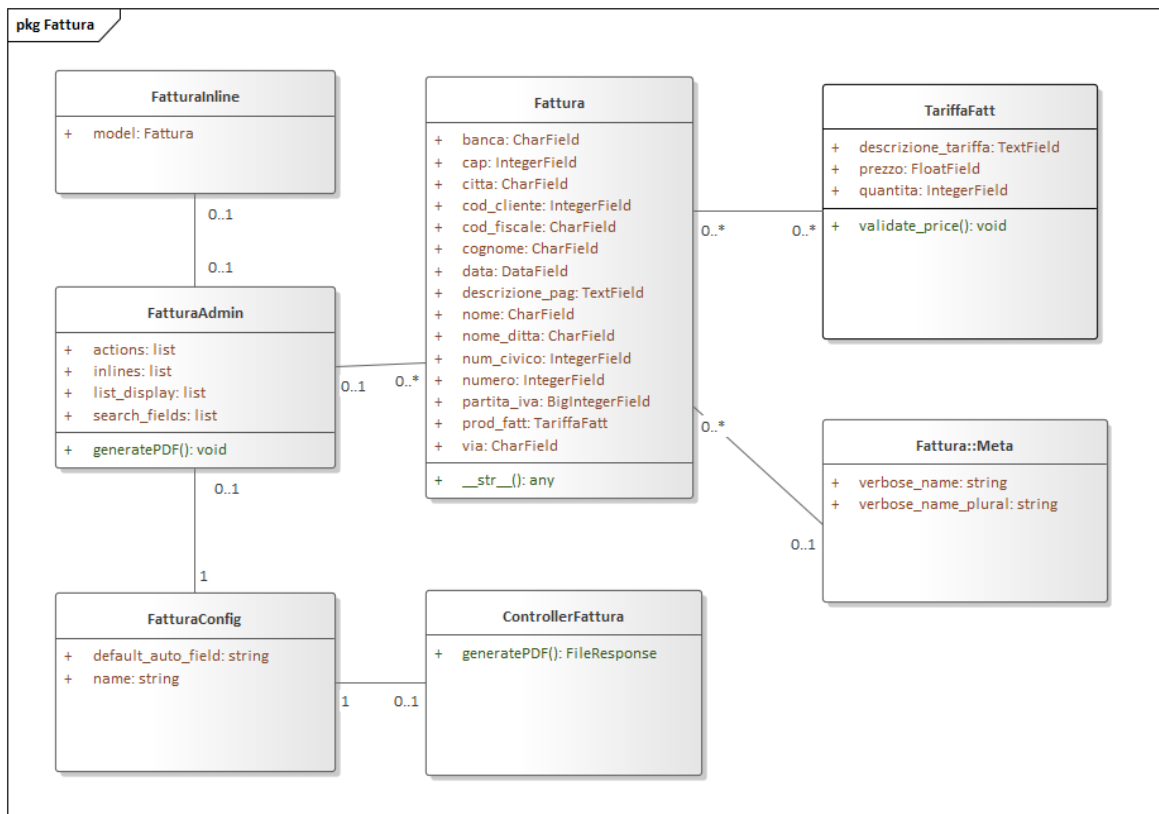


Figura 3.5: Package `Fattura` e relative classi

3.5.1 Fattura

La classe `Fattura` costituisce il modello da cui l’amministratore potrà gestire le fatture all’interno del software ed è situata nel file `models.py`. La fattura contiene come chiave esterna `TariffaFatt`, che descrive le tariffe che verranno utilizzate. Gli elementi che la costituiscono sono mostrati nella Tabella 3.20

Elemento	Nome	Tipo
Attributo	banca	CharField
Attributo	cap	IntegerField
Attributo	citta	CharField
Attributo	cod_cliente	IntegerField
Attributo	cod_fiscale	CharField
Attributo	cognome	CharField
Attributo	data	DataField
Attributo	descrizione_auto_field	TextField
Attributo	nome	CharField
Attributo	nome_ditta	CharField
Attributo	num_civico	IntegerField
Attributo	numero	IntegerField
Attributo	patita_iva	BigIntegerField
Attributo	prod_fatt	TariffaFatt
Attributo	via	CharField
Funzione	__str__	any

Tabella 3.20: Fattura: elementi della classe Fattura

Le associazioni della classe sono mostrate in Tabella 3.21.

Associazione	Nome
Fattura - TariffaFatt	0..* - 0..*
Fattura - FatturaAdmin	0..* - 0..1
Fattura - Meta	0..* - 0..1

Tabella 3.21: Fattura: associazioni della classe Fattura

3.5.2 Meta

La classe *Meta* consente di modificare alcuni comportamenti della classe *Fattura* e, per questo motivo, è definita all'interno di quest'ultima. Gli elementi della classe sono mostrati in Tabella 3.22 mentre le associazioni della classe sono mostrate in Tabella 3.23.

Elemento	Nome	Tipo
Attributo	verbose_name	string
Attributo	verbose_name_plural	string

Tabella 3.22: Fattura: elementi della classe Meta

Associazione	Nome
Meta - Fattura	0..1 - 0..*

Tabella 3.23: Fattura: associazioni della classe Meta

3.5.3 TariffaFatt

La classe `TariffaFatt` descrive il modello delle tariffe che dovranno essere inserite nella fattura. Gli elementi della classe sono mostrati in Tabella 3.24 mentre le associazioni della classe sono mostrate in Tabella 3.25.

Elemento	Nome	Tipo
Attributo	<code>descrizione_tariffa</code>	<code>TextField</code>
Attributo	<code>prezzo</code>	<code>FloatField</code>
Attributo	<code>quantita</code>	<code>IntegerField</code>
Funzione	<code>validate_price()</code>	<code>void</code>

Tabella 3.24: Fattura: elementi della classe `TariffaFatt`

Associazione	Nome
<code>Fattura - TariffaFatt</code>	<code>0..* - 0..*</code>

Tabella 3.25: Fattura: associazioni della classe `TariffaFatt`

3.5.4 FatturaAdmin

La classe `FatturaAdmin` gestisce la vista dell'amministratore relativa alle fatture ed è situata nel file `admin.py`. Gli elementi della classe sono mostrati in Tabella 3.26. Le associazioni della classe sono mostrate in Tabella 3.27.

Elemento	Nome	Tipo
Attributo	<code>actions</code>	<code>list</code>
Attributo	<code>inlines</code>	<code>list</code>
Attributo	<code>list_display</code>	<code>list</code>
Attributo	<code>search_fields</code>	<code>list</code>
Funzione	<code>generatePDFchoice()</code>	<code>void</code>

Tabella 3.26: Fattura: elementi della classe `FatturaAdmin`

Associazione	Nome
<code>FatturaAdmin - FatturaInline</code>	<code>0..1 - 0..1</code>
<code>FatturaAdmin - Fattura</code>	<code>0..1 - 0..*</code>
<code>FatturaConfig - FatturaAdmin</code>	<code>1 - 0..1</code>

Tabella 3.27: Fattura: associazioni della classe `FatturaConfig`

3.5.5 FatturaInline

La classe `FatturaInline` consente di inserire le varie tariffe all'interno della fattura al momento di creazione di quest'ultima. Gli elementi che costituiscono la classe sono mostrati in Tabella 3.28 mentre le associazioni della classe sono mostrate in Tabella 3.29.

Elemento	Nome	Tipo
Attributo	model	Fattura

Tabella 3.28: Fattura: elementi della classe `FatturaInline`

Associazione	Nome
<code>FatturaInline - FatturaAdmin</code>	0..1 - 0..1

Tabella 3.29: Fattura: associazioni della classe `FatturaInline`

3.5.6 ControllerFattura

La classe `ControllerFattura` permette di generare un file PDF a partire dalle informazioni di una tupla relativa alle fatture. Gli elementi sono mostrati in Tabella 3.30 mentre le associazioni della classe sono mostrate in Tabella 3.31.

Elemento	Nome	Tipo
Funzione	<code>generatePDF ()</code>	<code>FileResponse</code>

Tabella 3.30: Fattura: elementi della classe `ControllerFattura`

Associazione	Nome
<code>ControllerFattura - FatturaConfig</code>	0..1 - 1

Tabella 3.31: Fattura: associazioni della classe `ControllerFattura`

3.5.7 FatturaConfig

La classe `FatturaConfig` contiene gli elementi di configurazione dell'intero package, senza di essa il funzionamento di tutte le altre classi verrebbe meno. Gli elementi sono mostrati in Tabella 3.32. Le associazioni della classe sono mostrate in Tabella 3.33.

Elemento	Nome	Tipo
Attributo	<code>default_auto_field</code>	<code>string</code>
Attributo	<code>name</code>	<code>string</code>

Tabella 3.32: Fattura: elementi della classe `FatturaConfig`

Associazione	Nome
<code>FatturaConfig - ControllerFattura</code>	1 - 0..*
<code>FatturaConfig - FatturaAdmin</code>	1 - 0..*

Tabella 3.33: Fattura: associazioni della classe `FatturaConfig`

3.6 Prodotto

Il package `Prodotto` gestisce l’inserimento, la modifica e l’eliminazione di un prodotto che poi verrà visualizzato dall’amministratore. Le classi che lo costituiscono sono mostrate in Figura 3.6.

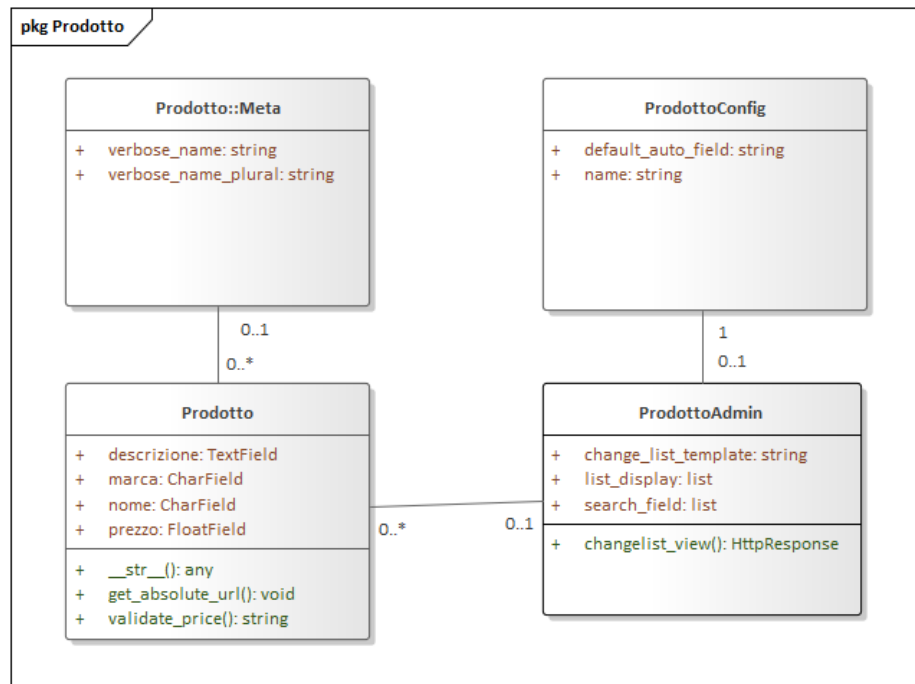


Figura 3.6: Package `Prodotto` e relative classi

3.6.1 Prodotto

La classe `Prodotto` definisce il modello utilizzato per l’inserimento di un prodotto da parte dell’amministratore ed è situata nel file `models.py`. Gli elementi della classe sono mostrati in Tabella 3.34 mentre le associazioni della classe sono mostrate in Tabella 3.35.

Elemento	Nome	Tipo
Attributo	descrizione	TextField
Attributo	marca	CharField
Attributo	nome	CharField
Attributo	prezzo	FloatField
Funzione	<code>__str__()</code>	any
Funzione	<code>validate_price()</code>	void
Funzione	<code>get_absolute_url()</code>	string

Tabella 3.34: `Prodotto`: elementi della classe `Prodotto`

Associazione	Nome
Prodotto - ProdottoAdmin	0..* - 0..1
Prodotto - Meta	0..* - 0..1

Tabella 3.35: Prodotto: associazioni della classe Prodotto

3.6.2 Meta

La classe `Meta` consente di modificare alcuni comportamenti della classe `Prodotto` e, per questo motivo, è definita all'interno di quest'ultima. Gli elementi della classe sono mostrati in Tabella 3.36 mentre le associazioni della classe sono mostrate in Tabella 3.37.

Elemento	Nome	Tipo
Attributo	<code>verbose_name</code>	<code>string</code>
Attributo	<code>verbose_name_plural</code>	<code>string</code>

Tabella 3.36: Prodotto: elementi della classe Meta

Associazione	Nome
Meta - Prodotto	0..1 - 0..*

Tabella 3.37: Prodotto: associazioni della classe Meta

3.6.3 ProdottoAdmin

La classe `ProdottoAdmin` gestisce la vista dell'amministratore relativa ai prodotti ed è situata nella classe `admin.py`. Gli elementi della classe sono mostrati in Tabella 3.38 mentre le associazioni della classe sono mostrate in Tabella 3.39.

Elemento	Nome	Tipo
Attributo	<code>change_list_template</code>	<code>string</code>
Attributo	<code>list_display</code>	<code>list</code>
Attributo	<code>search_field</code>	<code>list</code>
Funzione	<code>changelist_view()</code>	<code>HttpResponse</code>

Tabella 3.38: Prodotto: elementi della classe ProdottoAdmin

Associazione	Nome
ProdottoAdmin - Prodotto	0..1 - 0..*
ProdottoAdmin - ProdottoConfig	0..1 - 1

Tabella 3.39: Prodotto: elementi della classe ProdottoAdmin

3.6.4 ProdottoConfig

La classe `ProdottoConfig` contiene gli elementi di configurazione dell'intero package; senza di essa il funzionamento di tutte le altre classi verrebbe meno. Gli elementi sono mostrati in Tabella 3.40. Le associazioni della classe sono mostrate in Tabella 3.41.

Elemento	Nome	Tipo
Attributo	<code>default_auto_field</code>	<code>string</code>
Attributo	<code>name</code>	<code>string</code>

Tabella 3.40: Prodotto: elementi della classe `ProdottoConfig`

Associazione	Nome
<code>ProdottoConfig - Prodotto</code>	<code>0..1 - 0..*</code>

Tabella 3.41: Prodotto: associazioni della classe `ProdottoConfig`

3.7 Prenotazione

Il package `Prenotazione` gestisce la prenotazione effettuabile da un potenziale cliente. Le classi che lo costituiscono sono mostrate in Figura 3.7.

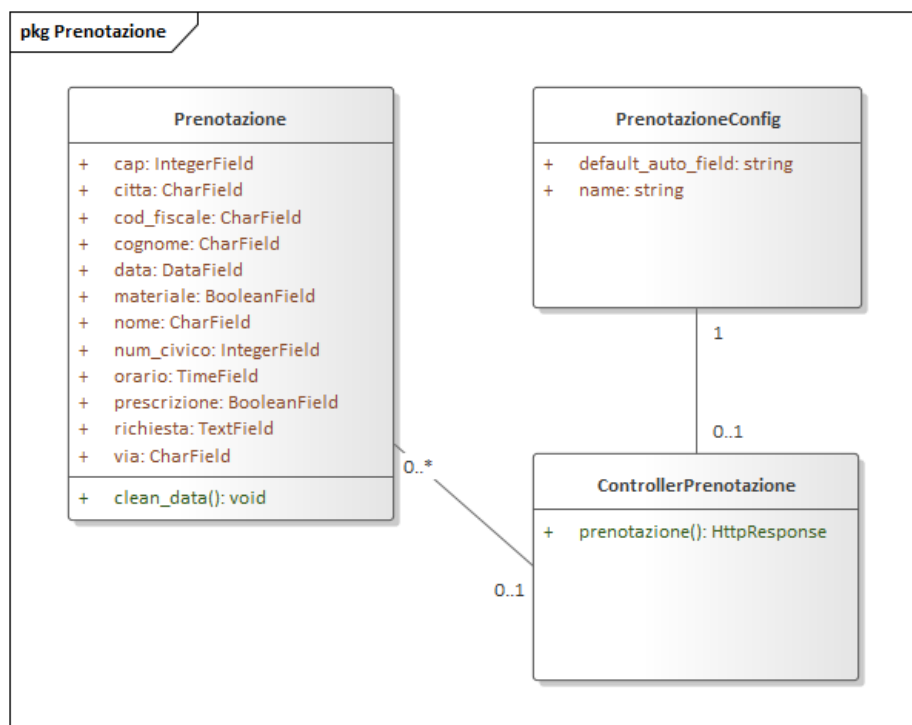


Figura 3.7: Package `Prenotazione` e relative classi

3.7.1 Prenotazione

La classe `Prenotazione` costituisce il modello su cui è basata la form di prenotazione dell'utente ed è situata in `models.py`. Gli elementi che costituiscono la classe sono mostrati in Tabella 3.42 mentre le associazioni della classe sono mostrate in Tabella 3.43.

Elemento	Nome	Tipo
Attributo	<code>cap</code>	<code>IntegerField</code>
Attributo	<code>citta</code>	<code>CharField</code>
Attributo	<code>cod_fiscale</code>	<code>CharField</code>
Attributo	<code>cognome</code>	<code>CharField</code>
Attributo	<code>data</code>	<code>DataField</code>
Attributo	<code>materiale</code>	<code>BooleanField</code>
Attributo	<code>nome</code>	<code>CharField</code>
Attributo	<code>num_civico</code>	<code>IntegerField</code>
Attributo	<code>orario</code>	<code>TimeField</code>
Attributo	<code>prescrizione</code>	<code>BooleanField</code>
Attributo	<code>richiesta</code>	<code>TextField</code>
Attributo	<code>via</code>	<code>CharField</code>
Funzione	<code>clean_data()</code>	<code>void</code>

Tabella 3.42: Prenotazione: elementi della classe `Prenotazione`

Associazione	Nome
<code>Prenotazione</code> <code>ControllerPrenotazione</code>	- 0..* - 0..1

Tabella 3.43: Prenotazione: associazioni della classe `Prenotazione`

3.7.2 ControllerPrenotazione

La classe `ControllerPrenotazione` si occupa di gestire la vista lato utente che effettuerà la prenotazione ed è situata nel file `views.py`. L'elemento della classe è mostrato in Tabella 3.44 mentre le associazioni della classe sono mostrate in Tabella 3.45.

Elemento	Nome	Tipo
Funzione	<code>prenotazione()</code>	<code>HttpResponse</code>

Tabella 3.44: Prenotazione: elementi della classe `ControllerPrenotazione`

3.7.3 PrenotazioneConfig

La classe `PrenotazioneConfig` contiene gli elementi di configurazione dell'intero package; senza di essa il funzionamento di tutte le altre classi verrebbe meno. Gli elementi sono mostrati in Tabella 3.46 mentre le associazioni della classe sono mostrate in Tabella 3.47.

Associazione	Nome
ControllerPrenotazione Prenotazione	- 0..1 - 0..*
ControllerPrenotazione PrenotazioneConfig	- 0..1 - 1

Tabella 3.45: Prenotazione: associazioni della classe ControllerPrenotazione

Elemento	Nome	Tipo
Attributo	default_auto_field	string
Attributo	name	string

Tabella 3.46: Prenotazione: elementi della classe PrenotazioneConfig

Associazione	Nome
PrenotazioneConfig ControllerPrenotazione	- 1 - 0..1

Tabella 3.47: Prenotazione: associazioni della classe PrenotazioneConfig

3.8 FAQ

Infine il package FAQ gestisce l’inserimento, la modifica e l’eliminazione delle FAQ che verranno poi visualizzate dai clienti. Il package comprende le classi mostrate in Figura 3.8.

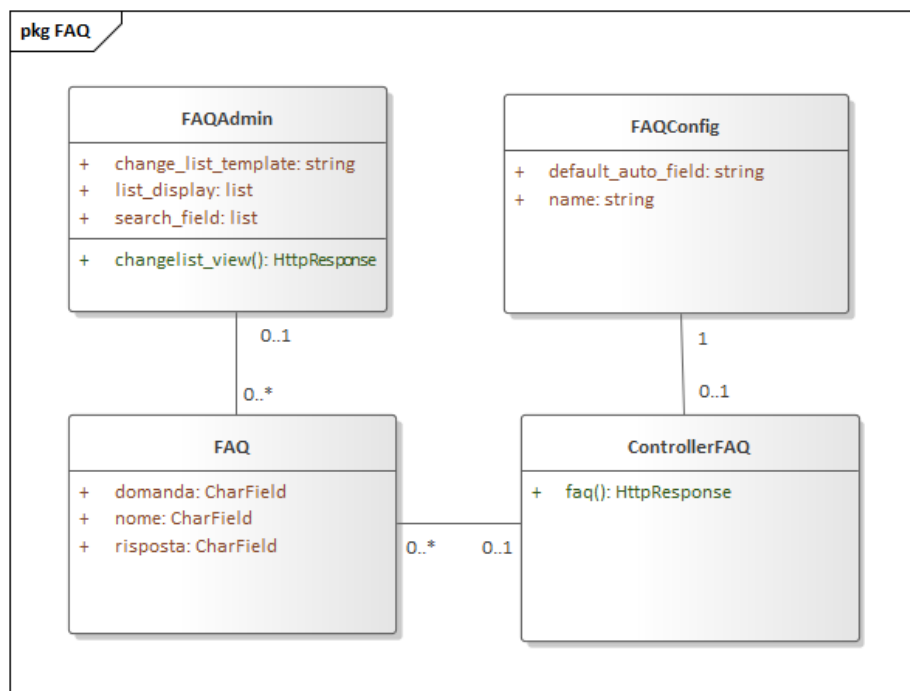


Figura 3.8: Package FAQ e relative classi

3.8.1 FAQ

La classe `FAQ` costituisce il modello che l'amministratore dovrà seguire nella creazione di FAQ ed è situato nella classe `models.py`. Gli elementi che lo costituiscono sono descritti in Tabella 3.48. Le associazioni della classe sono mostrate in Tabella 3.49.

Elemento	Nome	Tipo
Attributo	nome	CharField
Attributo	domanda	CharField
Attributo	risposta	CharField

Tabella 3.48: FAQ: elementi della classe FAQ

Associazione	Nome
FAQ - ControllerFAQ	0..* - 0..1
FAQ - FAQAdmin	0..* - 0..1

Tabella 3.49: FAQ: associazioni della classe FAQ

3.8.2 FAQAdmin

La classe `FAQAdmin` gestisce la vista dell'amministratore relativa ai prodotti ed è situato nella classe `admin.py`. Gli elementi della classe sono mostrati in Tabella 3.50 mentre le associazioni della classe sono mostrate in Tabella 3.51.

Elemento	Nome	Tipo
Attributo	change_list_template	string
Attributo	list_display	list
Attributo	search_field	list

Tabella 3.50: FAQ: elementi della classe FAQAdmin

Associazione	Nome
FAQAdmin - FAQ	0..1 - 0..*

Tabella 3.51: FAQ: associazioni della classe FAQAdmin

3.8.3 ControllerFAQ

`ControllerFAQ` è la classe che gestisce la vista lato utente delle FAQ ed è situata nel file `views.py`. L'unico elemento della classe è descritto in Tabella 3.52 mentre le associazioni della classe sono mostrate in Tabella 3.53.

Elemento	Nome	Tipo
Funzione	faq()	HttpResponse

Tabella 3.52: FAQ: elementi della classe ControllerFAQ

Associazione	Nome
ControllerFAQ - FAQConfig	0..1 - 1
ControllerFAQ - FAQ	0..1 - 1..*

Tabella 3.53: FAQ: associazioni della classe ControllerFaq

3.8.4 FAQConfig

La classe `FAQConfig` contiene gli elementi di configurazione dell'intero package; senza di essa il funzionamento di tutte le altre classi verrebbe meno. Gli elementi sono mostrati in Tabella 3.54 mentre le associazioni della classe sono mostrate in Tabella 3.55.

Elemento	Nome	Tipo
Attributo	default_auto_field	string
Attributo	name	string

Tabella 3.54: FAQ: elementi della classe FaqConfig

Associazione	Nome
FAQConfig - ControllerFAQ	1 - 0..1

Tabella 3.55: FAQ: associazioni della classe FAQConfig

Diagrammi del comportamento

In questo capitolo il nostro obiettivo sarà quello di individuare le interazioni che sussistono tra utente e sistema. Come modelli di interazione verranno utilizzati i diagrammi dei casi d'uso, i diagrammi di sequenza e i diagrammi di attività. I diagrammi rappresentano l'interazione a livelli di dettaglio differenti; questo ci consente di avere una visione più ampia del sistema. Infine la matrice di mapping ci permette di verificare che tutti i requisiti funzionali, analizzati nel Capitolo 2, vengano rispettati.

4.1 Diagramma dei casi d'uso

Un caso d'uso può essere considerato come una semplice descrizione di ciò che un utente si aspetta in tale interazione. In questa prima sezione analizzeremo i casi d'uso relativi al nostro software, come mostrato in Figura 4.1.

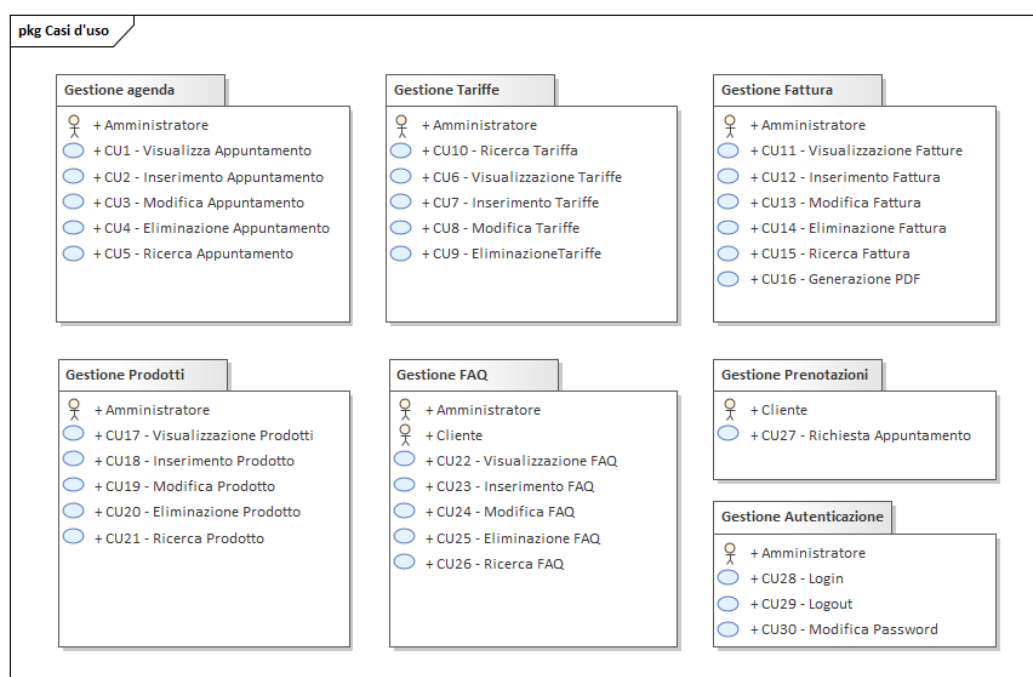


Figura 4.1: Elenco dei casi d'uso relativi al sistema

Nel seguito esamineremo i vari casi d'uso specificati nell'elenco. In particolare, dedicheremo una sottosezione per ciascun raggruppamento mostrato in Figura 4.1.

4.1.1 Gestione dell'agenda

Questa prima sottosezione ha lo scopo di illustrare i casi d'uso riguardanti la gestione dell'agenda, come mostrato in Figura 4.2.

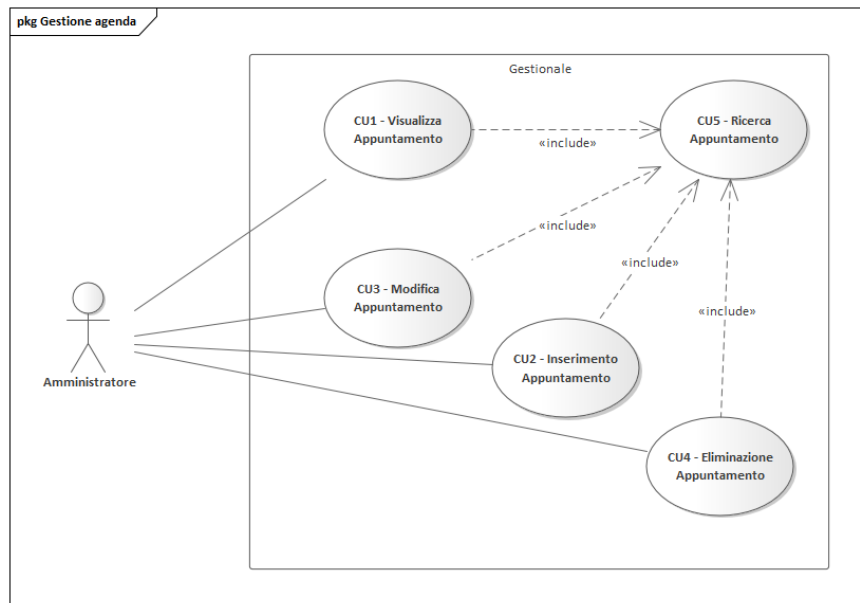


Figura 4.2: Casi d'uso relativi alla gestione dell'agenda

Visualizzazione di un appuntamento

Questo caso d'uso si verifica qualora l'amministratore voglia visualizzare un appuntamento all'interno dell'agenda. Le informazioni relative sono mostrate nella Tabella 4.1.

Elementi	Descrizione
Pre-Condizioni	L'appuntamento esiste nel sistema
Post-Condizioni	Nessuna
Sequenza degli eventi principali	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'amministratore del sistema vuole visualizzare le informazioni relative ad appuntamento 2. Il sistema recupera le informazioni dell'appuntamento cercato dall'amministratore 3. Il sistema visualizza nello schermo le informazioni relative all'appuntamento

Elementi	Descrizione
Sequenza degli eventi alternativa	Nessuna

Tabella 4.1: Caso d'uso relativo alla visualizzazione di un appuntamento

Inserimento di un appuntamento

Questo caso d'uso si verifica quando l'amministratore vuole inserire un appuntamento all'interno dell'agenda. Le informazioni relative sono mostrate nella Tabella 4.2.

Elementi	Descrizione
Pre-Condizioni	L'appuntamento non esiste nel sistema
Post-Condizioni	Esiste un nuovo appuntamento
Sequenza degli eventi principali	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'amministratore del sistema vuole inserire un nuovo appuntamento nella propria agenda 2. L'amministratore inserisce le informazioni relative al nuovo appuntamento 3. Il sistema registra con successo il nuovo appuntamento
Sequenza degli eventi alternativa	<ol style="list-style-type: none"> 2. L'amministratore inserisce informazioni errate 3. Il sistema mostra gli errori generati nello schermo

Tabella 4.2: Caso d'uso relativo all'inserimento di un appuntamento

Modifica di un appuntamento

Questo caso d'uso si verifica quando l'amministratore vuole modificare un appuntamento all'interno dell'agenda. Le informazioni relative sono mostrate nella Tabella 4.3.

Elementi	Descrizione
Pre-Condizioni	L'appuntamento esiste nel sistema
Post-Condizioni	L'appuntamento viene modificato

Elementi	Descrizione
Sequenza degli eventi principali	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'amministratore del sistema vuole modificare un appuntamento nella propria agenda 2. Il sistema recupera le informazioni dell'appuntamento che l'amministratore ha selezionato 3. L'amministratore aggiorna le informazioni relative all'appuntamento 4. Il sistema registra con successo l'appuntamento aggiornato
Sequenza degli eventi alternativa	<ol style="list-style-type: none"> 3. L'amministratore inserisce informazioni errate 4. Il sistema mostra gli errori generati nello schermo

Tabella 4.3: Caso d'uso relativo alla modifica di un appuntamento

Eliminazione di un appuntamento

Questo caso d'uso si verifica quando l'amministratore vuole eliminare un appuntamento all'interno dell'agenda. Le informazioni relative sono mostrate nella Tabella 4.4.

Elementi	Descrizione
Pre-Condizioni	L'appuntamento esiste nel sistema
Post-Condizioni	L'appuntamento non esiste più nel sistema
Sequenza degli eventi principali	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'amministratore del sistema vuole eliminare un appuntamento dalla propria agenda 2. Il sistema recupera le informazioni relative all'appuntamento che l'amministratore ha selezionato 3. L'amministratore elimina l'appuntamento 4. L'appuntamento viene eliminato dal sistema
Sequenza degli eventi alternativa	Nessuna

Tabella 4.4: Caso d'uso relativo all'eliminazione di un appuntamento

Ricerca di un appuntamento

Questo caso d'uso si verifica quando l'amministratore vuole cercare un appuntamento all'interno dell'agenda. Le informazioni relative sono mostrate nella Tabella 4.5.

Elementi	Descrizione
Pre-Condizioni	Nessuna
Post-Condizioni	Nessuna
Sequenza degli eventi principali	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'amministratore del sistema vuole cercare un appuntamento nella propria agenda 2. L'amministratore inserisce i dati chiave per la ricerca 3. Il sistema verifica che l'appuntamento sia presente 4. Il sistema mostra l'appuntamento cercato
Sequenza degli eventi alternativa	4. Il sistema non restituisce nulla

Tabella 4.5: Caso d'uso relativo alla ricerca di un appuntamento

4.1.2 Gestione delle tariffe

Questa sottosezione ha lo scopo di illustrare i casi d'uso riguardanti la gestione del tariffario, come mostrato in Figura 4.3.

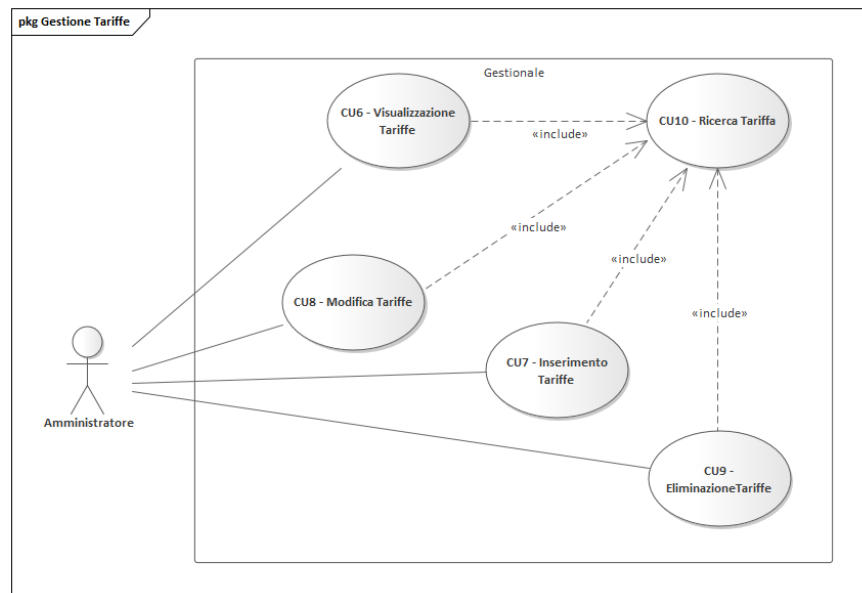


Figura 4.3: Casi d'uso relativi alla gestione del tariffario

Visualizzazione del tariffario

Questo caso d'uso si verifica quando l'amministratore vuole visualizzare una tariffa presente all'interno del tariffario. Le informazioni relative sono mostrate nella Tabella 4.6.

Elementi	Descrizione
Pre-Condizioni	La tariffa esiste nel sistema
Post-Condizioni	Nessuna
Sequenza degli eventi principali	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'amministratore del sistema vuole visualizzare le informazioni relative ad una tariffa 2. Il sistema recupera le informazioni della tariffa cercata dall'amministratore 3. Il sistema visualizza nello schermo le informazioni della tariffa
Sequenza degli eventi alternativa	Nessuna

Tabella 4.6: Caso d'uso relativo alla visualizzazione di una tariffa

Inserimento di una tariffa

Questo caso d'uso si verifica quando l'amministratore vuole inserire una tariffa all'interno del tariffario. Le informazioni relative sono mostrate nella Tabella 4.7.

Elementi	Descrizione
Pre-Condizioni	La tariffa non esiste nel sistema
Post-Condizioni	Esiste una nuova tariffa
Sequenza degli eventi principali	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'amministratore del sistema vuole inserire una nuova tariffa nel proprio tariffario 2. L'amministratore inserisce le informazioni relative alla nuova tariffa 3. Il sistema registra con successo la nuova tariffa
Sequenza degli eventi alternativa	<ol style="list-style-type: none"> 2. L'amministratore inserisce informazioni errate 3. Il sistema mostra gli errori generati nello schermo

Tabella 4.7: Caso d'uso relativo all'inserimento di una tariffa

Modifica della tariffa

Questo caso d'uso si verifica quando l'amministratore vuole modificare una tariffa all'interno del tariffario. Le informazioni relative sono mostrate nella Tabella 4.8.

Elementi	Descrizione
Pre-Condizioni	La tariffa esiste nel sistema

Elementi	Descrizione
Post-Condizioni	La tariffa viene modificata
Sequenza degli eventi principali	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'amministratore del sistema vuole modificare una tariffa nel proprio tariffario 2. Il sistema recupera le informazioni della tariffa che l'amministratore ha selezionato 3. L'amministratore aggiorna le informazioni relative alla tariffa 4. Il sistema registra con successo la tariffa aggiornata
Sequenza degli eventi alternativa	<ol style="list-style-type: none"> 3. L'amministratore inserisce informazioni errate 4. Il sistema mostra gli errori generati nello schermo

Tabella 4.8: Caso d'uso relativo alla modifica di una tariffa

Eliminazione di una tariffa

Questo caso d'uso si verifica quando l'amministratore vuole eliminare una tariffa all'interno del tariffario. Le informazioni relative sono mostrate nella Tabella 4.9.

Elementi	Descrizione
Pre-Condizioni	La tariffa esiste nel sistema
Post-Condizioni	La tariffa non esiste più nel sistema
Sequenza degli eventi principali	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'amministratore del sistema vuole eliminare una tariffa dal proprio tariffario 2. Il sistema recupera le informazioni della tariffa che l'amministratore ha selezionato 3. L'amministratore elimina la tariffa 4. La tariffa viene eliminata dal sistema
Sequenza degli eventi alternativa	Nessuna

Tabella 4.9: Caso d'uso relativo all'eliminazione di una tariffa

Ricerca di una tariffa

Questo caso d'uso si verifica quando l'amministratore vuole cercare una tariffa all'interno del tariffario. Le informazioni relative sono mostrate nella Tabella 4.10.

Elementi	Descrizione
Pre-Condizioni	Nessuna
Post-Condizioni	Nessuna
Sequenza degli eventi principali	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'amministratore del sistema vuole cercare una tariffa nella tariffario 2. L'amministratore inserisce i dati chiave per la ricerca 3. Il sistema verifica che la tariffa sia presente 4. Il sistema mostra la tariffa cercata
Sequenza degli eventi alternativa	4. Il sistema non restituisce nulla

Tabella 4.10: Caso d'uso relativo alla ricerca di una tariffa

4.1.3 Gestione delle fatture

Questa sottosezione ha lo scopo di illustrare i casi d'uso riguardanti la gestione delle fatture, come mostrato in Figura 4.4.

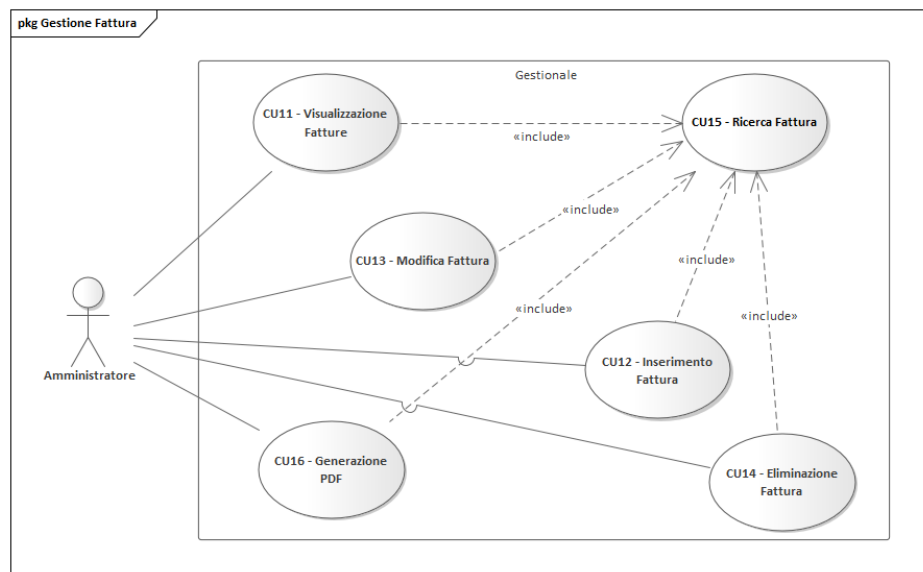


Figura 4.4: Casi d'uso relativi alla gestione delle fatture

Visualizzazione di una fattura

Questo caso d'uso si verifica quando l'amministratore vuole visualizzare una fattura. Le informazioni relative sono mostrate nella Tabella 4.11.

Elementi	Descrizione
Pre-Condizioni	La fattura esiste nel sistema

Elementi	Descrizione
Post-Condizioni	Nessuna
Sequenza degli eventi principali	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'amministratore del sistema vuole visualizzare le informazioni relative ad una fattura 2. Il sistema recupera le informazioni della fattura cercata dall'amministratore 3. Il sistema visualizza nello schermo le informazioni relative alla fattura
Sequenza degli eventi alternativa	nessuna

Tabella 4.11: Caso d'uso relativo alla visualizzazione di una fattura

Inserimento di una fattura

Questo caso d'uso si verifica quando l'amministratore vuole inserire una fattura. Le informazioni relative sono mostrate nella Tabella 4.12.

Elementi	Descrizione
Pre-Condizioni	La fattura non esiste nel sistema
Post-Condizioni	Esiste una nuova fattura
Sequenza degli eventi principali	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'amministratore del sistema vuole inserire una nuova fattura 2. L'amministratore inserisce le informazioni relative alla nuova fattura 3. Il sistema registra con successo la nuova fattura
Sequenza degli eventi alternativa	<ol style="list-style-type: none"> 2. L'amministratore inserisce informazioni errate 3. Il sistema mostra gli errori generati nello schermo

Tabella 4.12: Caso d'uso relativo all'inserimento di una fattura

Modifica di una fattura

Questo caso d'uso si verifica quando l'amministratore vuole modificare una fattura. Le informazioni relative sono mostrate nella Tabella 4.13.

Elementi	Descrizione
Pre-Condizioni	La fattura esiste nel sistema

Elementi	Descrizione
Post-Condizioni	La fattura viene modificata
Sequenza degli eventi principali	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'amministratore del sistema vuole modificare una fattura 2. Il sistema recupera le informazioni della fattura che l'amministratore ha selezionato 3. L'amministratore aggiorna le informazioni relative alla fattura 4. Il sistema registra con successo la fattura aggiornata
Sequenza degli eventi alternativa	<ol style="list-style-type: none"> 3. L'amministratore inserisce informazioni errate 4. Il sistema mostra gli errori generati nello schermo

Tabella 4.13: Caso d'uso relativo alla modifica di una fattura

Eliminazione di una fattura

Questo caso d'uso si verifica quando l'amministratore vuole eliminare una fattura. Le informazioni relative sono mostrate nella Tabella 4.14.

Elementi	Descrizione
Pre-Condizioni	La fattura esiste nel sistema
Post-Condizioni	La fattura non esiste più nel sistema
Sequenza degli eventi principali	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'amministratore del sistema vuole eliminare una fattura 2. Il sistema recupera le informazioni della fattura che l'amministratore ha selezionato 3. L'amministratore elimina la fattura 4. La fattura viene eliminata dal sistema
Sequenza degli eventi alternativa	Nessuna

Tabella 4.14: Caso d'uso relativo all'eliminazione di una fattura

Generazione del PDF

Questo caso d'uso si verifica quando l'amministratore vuole generare un file PDF a partire da una fattura. Le informazioni relative sono mostrate nella Tabella 4.15.

Elementi	Descrizione
Pre-Condizioni	La fattura esiste nel sistema
Post-Condizioni	Viene generato un file PDF con i dati della fattura
Sequenza degli eventi principali	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'amministratore del sistema vuole generare il PDF di una fattura 2. Il sistema recupera le informazioni della fattura che l'amministratore ha selezionato 3. L'amministratore genera il file PDF della fattura 4. Il PDF viene salvato in locale dal sistema
Sequenza degli eventi alternativa	Nessuna

Tabella 4.15: Caso d'uso relativo alla generazione di un file PDF relativo ad una fattura

Ricerca di una fattura

Questo caso d'uso si verifica quando l'amministratore vuole ricercare una fattura. Le informazioni relative sono mostrate nella Tabella 4.16.

Elementi	Descrizione
Pre-Condizioni	Nessuna
Post-Condizioni	Nessuna
Sequenza degli eventi principali	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'amministratore del sistema vuole ricercare una fattura 2. L'amministratore inserisce i dati chiave per la ricerca 3. Il sistema verifica che la fattura sia presente 4. Il sistema mostra la fattura cercata
Sequenza degli eventi alternativa	4. Il sistema non restituisce nulla

Tabella 4.16: Caso d'uso relativo alla ricerca di una fattura

4.1.4 Gestione dei prodotti

Questa sottosezione ha lo scopo di illustrare i casi d'uso riguardanti la gestione dell'inventario, come mostrato in Figura 4.5.

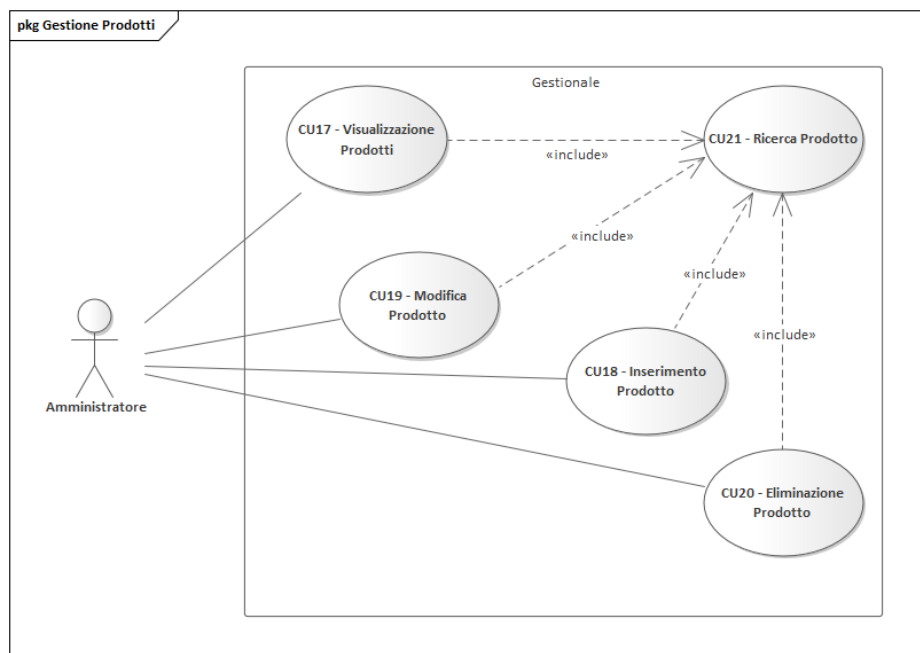


Figura 4.5: Casi d'uso relativi alla gestione dei prodotti

Visualizzazione dei prodotti

Questo caso d'uso si verifica qualora l'amministratore voglia visualizzare un prodotto all'interno dell'inventario. Le informazioni relative sono mostrate nella Tabella 4.17.

Elementi	Descrizione
Pre-Condizioni	Il prodotto esiste nel sistema
Post-Condizioni	Nessuna
Sequenza degli eventi principali	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'amministratore del sistema vuole visualizzare le informazioni di un prodotto 2. Il sistema recupera le informazioni del prodotto cercato dall'amministratore 3. Il sistema visualizza nello schermo le informazioni del prodotto
Sequenza degli eventi alternativa	nessuna

Tabella 4.17: Caso d'uso relativo alla visualizzazione di un prodotto

Inserimento di un prodotto

Questo caso d'uso si verifica quando l'amministratore vuole inserire un prodotto all'interno dell'inventario. Le informazioni relative sono mostrate nella Tabella 4.18.

Elementi	Descrizione
Pre-Condizioni	Il prodotto non esiste nel sistema
Post-Condizioni	Esiste un nuovo prodotto
Sequenza degli eventi principali	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'amministratore del sistema vuole inserire un nuovo prodotto nel proprio inventario 2. L'amministratore inserisce le informazioni relative al nuovo prodotto 3. Il sistema registra con successo il nuovo prodotto
Sequenza degli eventi alternativa	<ol style="list-style-type: none"> 2. L'amministratore inserisce informazioni errate 3. Il sistema mostra gli errori generati nello schermo

Tabella 4.18: Caso d'uso relativo all'inserimento di un prodotto

Modifica di un prodotto

Questo caso d'uso si verifica quando l'amministratore vuole modificare un prodotto all'interno dell'inventario. Le informazioni relative sono mostrate nella Tabella 4.19.

Elementi	Descrizione
Pre-Condizioni	Il prodotto esiste nel sistema
Post-Condizioni	Il prodotto viene modificato
Sequenza degli eventi principali	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'amministratore del sistema vuole modificare un prodotto nel proprio inventario 2. Il sistema recupera le informazioni sul prodotto che l'amministratore ha selezionato 3. L'amministratore aggiorna le informazioni relative al prodotto 4. Il sistema registra con successo il prodotto aggiornato
Sequenza degli eventi alternativa	<ol style="list-style-type: none"> 3. L'amministratore inserisce informazioni errate 4. Il sistema mostra gli errori generati nello schermo

Tabella 4.19: Caso d'uso relativo alla modifica di un prodotto

Eliminazione di un prodotto

Questo caso d'uso si verifica quando l'amministratore vuole eliminare un prodotto all'interno dell'inventario. Le informazioni relative sono mostrate nella Tabella 4.20.

Elementi	Descrizione
Pre-Condizioni	Il prodotto esiste nel sistema
Post-Condizioni	Il prodotto non esiste più nel sistema
Sequenza degli eventi principali	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'amministratore del sistema vuole eliminare un prodotto dal proprio inventario 2. Il sistema recupera le informazioni sul prodotto che l'amministratore ha selezionato 3. L'amministratore elimina il prodotto 4. Il prodotto viene eliminato dal sistema
Sequenza degli eventi alternativa	Nessuna

Tabella 4.20: Caso d'uso relativo all'eliminazione di un prodotto

Ricerca di un prodotto

Questo caso d'uso si verifica quando l'amministratore vuole cercare un prodotto. Le informazioni relative sono mostrate nella Tabella 4.21.

Elementi	Descrizione
Pre-Condizioni	Nessuna
Post-Condizioni	Nessuna
Sequenza degli eventi principali	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'amministratore del gestionale vuole cercare un prodotto nell'inventario 2. L'amministratore inserisce i dati chiave per la ricerca 3. Il sistema verifica che il prodotto sia presente 4. Il sistema mostra il prodotto cercato
Sequenza degli eventi alternativa	4. Il sistema non restituisce nulla

Tabella 4.21: Caso d'uso relativo alla ricerca di un prodotto

4.1.5 Gestione delle FAQ

Questa sottosezione ha lo scopo di illustrare i casi d'uso riguardanti la gestione delle FAQ, come mostrato in Figura 4.6.

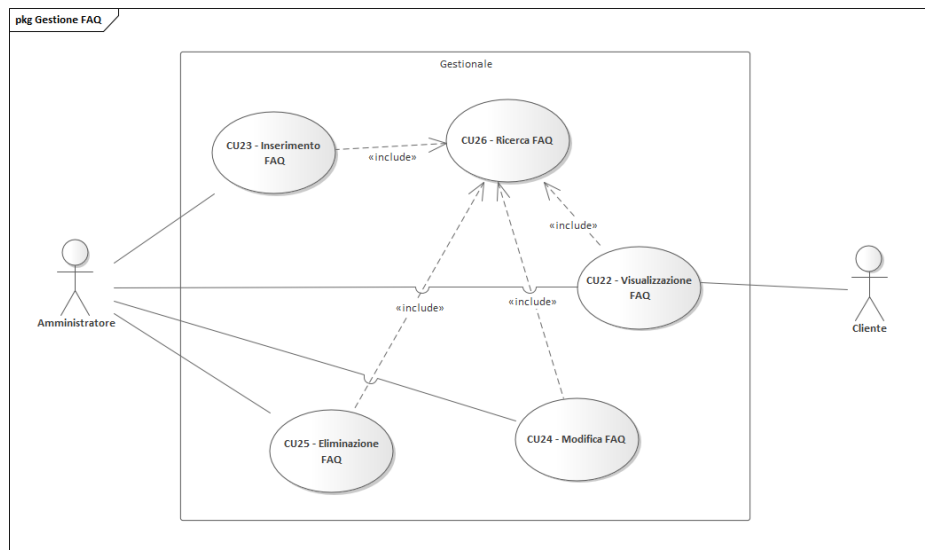


Figura 4.6: Casi d'uso relativi alla gestione delle FAQ

Visualizzazione di una FAQ

Questo caso d'uso si verifica qualora l'amministratore o un cliente vuole visualizzare una FAQ oppure. Le informazioni relative sono mostrate nella Tabella 4.22.

Elementi	Descrizione
Pre-Condizioni	La FAQ esiste nel sistema
Post-Condizioni	Nessuna
Sequenza degli eventi principali	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'amministratore del sistema o un cliente, vuole visualizzare le informazioni di una FAQ 2. Il sistema recupera le informazioni della FAQ cercata dall'amministratore, oppure conduce il cliente nella pagina delle FAQ 3. Il sistema visualizza nello schermo le informazioni delle FAQ
Sequenza degli eventi alternativa	Nessuna

Tabella 4.22: Caso d'uso relativo alla visualizzazione di una FAQ

Inserimento di una FAQ

Questo caso d'uso si verifica quando l'amministratore vuole inserire una FAQ nel sistema. Le informazioni relative sono mostrate nella Tabella 4.23.

Elementi	Descrizione
Pre-Condizioni	La FAQ non esiste nel sistema
Post-Condizioni	Esiste una nuova FAQ
Sequenza degli eventi principali	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'amministratore del sistema vuole inserire una nuova FAQ 2. L'amministratore inserisce le informazioni relative alla nuova FAQ 3. Il sistema registra con successo la nuova FAQ
Sequenza degli eventi alternativa	<ol style="list-style-type: none"> 2. L'amministratore inserisce informazioni errate 3. Il sistema mostra gli errori generati nello schermo

Tabella 4.23: Caso d'uso relativo all'inserimento di una FAQ

Modifica di una FAQ

Questo caso d'uso si verifica quando l'amministratore vuole modificare una FAQ nel sistema. Le informazioni relative sono mostrate nella Tabella 4.24.

Elementi	Descrizione
Pre-Condizioni	La FAQ esiste nel sistema
Post-Condizioni	La FAQ viene modificata
Sequenza degli eventi principali	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'amministratore del sistema vuole modificare una FAQ 2. Il sistema recupera le informazioni della FAQ che l'amministratore ha selezionato 3. L'amministratore aggiorna le informazioni relative alla FAQ 4. Il sistema registra con successo la FAQ aggiornata
Sequenza degli eventi alternativa	<ol style="list-style-type: none"> 3. L'amministratore inserisce informazioni errate 4. Il sistema mostra gli errori generati nello schermo

Tabella 4.24: Caso d'uso relativo alla modifica di una FAQ

Eliminazione di una FAQ

Questo caso d'uso si verifica quando l'amministratore vuole eliminare una FAQ. Le informazioni relative sono mostrate nella Tabella 4.25.

Elementi	Descrizione
Pre-Condizioni	La FAQ esiste nel sistema
Post-Condizioni	La FAQ non esiste più nel sistema
Sequenza degli eventi principali	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'amministratore del sistema vuole eliminare una FAQ 2. Il sistema recupera le informazioni della FAQ che l'amministratore ha selezionato 3. L'amministratore elimina la FAQ 4. La FAQ viene eliminata dal sistema
Sequenza degli eventi alternativa	Nessuna

Tabella 4.25: Caso d'uso relativo all'eliminazione di una FAQ

Ricerca di una FAQ

Questo caso d'uso si verifica quando l'amministratore vuole cercare una FAQ. Le informazioni relative sono mostrate nella Tabella 4.26.

Elementi	Descrizione
Pre-Condizioni	Nessuna
Post-Condizioni	Nessuna
Sequenza degli eventi principali	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'amministratore del sistema vuole cercare una FAQ 2. L'amministratore inserisce i dati chiave per la ricerca 3. Il sistema verifica che la FAQ sia presente 4. Il sistema mostra la FAQ cercata
Sequenza degli eventi alternativa	4. Il sistema non restituisce nulla

Tabella 4.26: Caso d'uso relativo alla ricerca di una FAQ

4.1.6 Gestione delle prenotazioni

Questa sottosezione ha lo scopo di illustrare i casi d'uso riguardanti la gestione delle prenotazioni, come mostrato in Figura 4.7.

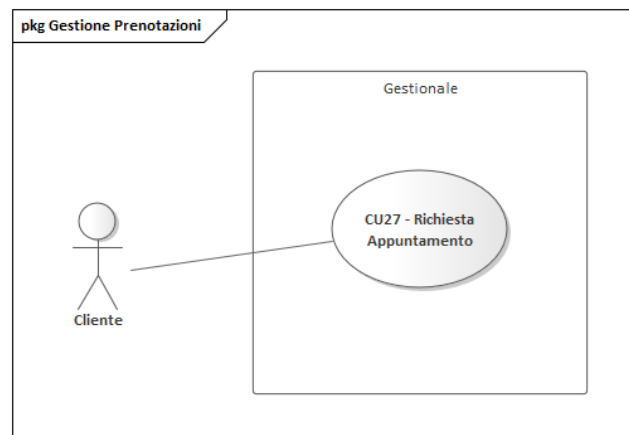


Figura 4.7: Caso d'uso relativo alla gestione delle prenotazioni

Richiesta di un appuntamento

Questo caso d'uso si verifica quando un cliente vuole richiedere un appuntamento all'amministratore. Le informazioni relative sono mostrate nella Tabella 4.27.

Elementi	Descrizione
Pre-Condizioni	La richiesta non è stata effettuata
Post-Condizioni	Viene inviata una mail all'amministratore con i dati inseriti
Sequenza degli eventi principali	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando un cliente vuole prenotare un appuntamento 2. Il cliente viene reindirizzato dal sistema alla pagina per la prenotazione 3. Il cliente inserisce i propri dati all'interno della form 4. Il sistema invia una mail all'amministratore e viene mostrato un messaggio di corretto invio
Sequenza degli eventi alternativa	<ol style="list-style-type: none"> 3. Il cliente inserisce informazioni errate nella form 4. Il sistema mostra gli errori generati nello schermo

Tabella 4.27: Caso d'uso relativo alla prenotazione di un appuntamento

4.1.7 Gestione dell'autenticazione

L'autenticazione non è stata trattata nel precedente capitolo poichè *Django* fornisce delle classi predefinite; tuttavia è importante analizzarla dal punto di vista dei casi d'uso.

Quest'ultima sottosezione ha lo scopo di mostrare proprio tali casi d'uso, come mostrato nella Figura 4.8.

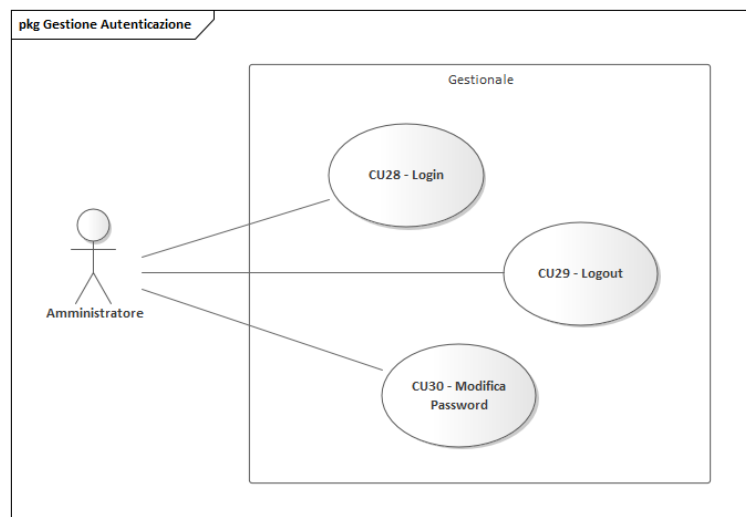


Figura 4.8: Casi d'uso relativi alla gestione dell'autenticazione

Login

Questo caso d'uso si verifica quando l'amministratore vuole accedere al sistema. Le informazioni relative sono mostrate nella Tabella 4.28.

Elementi	Descrizione
Pre-Condizioni	L'utente esiste
Post-Condizioni	Viene effettuato l'accesso
Sequenza degli eventi principali	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'amministratore vuole effettuare l'accesso al sistema 2. L'amministratore inserisce i dati nella form di autenticazione 3. Il sistema verifica che i dati siano corretti 4. Il sistema reindirizza l'amministratore nell'area a lui dedicata
Sequenza degli eventi alternativa	<ol style="list-style-type: none"> 3. L'amministratore inserisce dati errati nella form 4. Il sistema mostra gli errori generati nello schermo

Tabella 4.28: Caso d'uso relativo al login da parte dell'amministratore

Logout

Questo caso d'uso si verifica quando un amministratore vuole uscire dall'area a lui dedicata. Le informazioni relative sono mostrate nella Tabella 4.29.

Elementi	Descrizione
Pre-Condizioni	L'amministratore ha effettuato il login
Post-Condizioni	L'amministratore effettua il logout
Sequenza degli eventi principali	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando un amministratore intende uscire dall'area a lui dedicata 2. L'amministratore effettua il logout 3. Il sistema reindirizza l'amministratore alla schermata di login
Sequenza degli eventi alternativa	Nessuna

Tabella 4.29: Caso d'uso relativo al logout da parte dell'amministratore

Modifica della password

Questo caso d'uso si verifica quando un cliente vuole modificare la propria password. Le informazioni relative sono mostrate nella Tabella 4.30.

Elementi	Descrizione
Pre-Condizioni	L'amministratore ha effettuato il login
Post-Condizioni	La password viene modificata
Sequenza degli eventi principali	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'amministratore vuole modificare la propria password 2. L'amministratore inserisce la vecchia password 3. Il sistema verifica che i dati inseriti siano corretti 4. L'amministratore inserisce la nuova password 5. Il sistema verifica che i vincoli siano rispettati 6. Il sistema aggiorna la password dell'amministratore
Sequenza degli eventi alternativa	<ol style="list-style-type: none"> 3. L'amministratore inserisce informazioni errate nella form o non rispetta i vincoli 4. Il sistema mostra gli errori generati nello schermo

Tabella 4.30: Caso d'uso relativo alla prenotazione di un appuntamento

4.2 Matrice di mapping dei requisiti

La matrice di mapping è di fondamentale importanza per capire se ad ogni requisito funzionale, evidenziati nel capitolo due, è collegato almeno un caso d'uso mostrato nelle

precedenti sottosezioni. Se questo accade, potremo affermare che tutti i requisiti sono stati rispettati.

In Figura 4.9 è riportata la matrice di mapping tra requisiti funzionali e casi d’uso.

Source \ Target	Gestione Agenda::RF1 - Visualizzazione Agenda	Gestione Agenda::RF2 - Inserimento Appuntamento	Gestione Agenda::RF3 - Modifica Appuntamento	Gestione Agenda::RF4 - Eliminazione Appuntamento	Gestione Autenticazione::RF5 - Autenticazione Amministratore	Gestione FAQ::RF6 - Visualizzazione FAQ	Gestione FAQ::RF7 - Inserimento FAQ	Gestione FAQ::RF8 - Modifica FAQ	Gestione FAQ::RF9 - Eliminazione FAQ	Gestione Fatture::RF10 - Visualizzazione Fatture	Gestione Fatture::RF11 - Inserimento Fattura	Gestione Fatture::RF12 - Modifica Fattura	Gestione Fatture::RF13 - Eliminazione Fattura	Gestione Prenotazioni::RF14 - Prenotazione Appuntamento	Gestione Prodotti::RF15 - Visualizzazione Inventario	Gestione Prodotti::RF16 - Inserimento Prodotto	Gestione Prodotti::RF17 - Modifica Prodotto	Gestione Prodotti::RF18 - Eliminazione Prodotto	Gestione Tariffe::RF19 - Visualizzazione Tariffario	Gestione Tariffe::RF20 - Inserimento Tariffa	Gestione Tariffe::RF21 - Modifica Tariffa	Gestione Tariffe::RF22 - Eliminazione Tariffa
Gestione agenda::CU1 - Visualizza Appuntamento	↑																					
Gestione agenda::CU2 - Inserimento Appuntamento		↑																				
Gestione agenda::CU3 - Modifica Appuntamento			↑																			
Gestione agenda::CU4 - Eliminazione Appuntamento				↑																		
Gestione agenda::CU5 - Ricerca Appuntamento	↑	↑	↑	↑																		
Gestione Autenticazione::CU28 - Login					↑																	
Gestione Autenticazione::CU29 - Logout					↑																	
Gestione Autenticazione::CU30 - Modifica Password					↑																	
Gestione FAQ::CU22 - Visualizzazione FAQ						↑																
Gestione FAQ::CU23 - Inserimento FAQ							↑															
Gestione FAQ::CU24 - Modifica FAQ								↑														
Gestione FAQ::CU25 - Eliminazione FAQ									↑													
Gestione FAQ::CU26 - Ricerca FAQ						↑	↑	↑	↑													
Gestione Fattura::CU11 - Visualizzazione Fatture										↑												
Gestione Fattura::CU12 - Inserimento Fattura											↑											
Gestione Fattura::CU13 - Modifica Fattura												↑										
Gestione Fattura::CU14 - Eliminazione Fattura													↑									
Gestione Fattura::CU15 - Ricerca Fattura										↑	↑	↑	↑									
Gestione Fattura::CU16 - Generazione PDF										↑												
Gestione Prenotazioni::CU27 - Richiesta Appuntamento														↑								
Gestione Prodotti::CU17 - Visualizzazione Prodotti															↑							
Gestione Prodotti::CU18 - Inserimento Prodotto																↑						
Gestione Prodotti::CU19 - Modifica Prodotto																	↑					
Gestione Prodotti::CU20 - Eliminazione Prodotto																		↑				
Gestione Prodotti::CU21 - Ricerca Prodotto															↑	↑	↑	↑				
Gestione Tariffe::CU10 - Ricerca Tariffa																			↑	↑	↑	↑
Gestione Tariffe::CU6 - Visualizzazione Tariffe																			↑			
Gestione Tariffe::CU7 - Inserimento Tariffe																				↑		
Gestione Tariffe::CU8 - Modifica Tariffe																					↑	
Gestione Tariffe::CU9 - Eliminazione Tariffe																						↑

Figura 4.9: Matrice di mapping dei requisiti

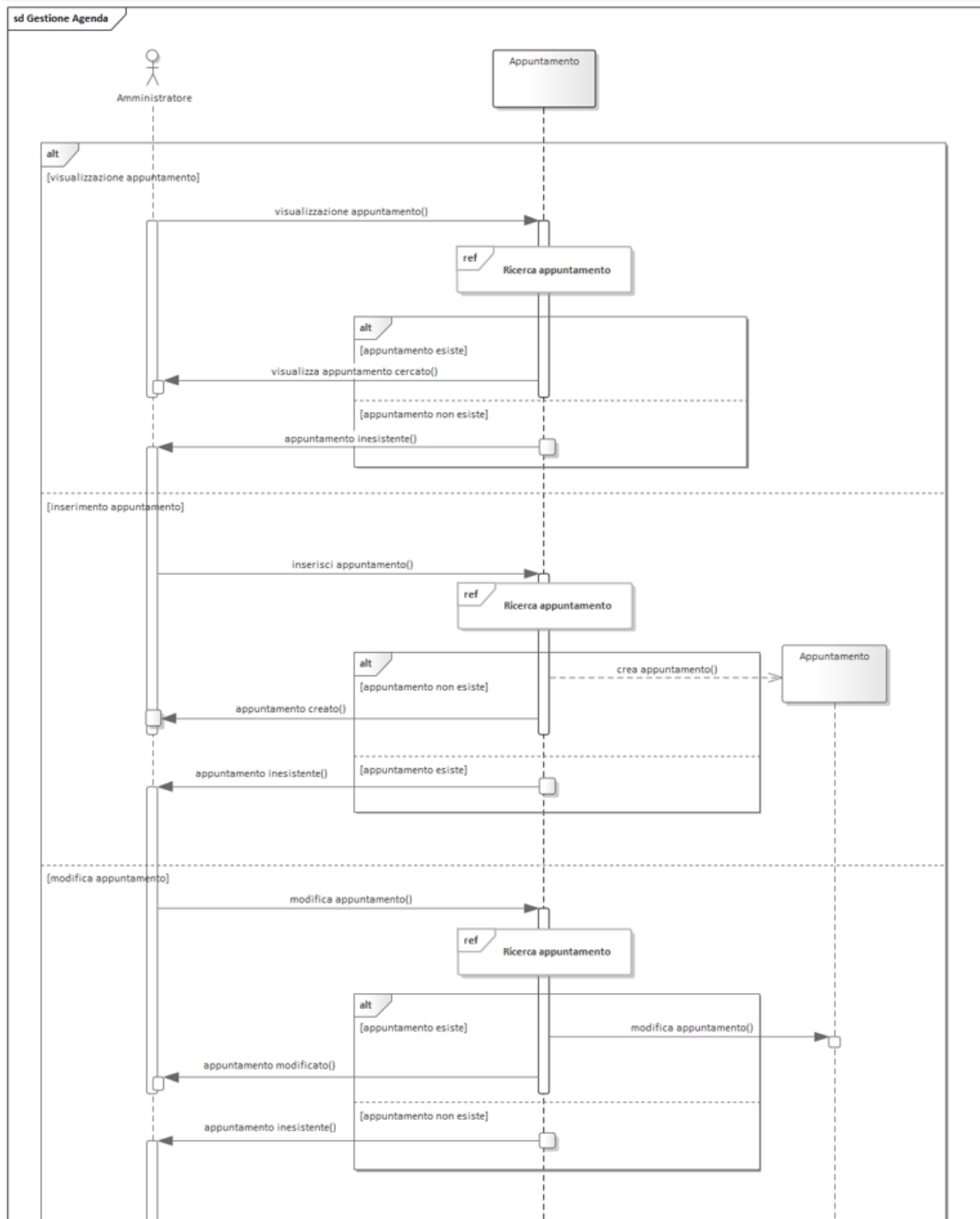
4.3 Diagrammi di sequenza

In questa sezione utilizzeremo i diagrammi di sequenza per modellare le interazioni tra gli attori e gli oggetti del nostro sistema e le interazioni tra gli stessi oggetti. I diagrammi di

sequenza in esame mostreranno la sequenza delle interazioni che svolte in un particolare caso d'uso.

4.3.1 Gestione dell'agenda

Questa sottosezione riguarda la sequenza degli eventi relativi alla gestione dell'agenda in cui è possibile visualizzare, ricercare, inserire, modificare o eliminare un appuntamento. Il diagramma è mostrato nella Figura 4.10.



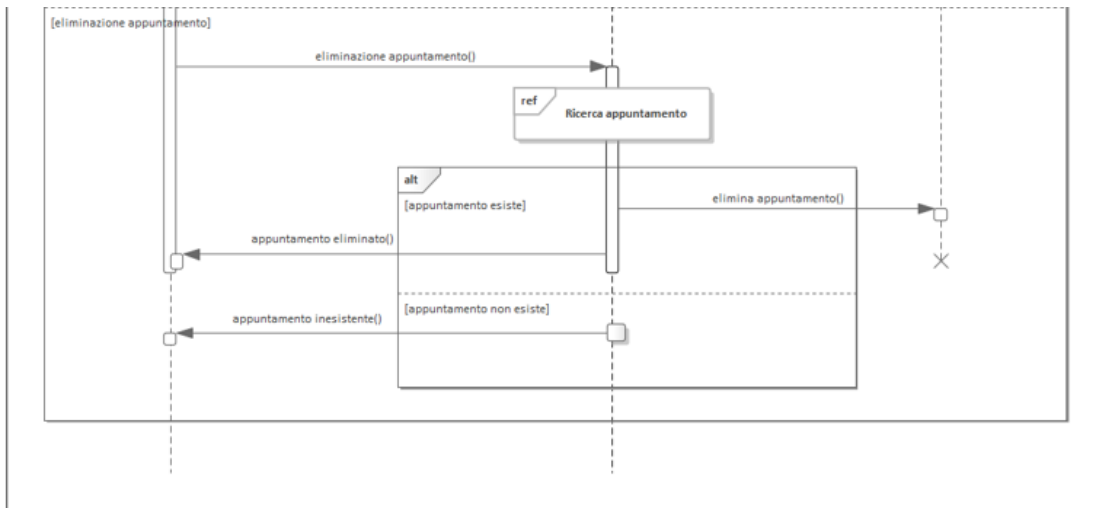
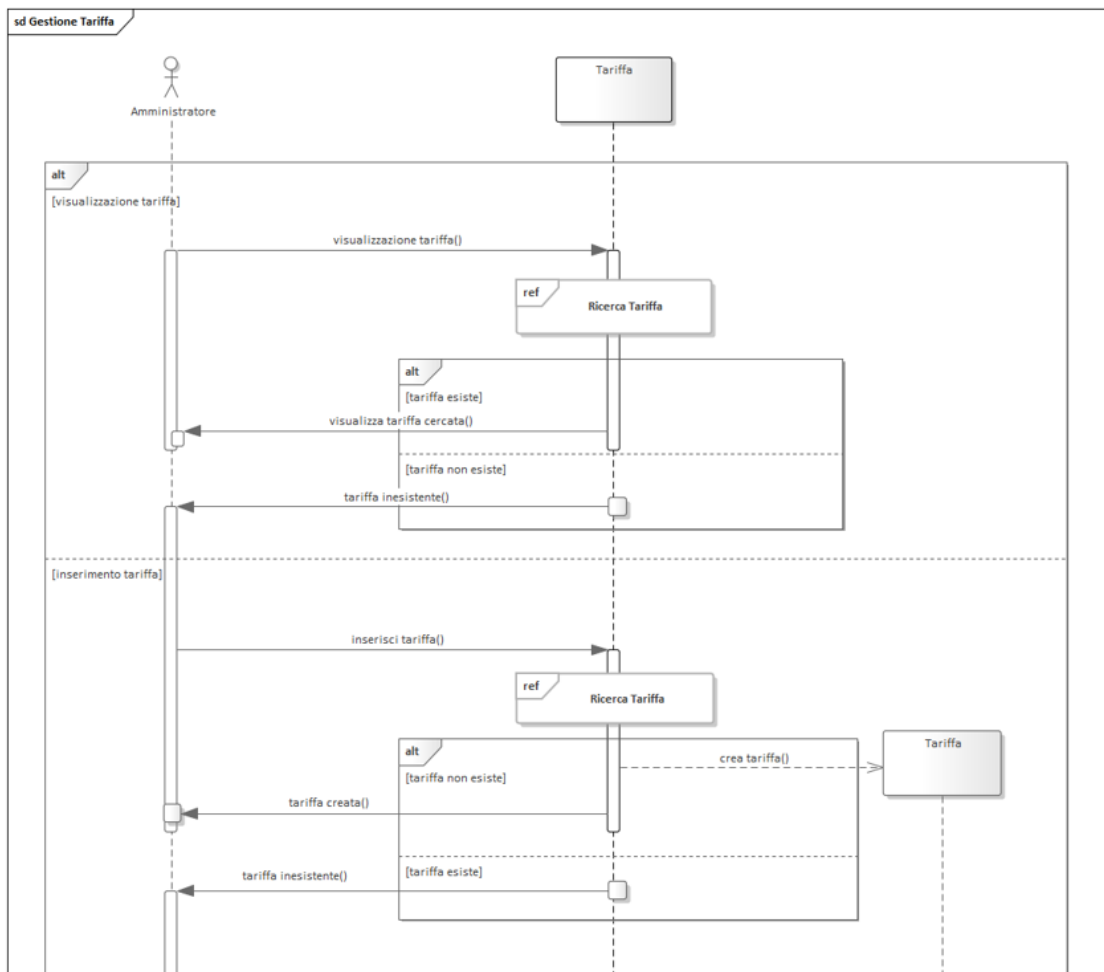


Figura 4.10: Diagramma di sequenza relativo alla gestione dell'agenda

4.3.2 Gestione delle tariffe

Questa sottosezione riguarda la sequenza degli eventi relativi alla gestione delle tariffe. Anche in questo caso è possibile visualizzare, ricercare, inserire, modificare o eliminare un appuntamento, come mostrato nella Figura 4.11.



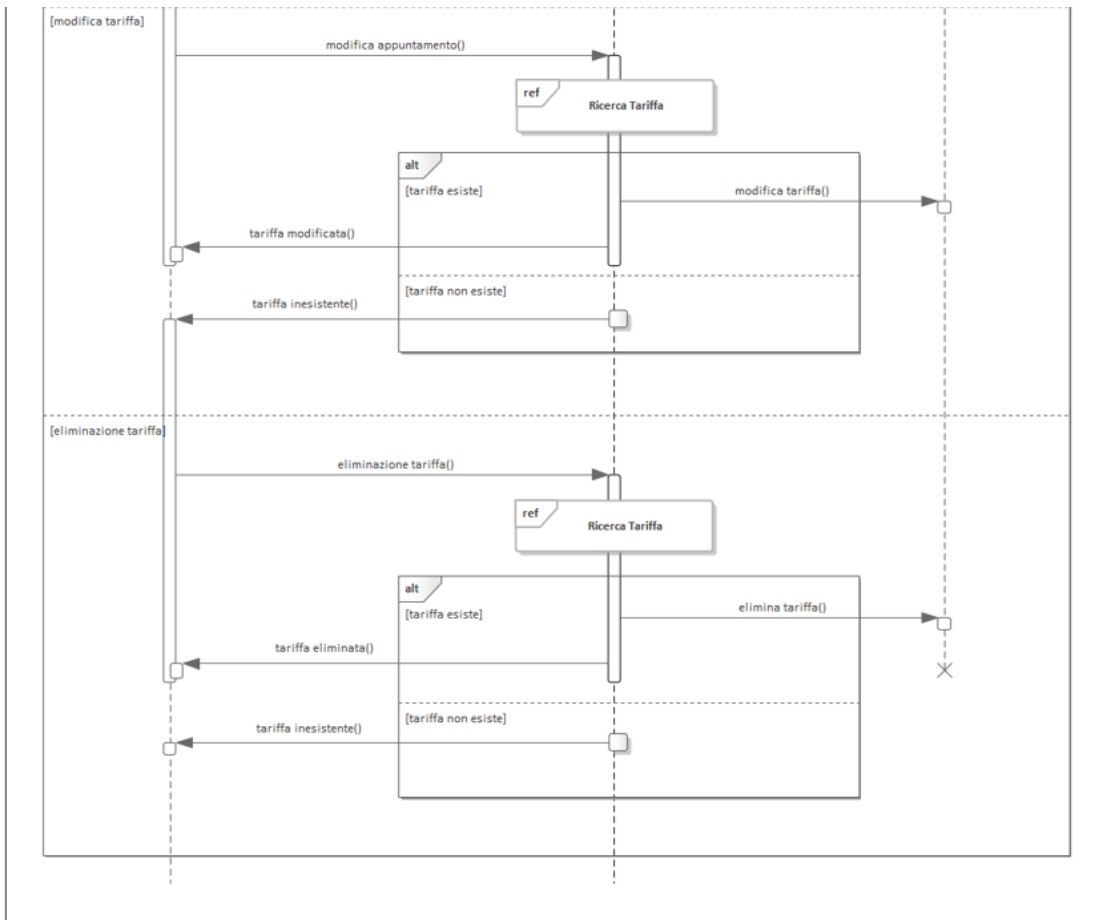
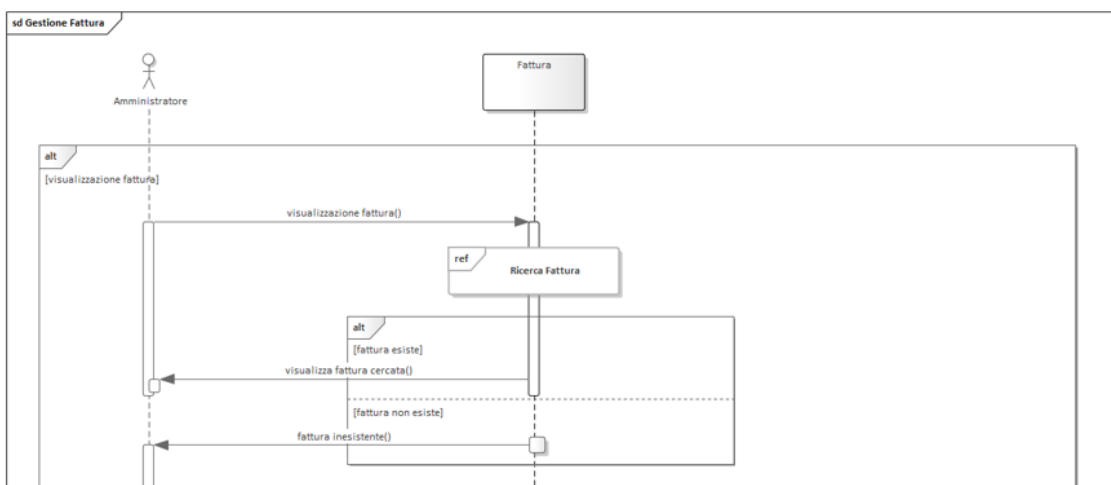


Figura 4.11: Diagramma di sequenza relativo alla gestione delle tariffe

4.3.3 Gestione delle fatture

Questa sottosezione riguarda la sequenza degli eventi relativi alla gestione delle fatture. In questo caso, in aggiunta alle precedenti, è possibile generare un file PDF con le informazioni di una fattura. Il diagramma è come in Figura 4.12.



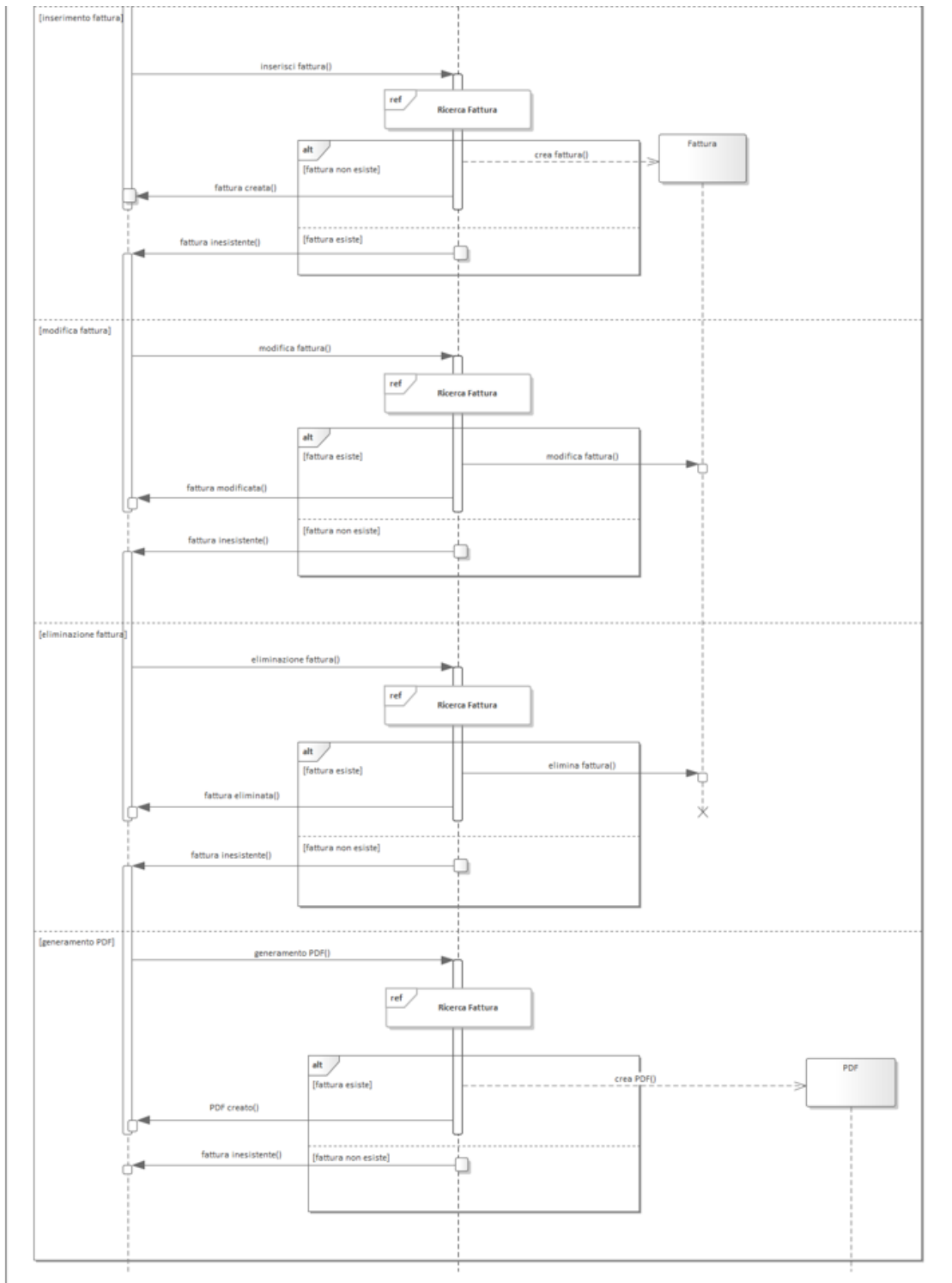


Figura 4.12: Diagramma di sequenza relativo alla gestione delle fatture

4.3.4 Gestione dei prodotti

Questa sottosezione riguarda la sequenza degli eventi relativi alla gestione dei prodotti in cui è possibile visualizzare, ricercare, inserire, modificare o eliminare un prodotto. Il diagramma è mostrato in Figura 4.13.

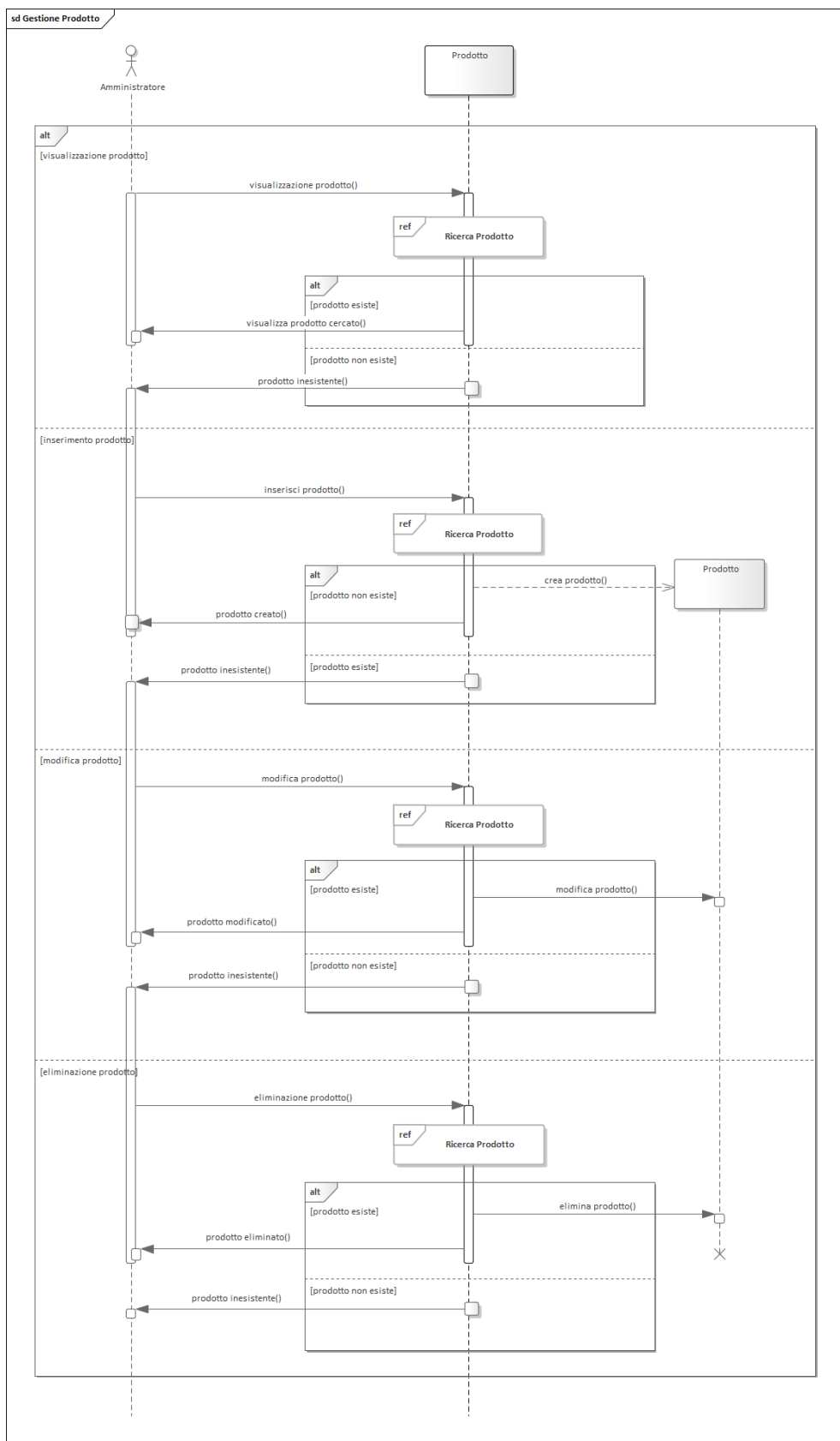
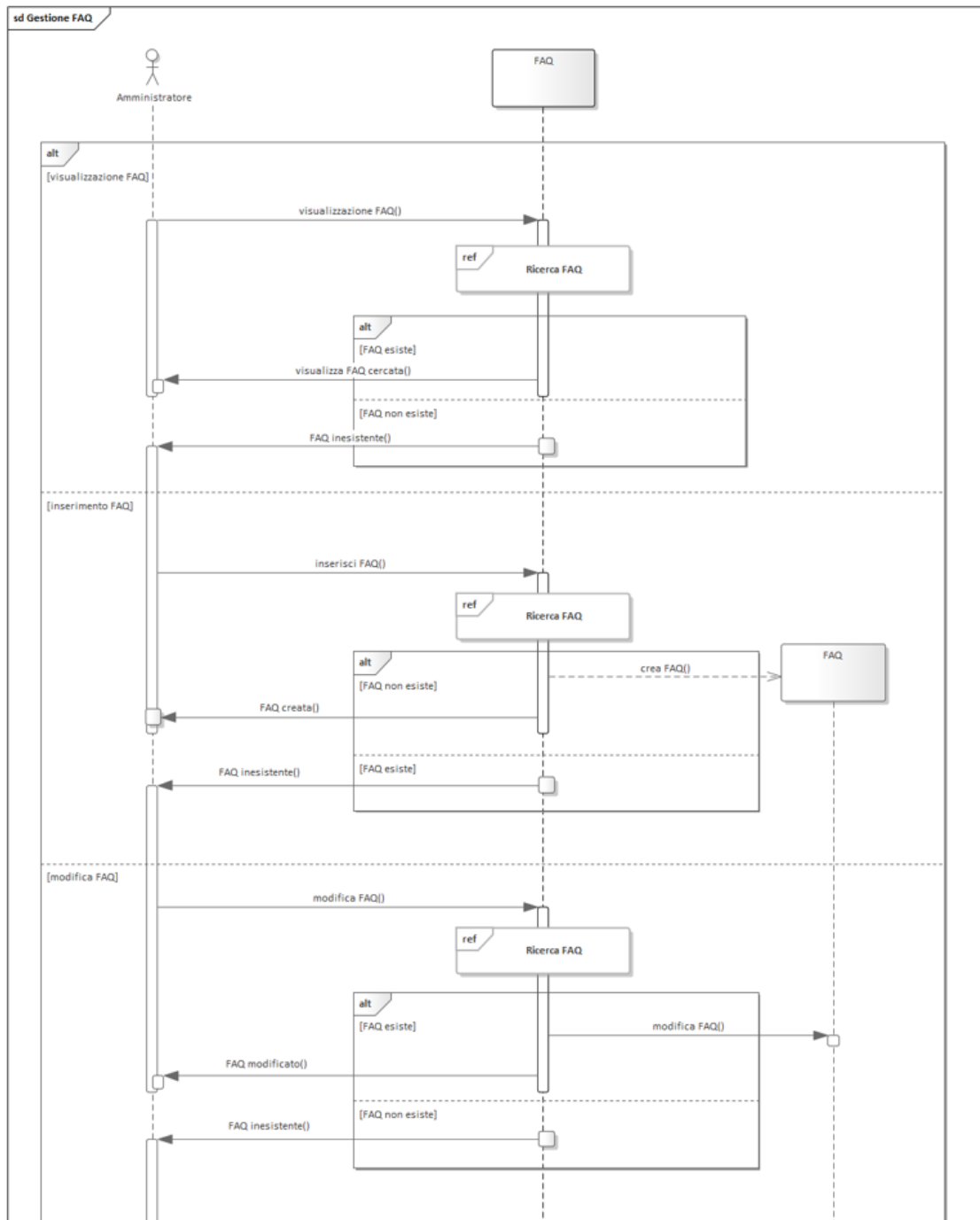


Figura 4.13: Diagramma di sequenza relativo alla gestione dei prodotti

4.3.5 Gestione delle FAQ

La quinta sottosezione riguarda la sequenza degli eventi relativi alla gestione delle FAQ in cui è possibile visualizzare, ricercare, inserire, modificare o eliminare una FAQ da parte dell'amministratore, mentre è possibile soltanto visualizzare la pagina contenente tutte le FAQ da parte di un cliente. Il diagramma è mostrato nelle figure 4.14 e 4.15.



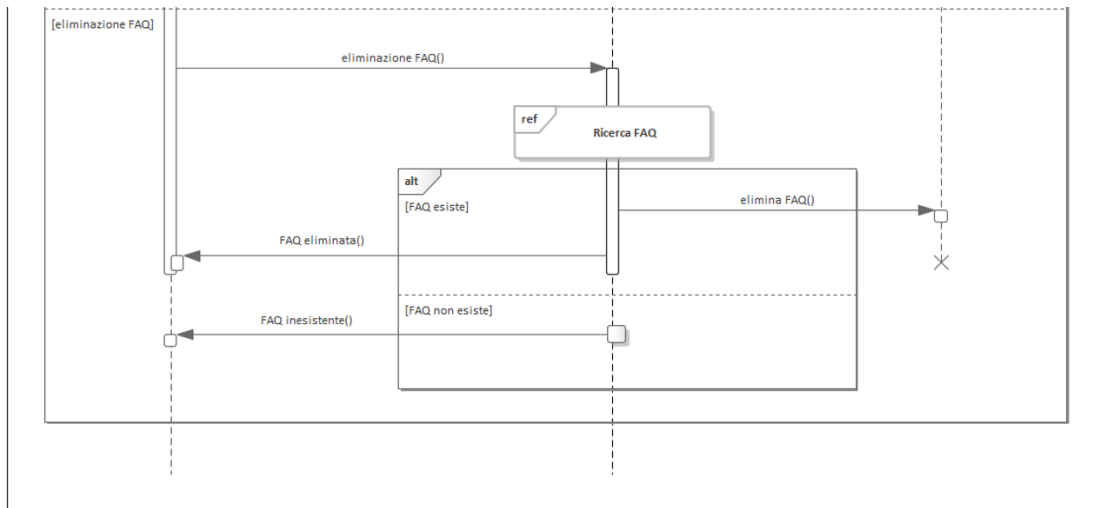


Figura 4.14: Diagramma di sequenza relativo alla gestione delle FAQ per l'amministratore

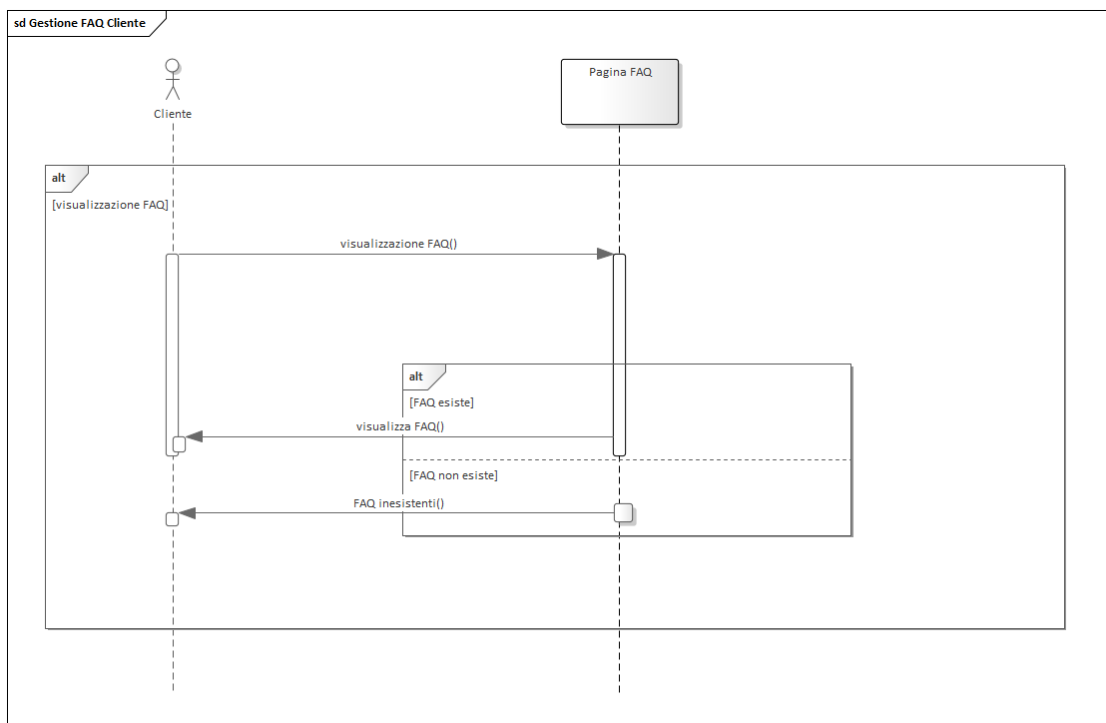


Figura 4.15: Diagramma di sequenza relativo alla gestione delle FAQ per il cliente

4.3.6 Gestione delle prenotazioni

Questa sottosezione riguarda la sequenza degli eventi relativi alla gestione delle prenotazioni. La sequenza ci mostra come sia possibile effettuare una prenotazione tramite compilazione di una form, la quale verrà inviata all'amministratore tramite email. Il diagramma è mostrato in Figura 4.16.

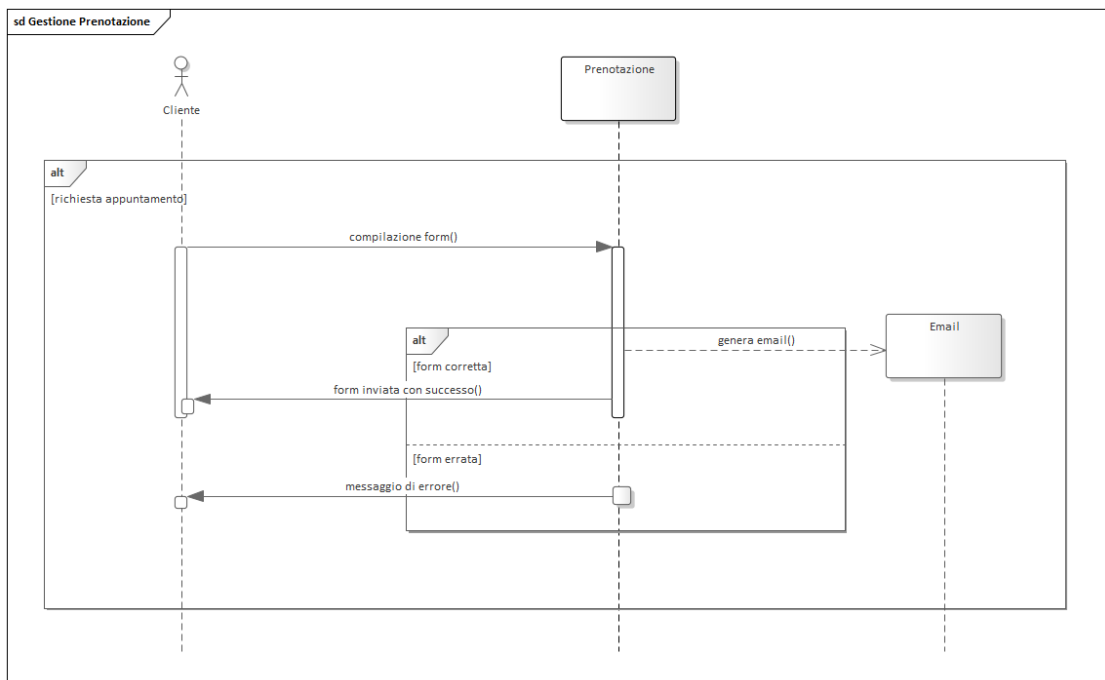
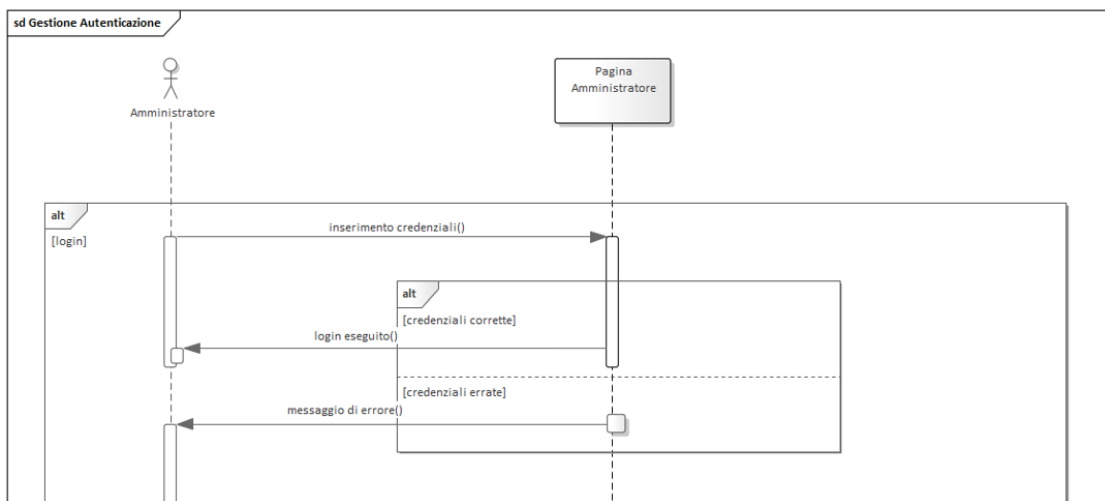


Figura 4.16: Diagramma di sequenza relativo alla gestione delle prenotazioni

4.3.7 Gestione dell'autenticazione

Infine la seguente sottosezione riguarda la sequenza degli eventi relativi alla gestione dell'autenticazione, in cui è possibile effettuare il login, il logout e la modifica della propria password. Il diagramma è mostrato in Figura 4.17.



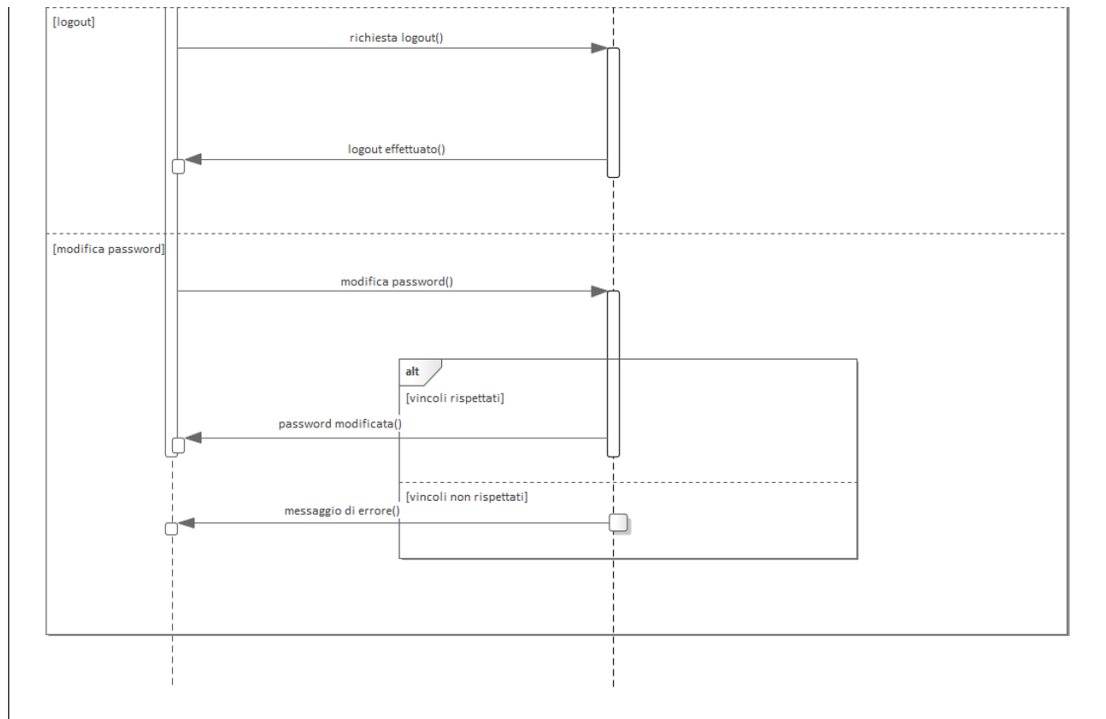


Figura 4.17: Diagramma di sequenza relativo alla gestione dell'autenticazione

4.4 Diagramma di attività

I diagrammi di attività ci mostrano che cosa accade, o che cosa si suppone che accada, quando un sistema risponde a uno stimolo del suo ambiente. Gli stimoli possono essere dati o eventi.

Come per i diagrammi di sequenza, per ogni scenario possibile nel nostro software forniremo un diagramma di attività che verrà mostrato in una sottosezione ad esso dedicato.

4.4.1 Gestione dell'agenda

Questa sottosezione riguarda la gestione dell'agenda. Il diagramma di attività è mostrato in Figura 4.18.

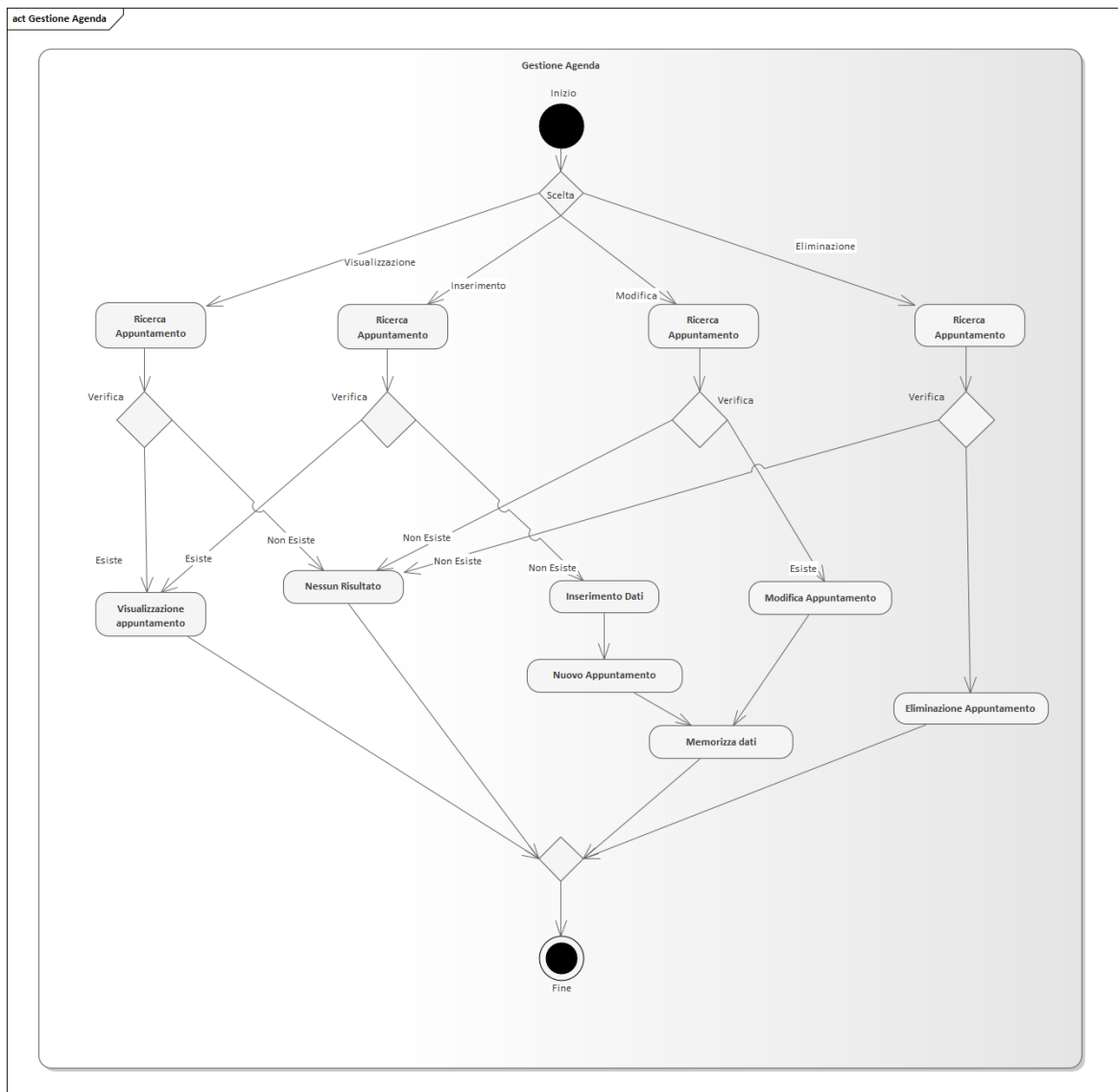


Figura 4.18: Diagramma di attività relativo alla gestione dell'agenda

4.4.2 Gestione delle tariffe

Questa sottosezione riguarda la gestione delle tariffe. Il diagramma di attività è mostrato in Figura 4.19.

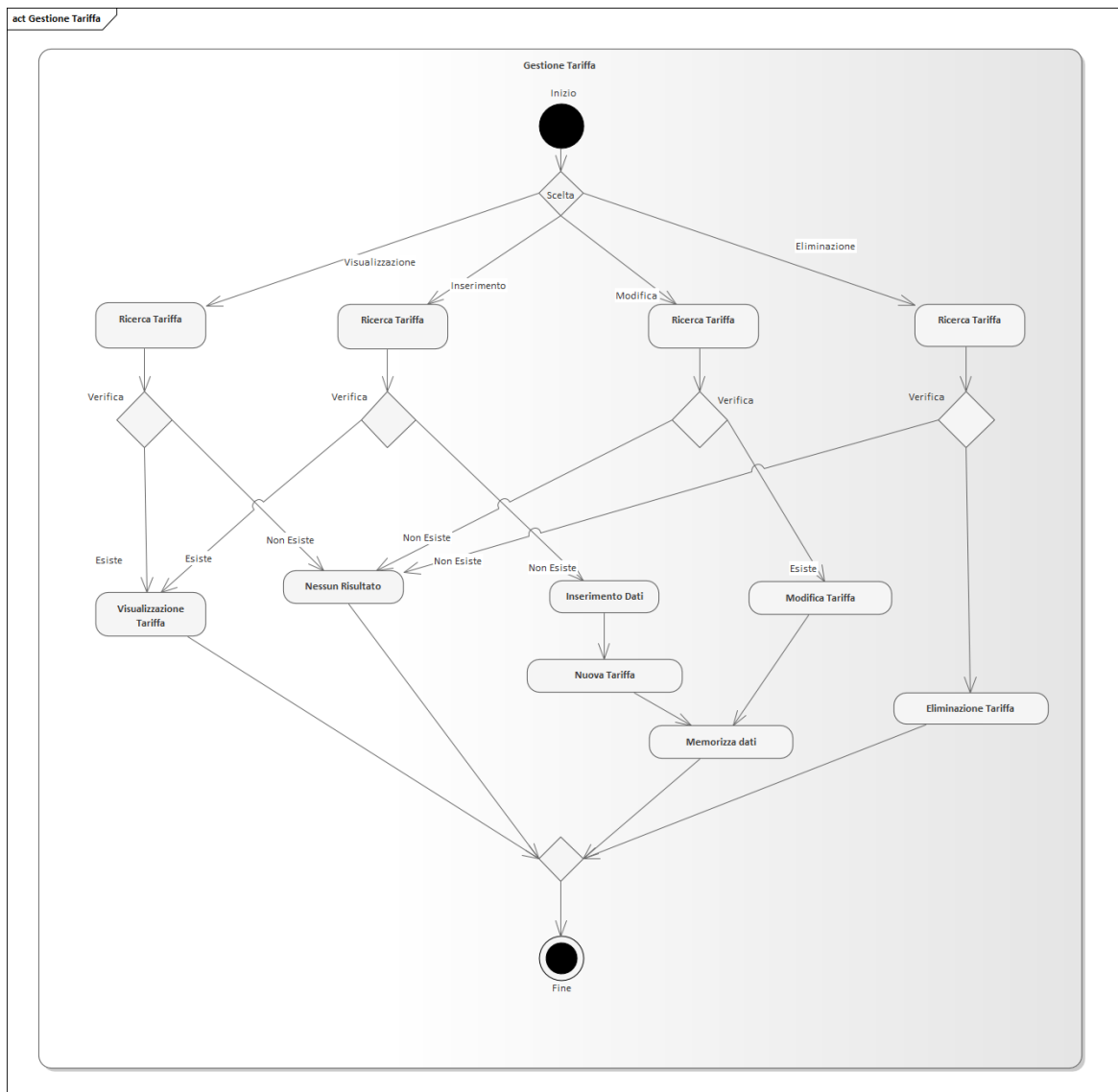


Figura 4.19: Diagramma di attività relativo alla gestione delle tariffe

4.4.3 Gestione delle fatture

Questa sottosezione riguarda la gestione delle fatture. Il diagramma di attività è mostrato in Figura 4.20.

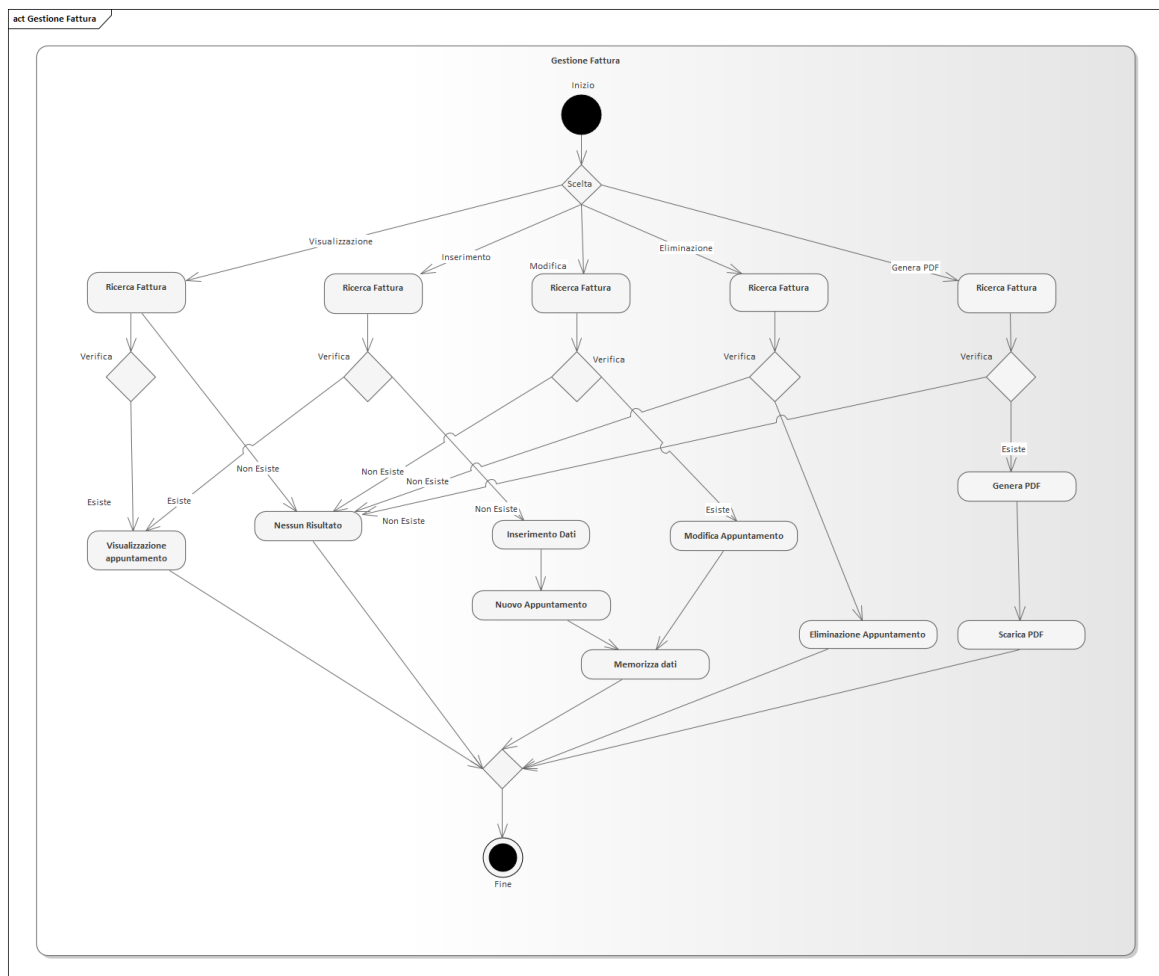


Figura 4.20: Diagramma di attività relativo alla gestione delle fatture

4.4.4 Gestione dei prodotti

Questa sottosezione riguarda la gestione dell'agenda. Il diagramma di attività è mostrato in Figura 4.21.

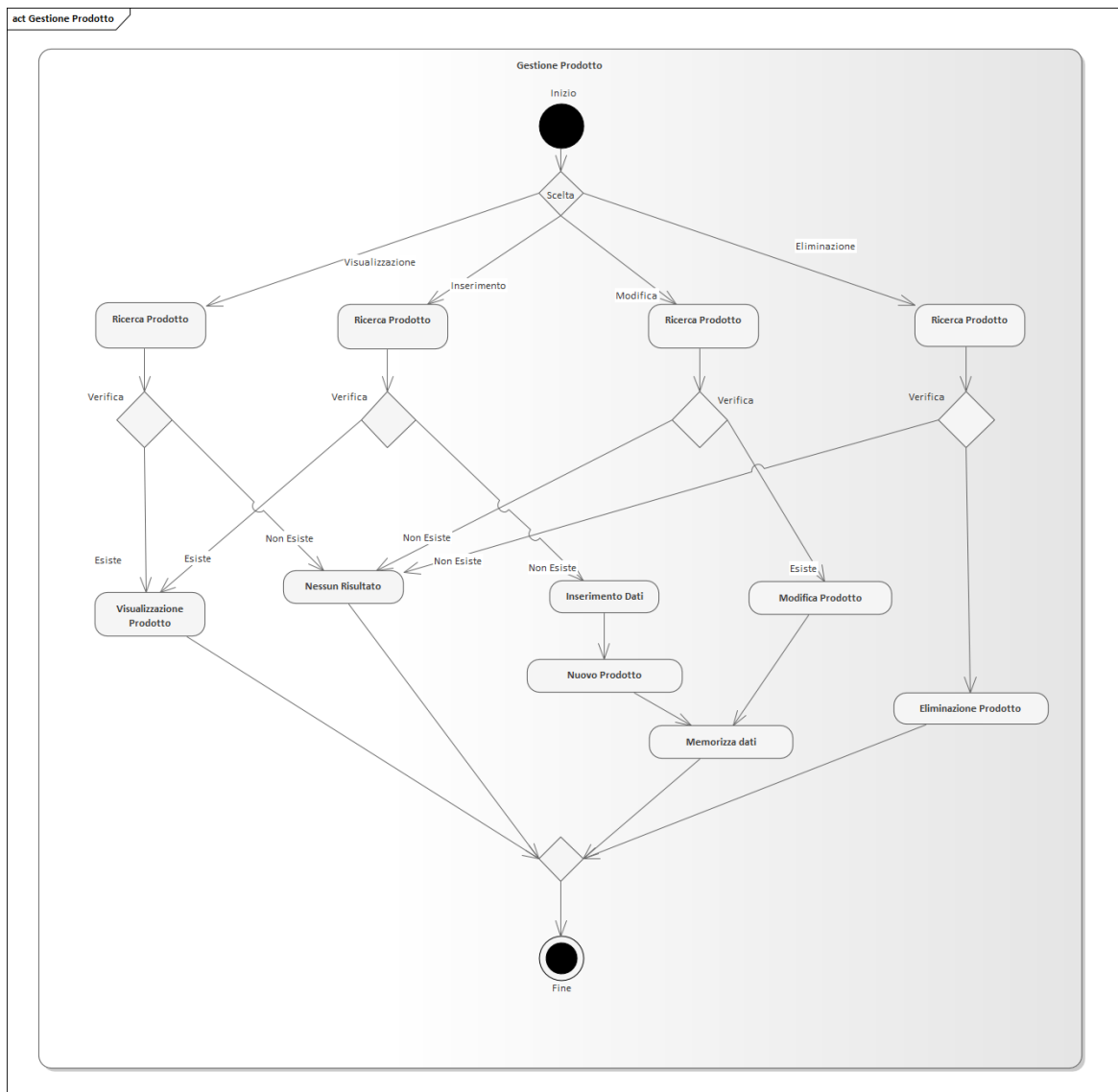


Figura 4.21: Diagramma di attività relativo alla gestione dei prodotti

4.4.5 Gestione delle FAQ

Questa sottosezione riguarda la gestione delle FAQ. Il diagramma di attività è mostrato in Figura 4.22.

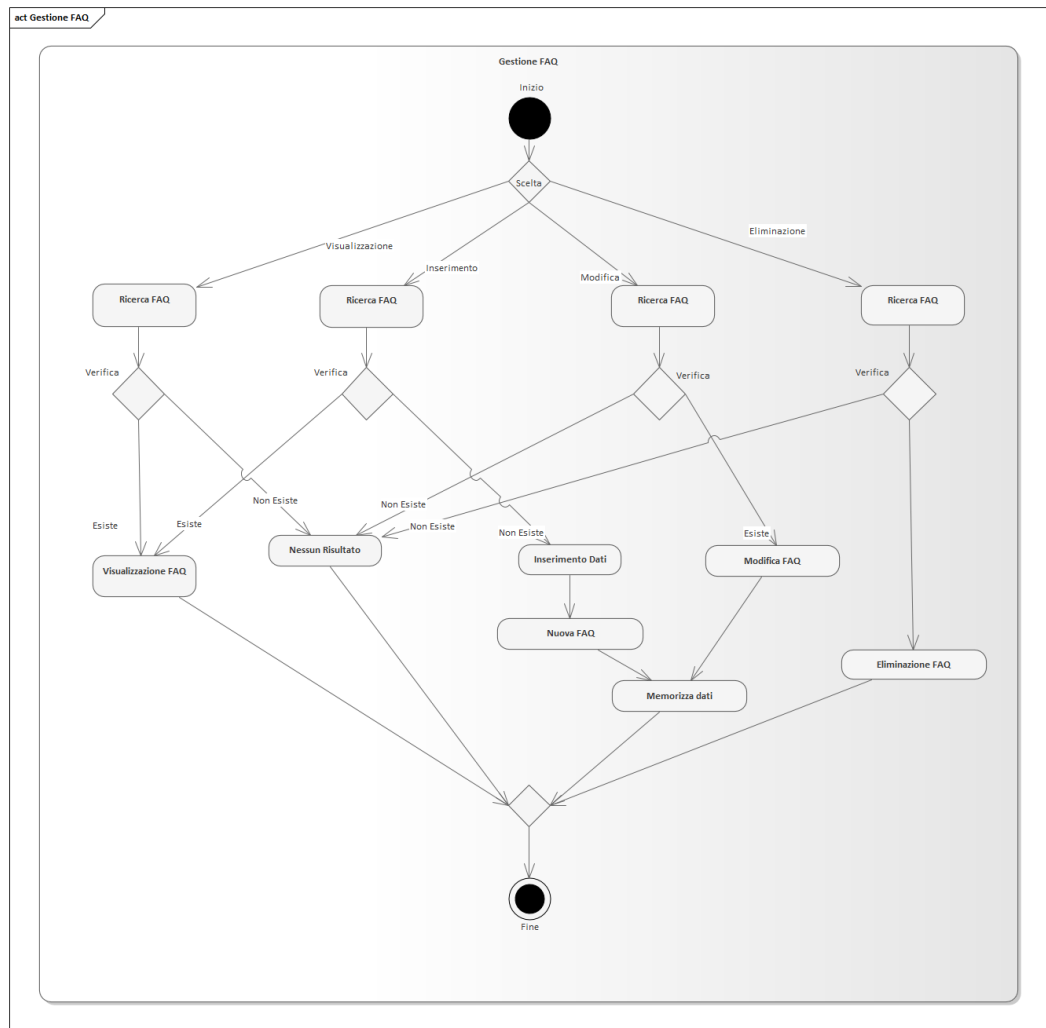


Figura 4.22: Diagramma di attività relativo alla gestione delle FAQ per un amministratore

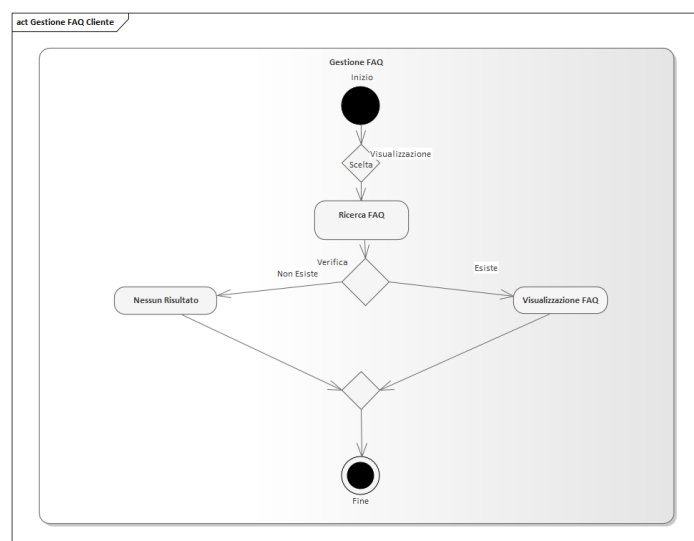


Figura 4.23: Diagramma di attività relativo alla gestione delle FAQ per un cliente

4.4.6 Gestione delle prenotazioni

Questa sottosezione riguarda la gestione delle prenotazioni. Il diagramma di attività è mostrato nella Figura 4.23.

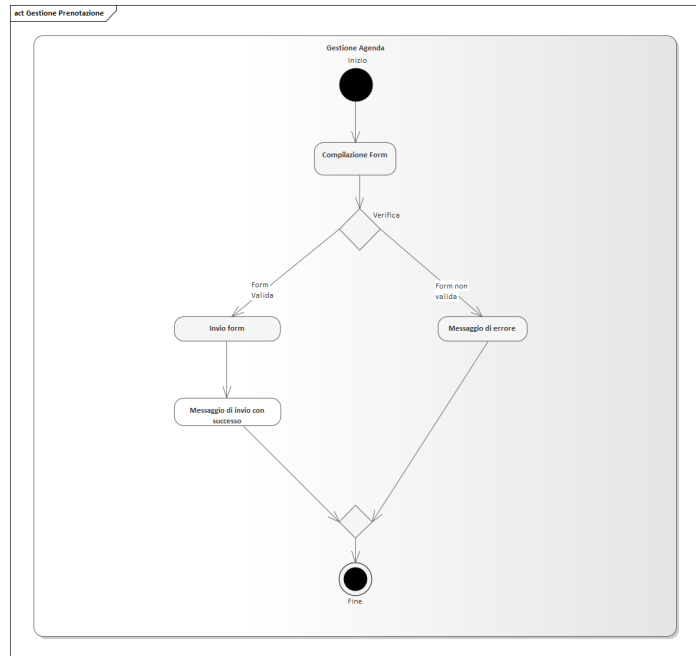


Figura 4.24: Diagramma di attività relativo alla gestione delle prenotazioni

4.4.7 Gestione dell'autenticazione

L'ultima sottosezione riguarda la gestione dell'autenticazione. Il diagramma di attività è mostrato in Figura 4.24.

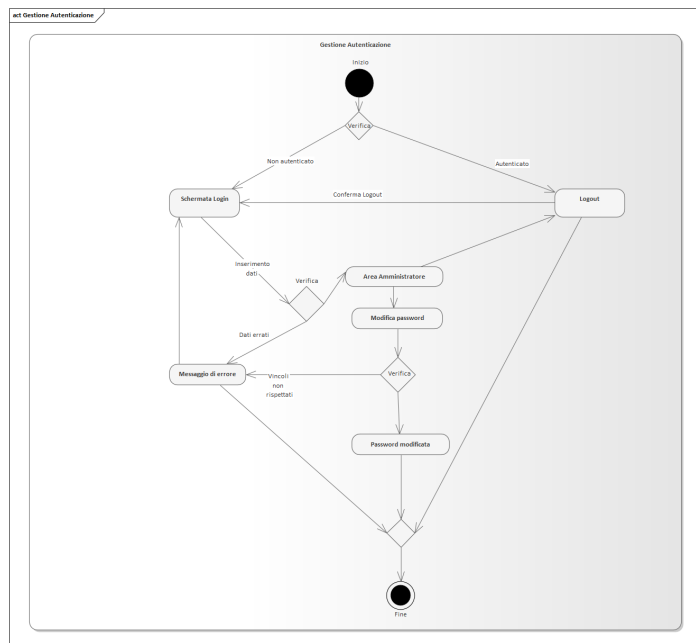


Figura 4.25: Diagramma di attività relativo alla gestione dell'autenticazione

Definizione dei mockup

In questo capitolo mostreremo i mockup del progetto, ossia l'anello di congiunzione tra quanto è stato mostrato fino ad ora e l'effettivo prodotto finale. Abbiamo suddiviso i mockup in base all'utenza, quindi, amministratore e cliente, e, per ognuno di essi, abbiamo analizzato tutte le schermate di interazione seguendo la modellazione dei capitoli precedenti.

5.1 Area Amministratore

I mockup relativi all'area amministratore sono predefiniti da Django; dunque quello che è stato fatto riguarda principalmente l'importazione delle tabelle da noi modellate e qualche piccolo ritocco al CSS. Per ogni area verrà mostrato il mockup in questione.

5.1.1 Login

La schermata di login può essere acceduta alla rotta `admin` ed è mostrata in Figura 5.1.

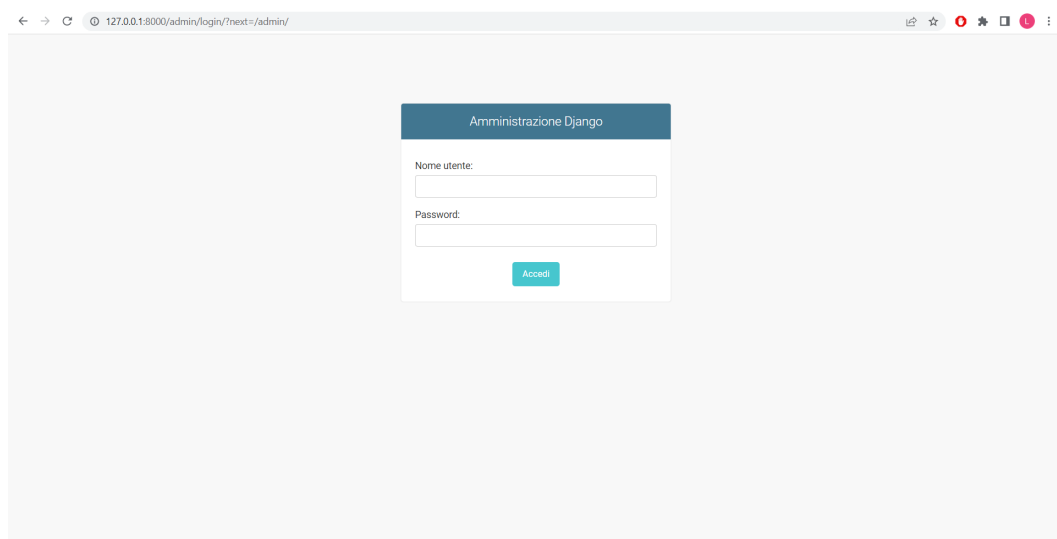


Figura 5.1: Mockup della schermata di login

5.1.2 Home

Una volta effettuato il login, la prima pagina che verrà mostrata è quella della home page, la quale comprende le tabelle Agenda, FAQ, Fattura, Prodotto e Tariffa, oltre ad Autenticazione e Autorizzazione, che, però, sono predefinite. Cliccando sopra la tabella di interesse, verremo riportati alla pagina corrispondente. Da questa schermata è possibile, inoltre, effettuare il logout e modificare la propria password. La home page è mostrata in Figura 5.2

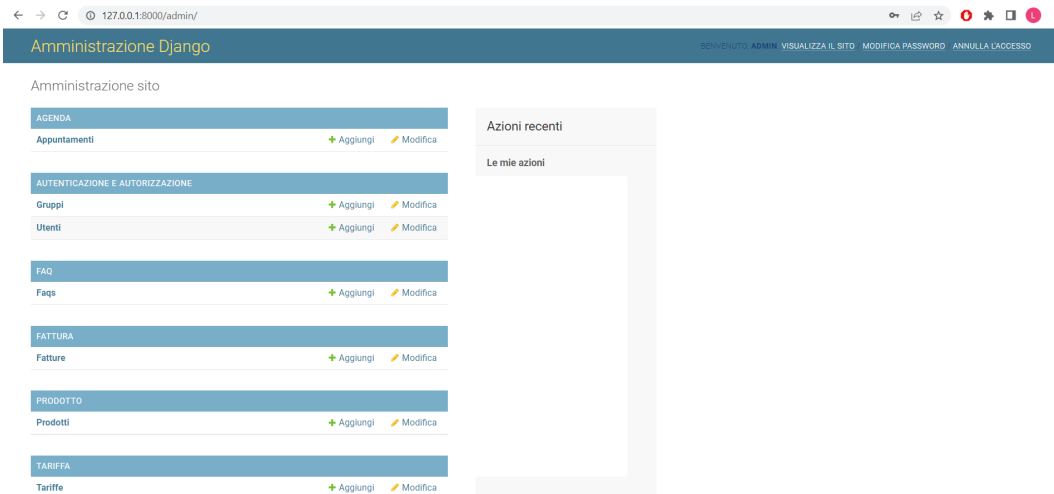


Figura 5.2: Mockup della schermata home page

5.1.3 Agenda

All'interno dell'agenda è possibile inserire, modificare, ricercare oppure eliminare un appuntamento. Questa sezione è stata manualmente modificata con HTMLCalendar per renderla più user-friendly, come mostrato nelle Figure 5.3 - 5.7.

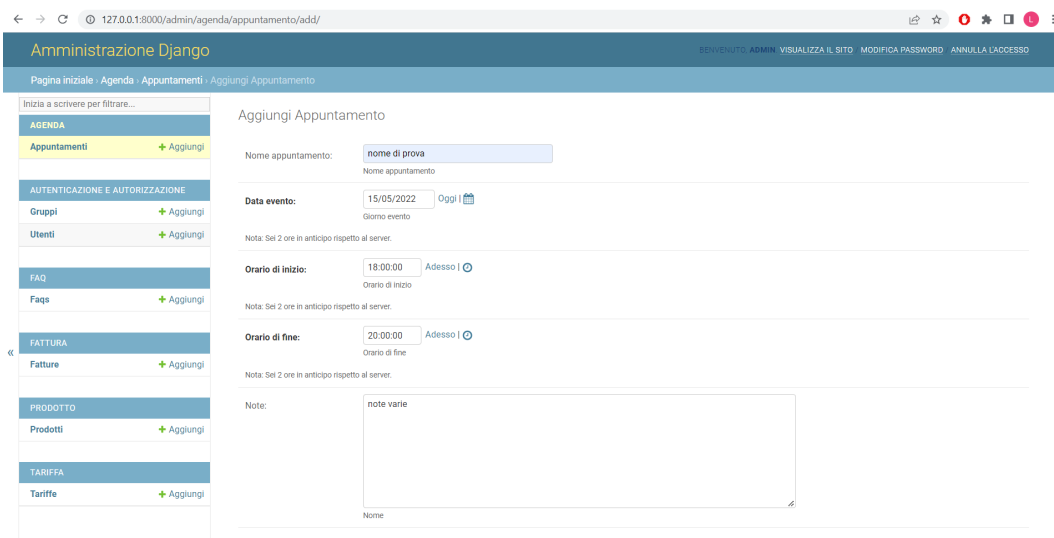


Figura 5.3: Mockup della schermata relativa all'agenda: inserimento/modifica di un appuntamento

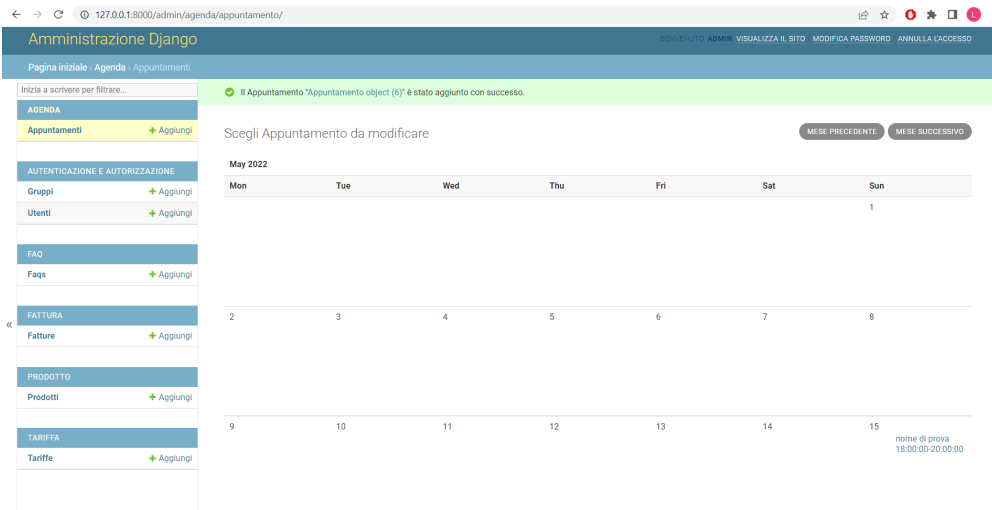


Figura 5.4: Mockup della schermata relativa all’agenda: visualizzazione di un appuntamento

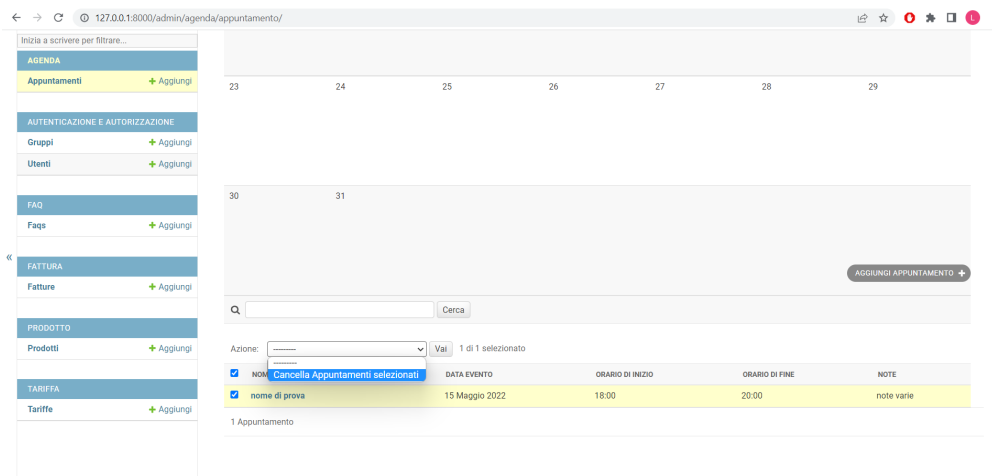


Figura 5.5: Mockup della schermata relativa all’agenda: eliminazione di un appuntamento

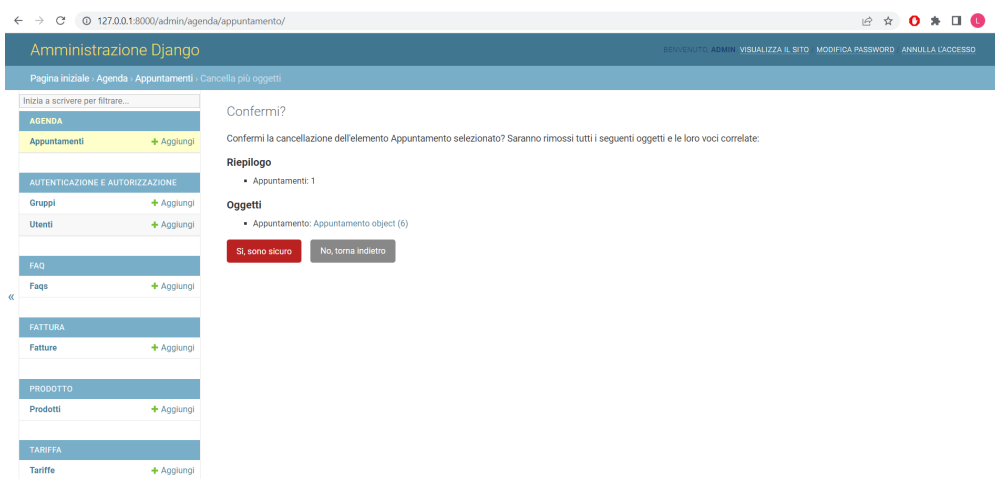


Figura 5.6: Mockup della schermata relativa all’agenda: eliminazione di un appuntamento

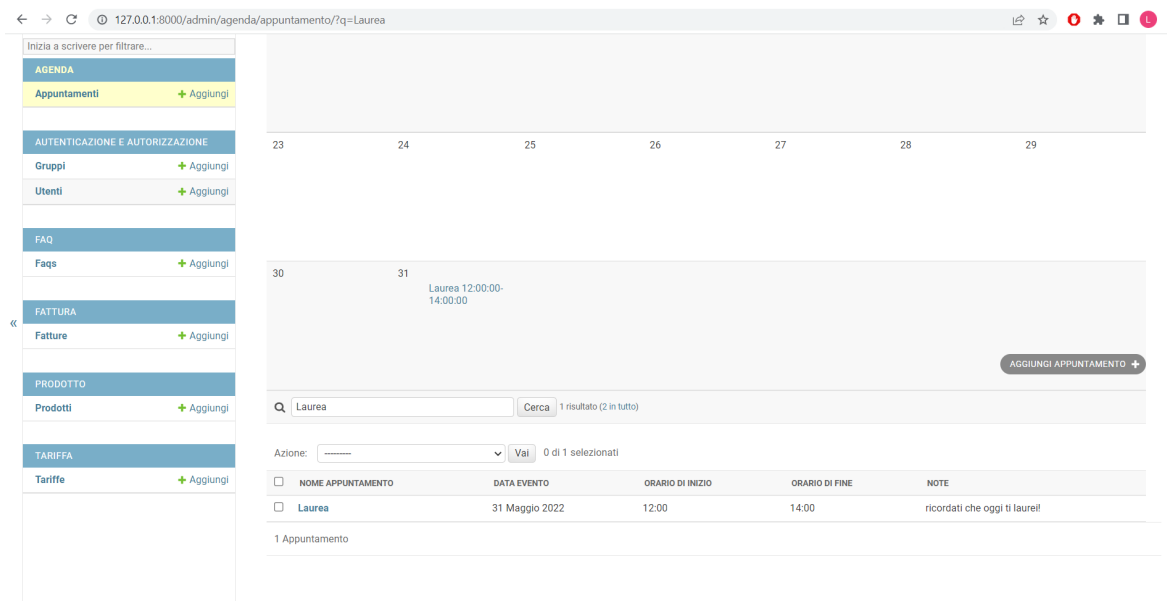


Figura 5.7: Mockup della schermata relativa all'agenda: ricerca di un appuntamento

5.1.4 FAQ

All'interno di questa sezione è possibile inserire, modificare, ricercare oppure eliminare una FAQ, come mostrato nelle Figure 5.8 - 5.11.

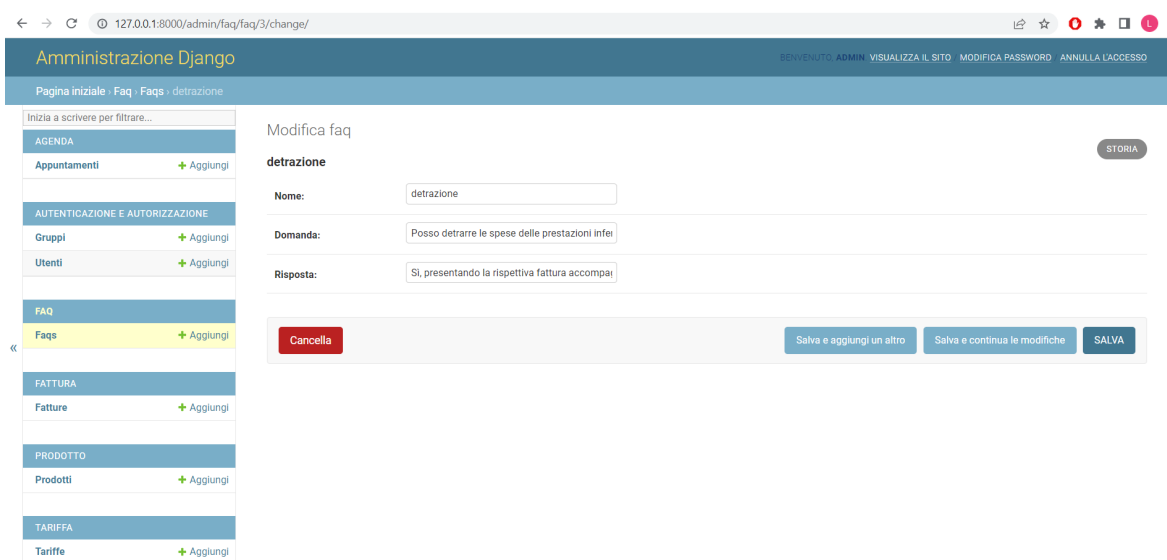


Figura 5.8: Mockup della schermata relativa alle FAQ: inserimento/modifica di una FAQ

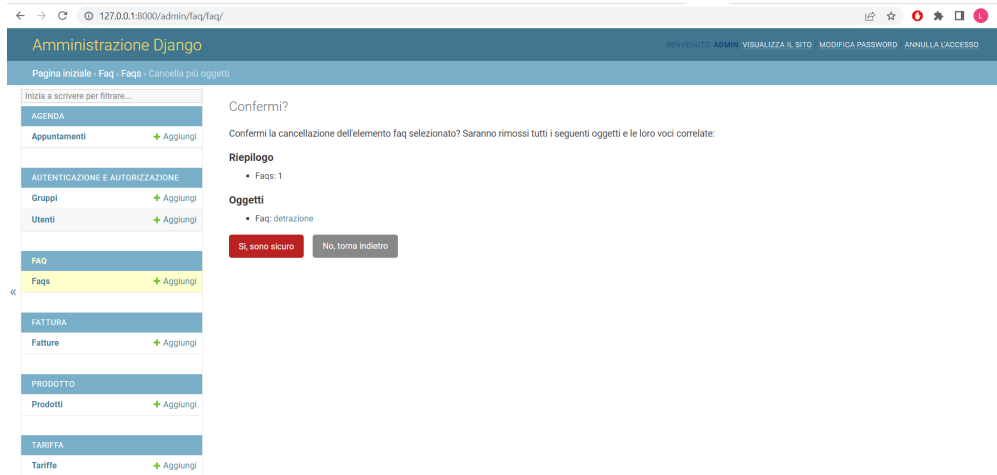


Figura 5.9: Mockup della schermata relativa alle FAQ: eliminazione di una FAQ

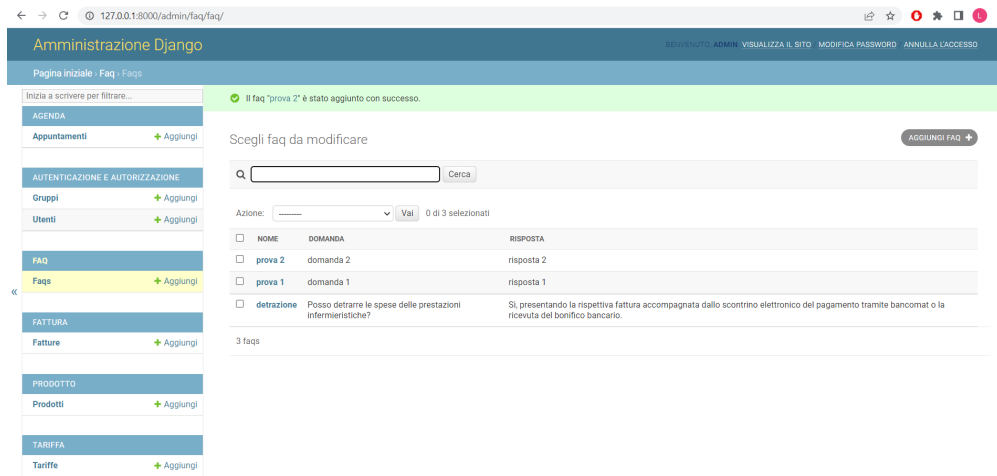


Figura 5.10: Mockup della schermata relativa alle FAQ: visualizzazione di tutte le FAQ

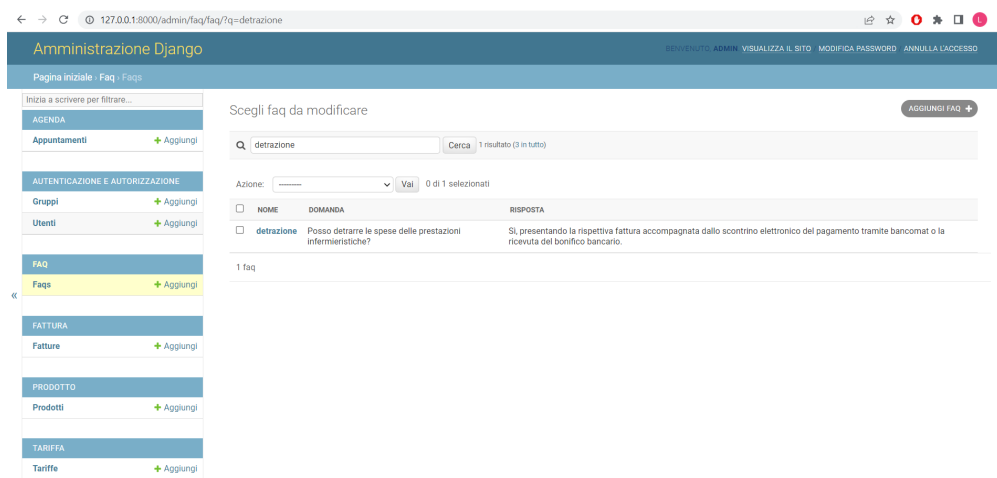


Figura 5.11: Mockup della schermata relativa alle FAQ: ricerca di una FAQ

5.1.5 Fattura

La sezione relativa alle fatture comprende, in aggiunta alle funzioni mostrate nei casi precedenti, la possibilità di generare il file PDF a partire dalla fattura.

Quando si crea una fattura, è possibile inserire vari prodotti grazie all'utilizzo della chiave esterna presente nel modello relativo ai prodotti della fattura. I mockup sono mostrati nelle Figure 5.12 - 5.17.

Figura 5.12: Mockup della schermata relativa alle fatture: inserimento/modifica di una fattura

Figura 5.13: Mockup della schermata relativa alle fatture: inserimento/modifica di una fattura

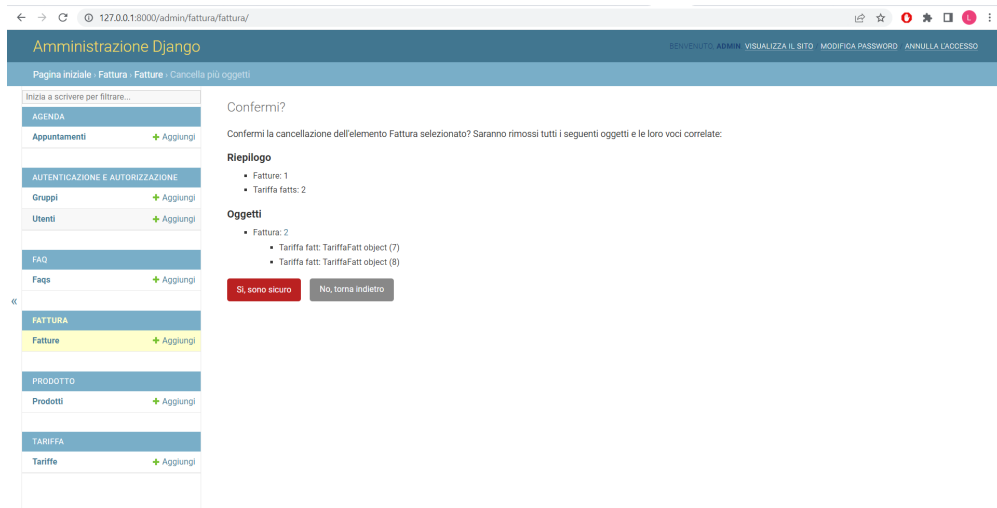


Figura 5.14: Mockup della schermata relativa alle fatture: eliminazione di una fattura

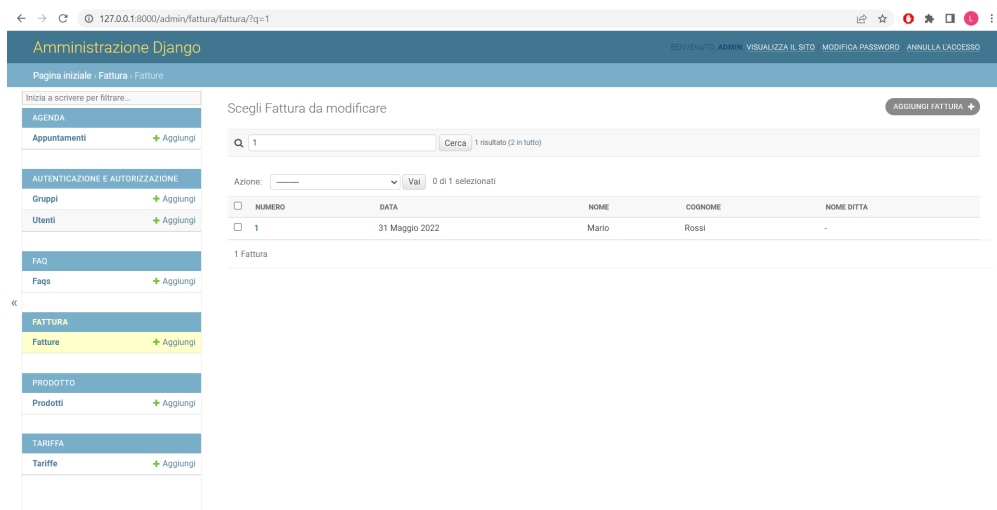


Figura 5.15: Mockup della schermata relativa alle fatture: ricerca di una fattura

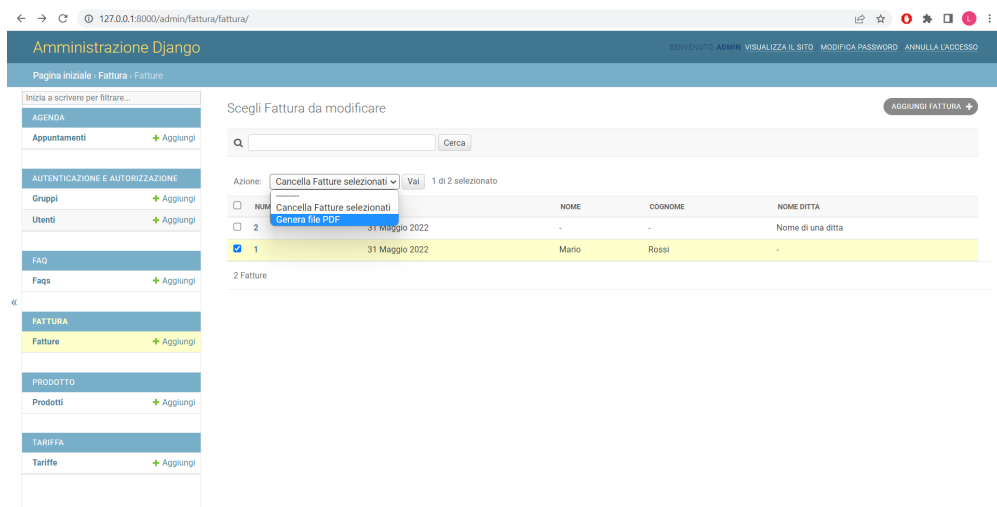


Figura 5.16: Mockup della schermata relativa alle fatture: generazione del file PDF

Luana Bibini
Via Giovanni Verga, 42 - 62100 Macerata (MC)
Codice Fiscale: BBNLNU66D70E783O - P.IVA: 02064010438
Telefono: +39 328 0541830
PEC: luanabibini@pec.opimacerata.it

Destinatario: Mario Rossi Via Scrima, 35 - 60126 Ancona Italia P.IVA: 11111111111	Fattura: 1 Tipo di documento: Fattura Data Fattura: 2022-05-31
---	---

Descrizione pagamento: BONIFICO BANCARIO	Banca d'appoggio: IT000000000000000000000012345
---	--

Descrizione	Quantità	Prezzo	IVA	Importo
PRODOTTO1	4	8.5	0%	34.0
PRODOTTO2	3	9.99	0%	29.97

	Imponibile	63.97
	Contributi (INPS) 4%	2.56
IVA 0% di 66.53 (art.10 n.18 DPR 633/1972)		0
	Totale EUR	66.53
	Fattura soggetta a bollo	2.00
	Da Pagare: (EUR)	68.53

Termini e Condizioni

Operazione esente IVA articolo 10, n.18, D.P.R. n.633/1972.

Imposta di bollo da 2 euro assolta sull'originale per importi maggiori di 77.47 euro.

Figura 5.17: Mockup della schermata relativa alle fatture: file PDF generato dalla fattura

5.1.6 Prodotto

All'interno di questa sezione è possibile inserire, modificare, ricercare oppure eliminare un prodotto, come mostrato nelle Figure 5.18 - 5.21.

Figura 5.18: Mockup della schermata relativa ai prodotti: inserimento/modifica di un prodotto

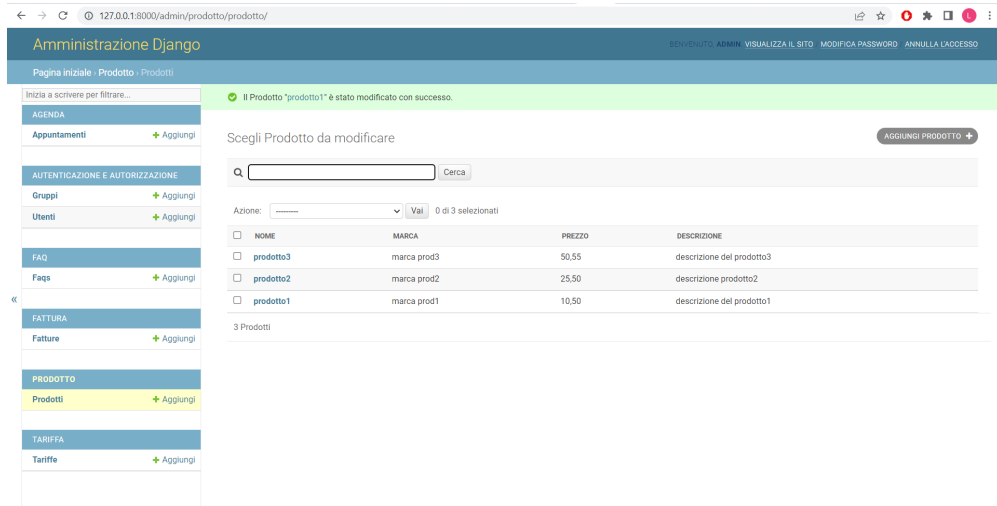


Figura 5.19: Mockup della schermata relativa ai prodotti: visualizzazione di un prodotto

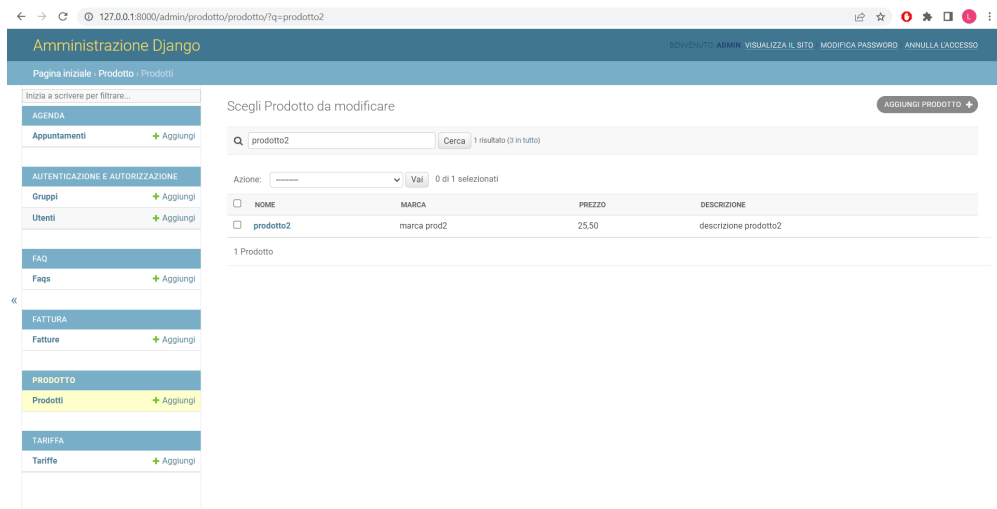


Figura 5.20: Mockup della schermata relativa ai prodotti: ricerca di un prodotto

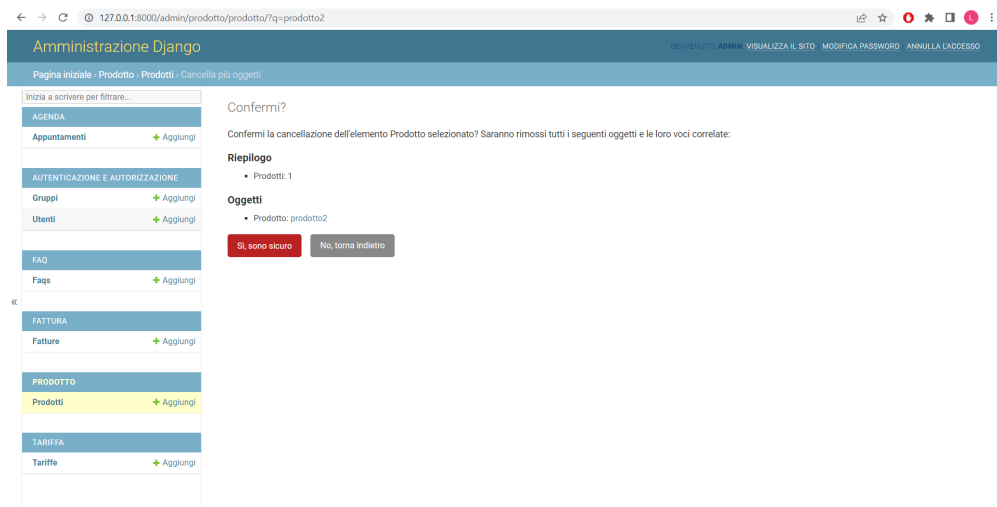


Figura 5.21: Mockup della schermata relativa ai prodotti: eliminazione di un prodotto

5.1.7 Tariffa

All'interno di questa sezione è possibile inserire, modificare, ricercare oppure eliminare una tariffa, come mostrato nelle Figure 5.22 - 5.25.

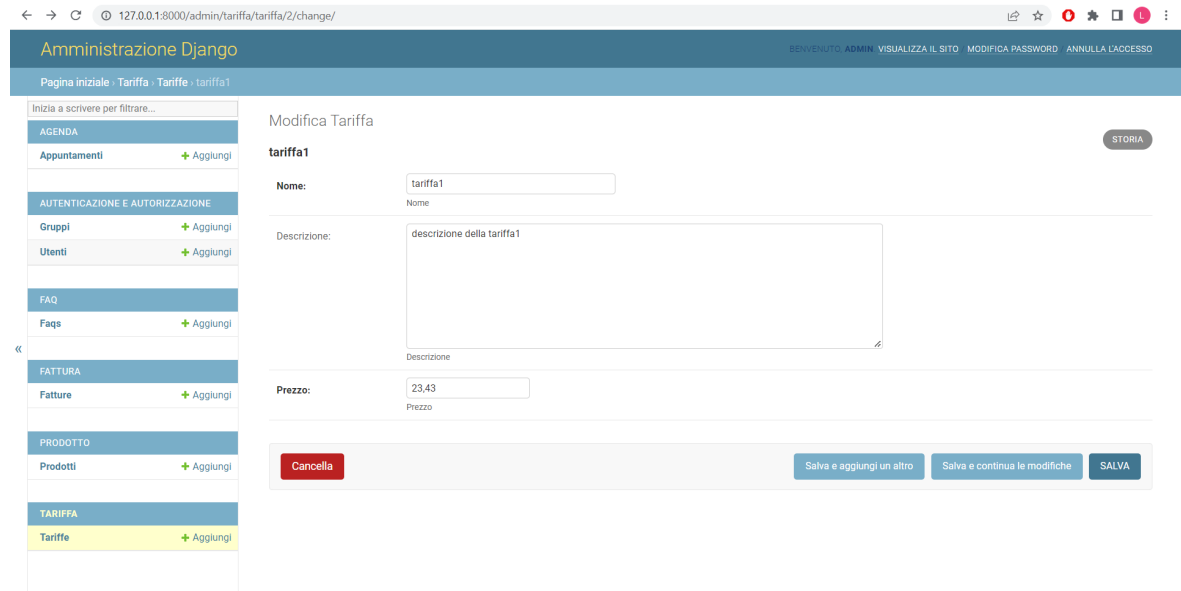


Figura 5.22: Mockup della schermata relativa alle tariffe: eliminazione di una tariffa

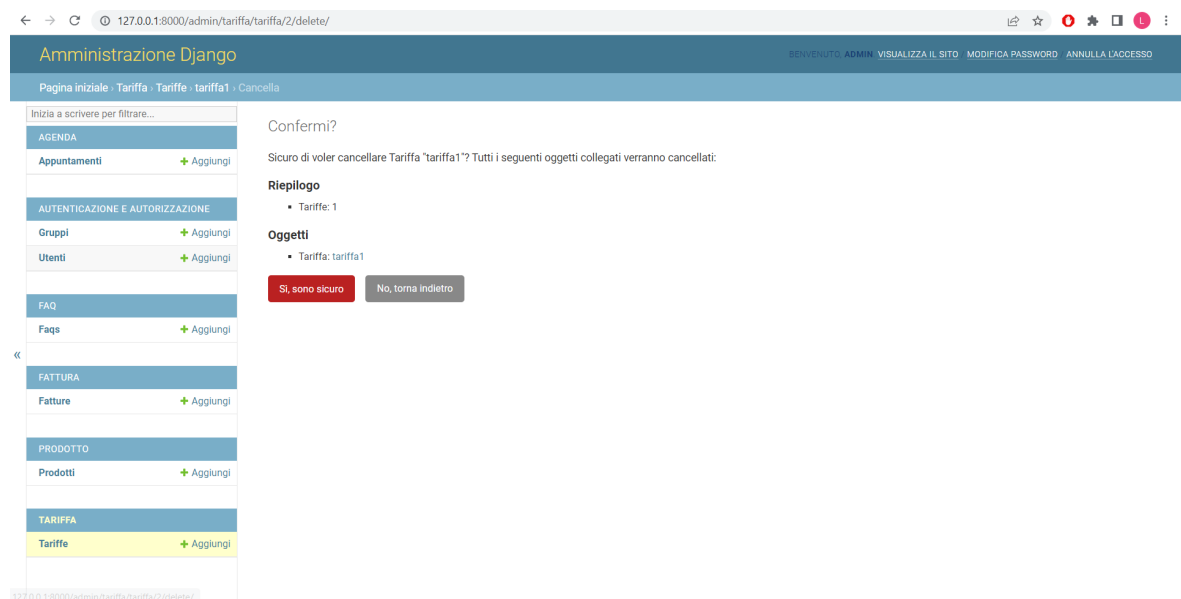


Figura 5.23: Mockup della schermata relativa alle tariffe: eliminazione di una tariffa

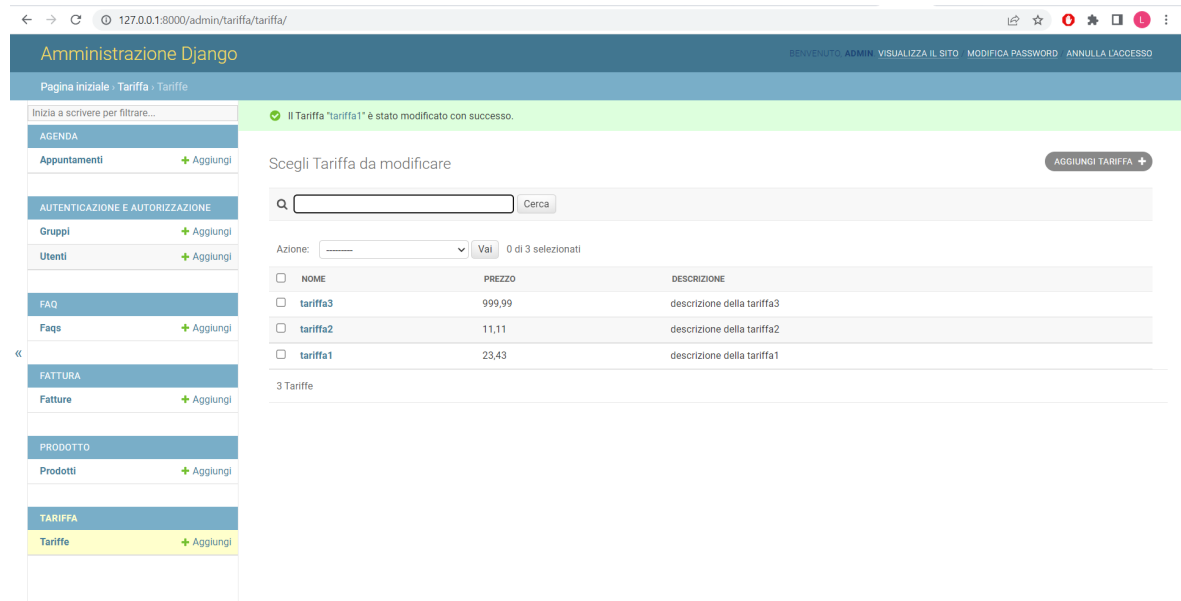


Figura 5.24: Mockup della schermata relativa alle tariffe: eliminazione di una tariffa

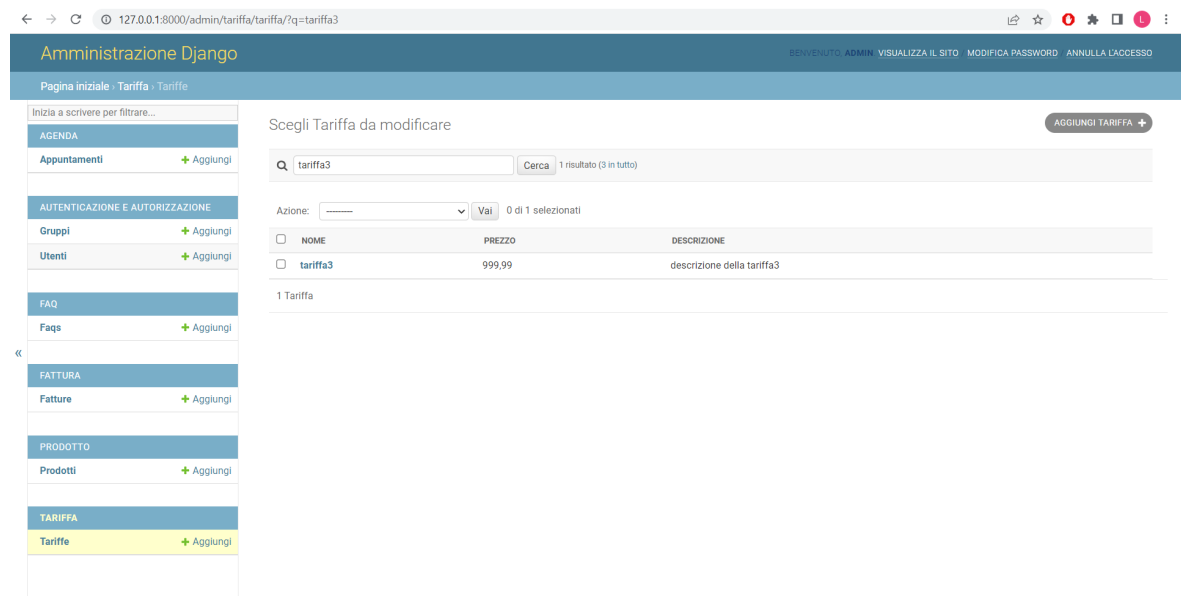


Figura 5.25: Mockup della schermata relativa alle tariffe: ricerca di una tariffa

5.1.8 Modifica Password

Per modificare la propria password è sufficiente cliccare sul pulsante in alto a destra. L'utente verrà reindirizzato alla pagina in Figura 5.26.

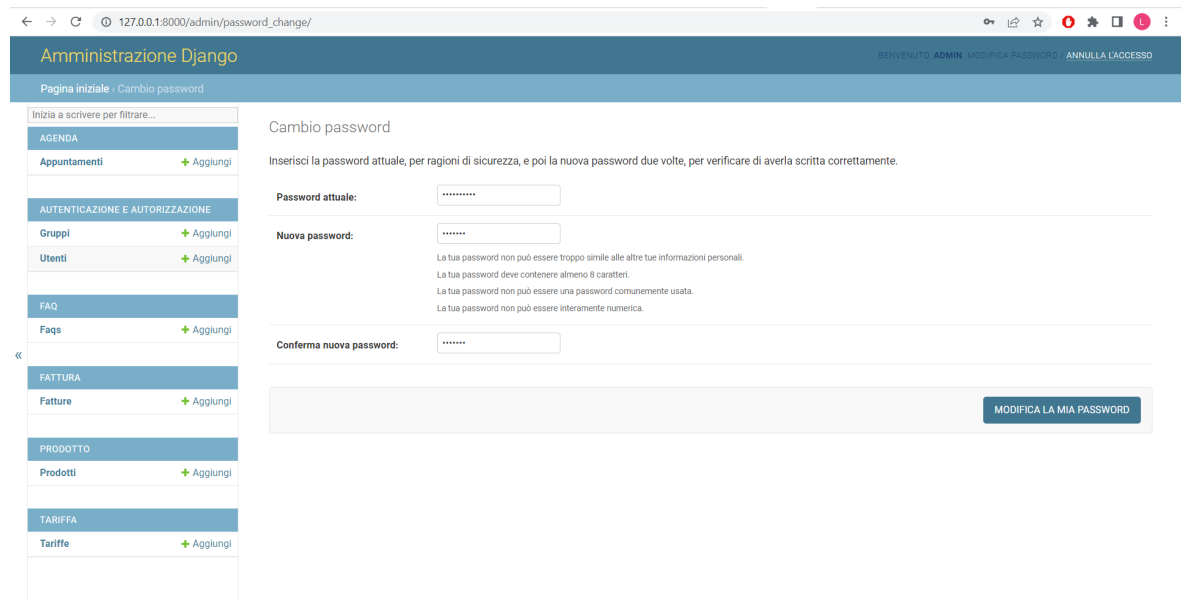


Figura 5.26: Mockup della schermata relativa alla modifica della password

5.1.9 Logout

Per effettuare il logout, è sufficiente cliccare sul pulsante in alto a destra. L'utente verrà reindirizzato alla pagina in Figura 5.27 e, una volta confermata la decisione di effettuare il logout, tornerà alla pagina di login.

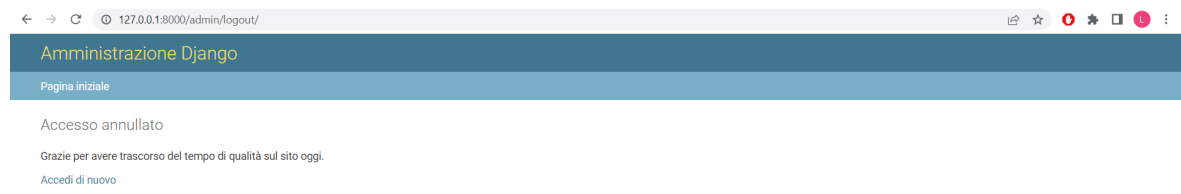


Figura 5.27: Mockup della schermata relativa al logout

5.2 Area Cliente

L'area dedicata al cliente è composta da quattro viste principali. Ogni vista possiede un Header ed un Footer, mostrati nelle Figure 5.28 e 5.29. In essi sono contenuti i riferimenti a

tutte le pagine, alcune informazioni relative all'amministratore e i riferimenti ai canali social dell'amministratore. Nel seguito analizzeremo ogni vista.

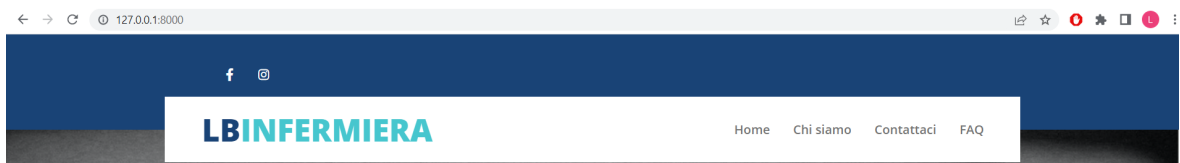


Figura 5.28: Mockup relativo all'header



Figura 5.29: Mockup relativo al footer

5.2.1 Home page

La *Home page* mostra informazioni riguardanti il lavoro svolto dall'amministratore; essa contiene alcuni link utili per ricondursi alle altre pagine, come mostrato nelle Figure 5.30 - 5.32.



Figura 5.30: Mockup relativo alla home page: slider iniziale

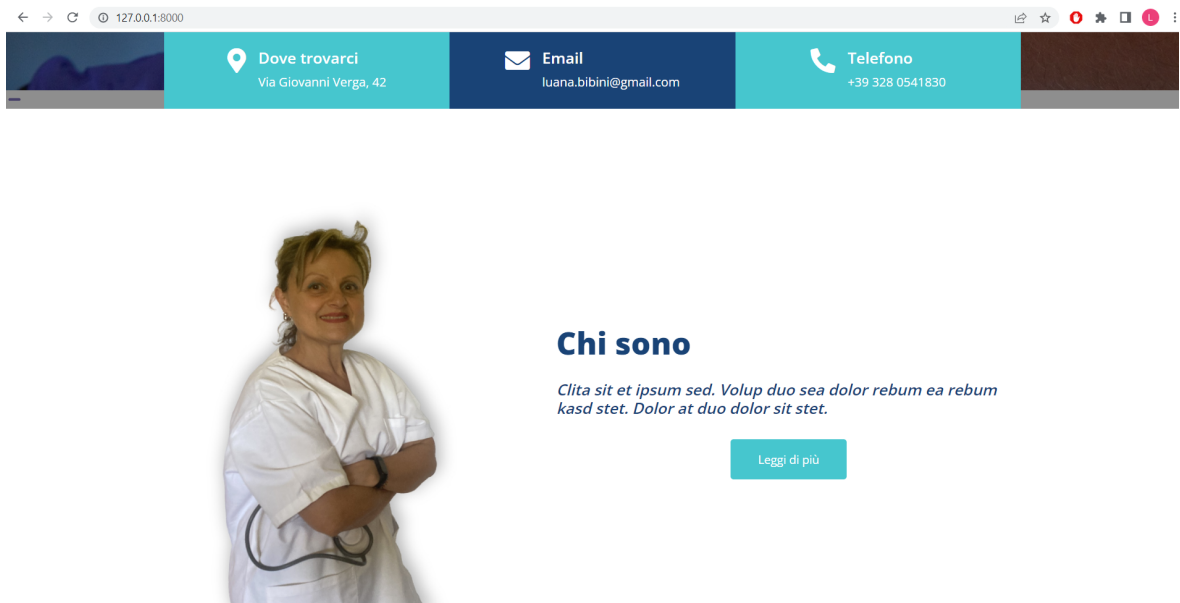


Figura 5.31: Mockup relativo alla home page: informazioni personali



Figura 5.32: Mockup relativo alla home page: informazioni riguardanti il lavoro

5.2.2 Contattaci

La pagina *Contattaci* è accessibile dalla rotta `contact/` e contiene le informazioni utili per contattare l'amministratore; inoltre, è presente una form che consente di inviare una richiesta all'amministratore tramite email. La pagina è mostrata nelle Figure 5.33 - 5.35.

Contattaci Home > Contattaci

Via Giovanni Verga, 42 luana.bibini@gmail.com +39 328 0541830

Via Giovanni Verga
62100 Macerata MC
Visualizza mappa più grande

Nome: Mario Cognome: Rossi
Città: Macerata Via: Rossini
Numero Civico: 8 Cap: 62100

Figura 5.33: Mockup della schermata relativa a *Contattaci*: compilazione form

Macerata Rossini
Numero Civico: 8 Cap: 62100
Codice Fiscale: BCDTR234LFK99KKK Email: mario.rossi@gmail.com

Richiesta:
richiesta di laurea

Data: 31/05/2022 Orario: 14:00
Possiedo il materiale necessario:
Possiedo la prescrizione del medico:

Invia

Figura 5.34: Mockup della schermata relativa a *Contattaci*: compilazione form

LBINFERMIERA Home Chi siamo Contattaci FAQ

Email inviata con successo

Contattaci Home > Contattaci

Via Giovanni Verga, 42 luana.bibini@gmail.com +39 328 0541830

Figura 5.35: Mockup della schermata relativa a *Contattaci*: successo nell'invio della form

5.2.3 Chi siamo

La sezione *Chi siamo* è accessibile dalla rotta `about/` e contiene le informazioni legate all'amministratore, come mostrato in Figura 5.36.

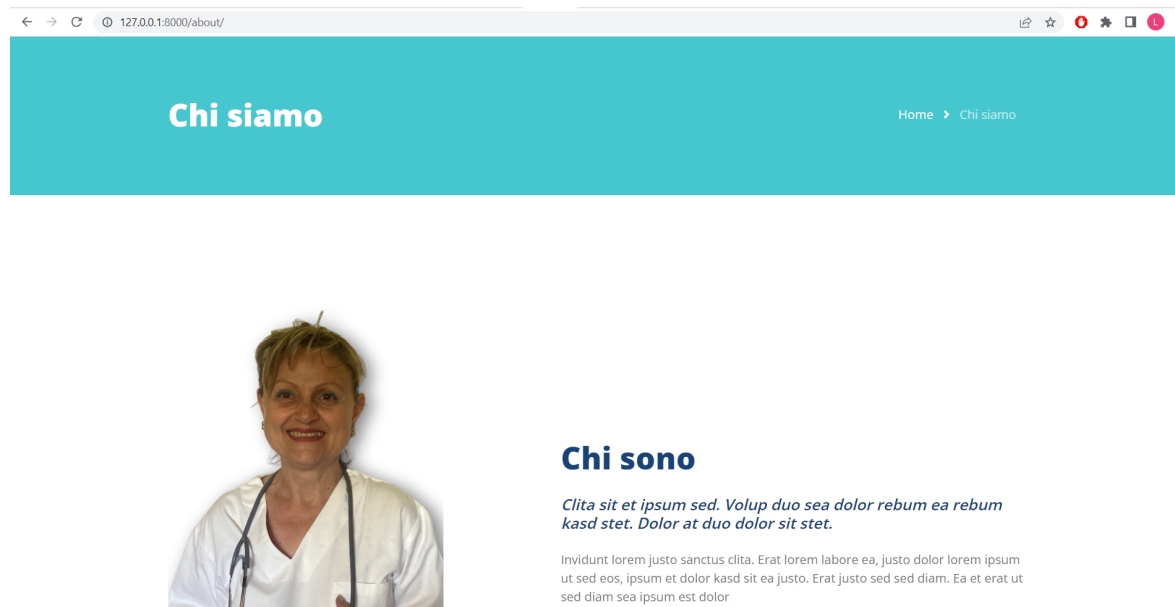


Figura 5.36: Mockup della schermata relativa a *Chi siamo*

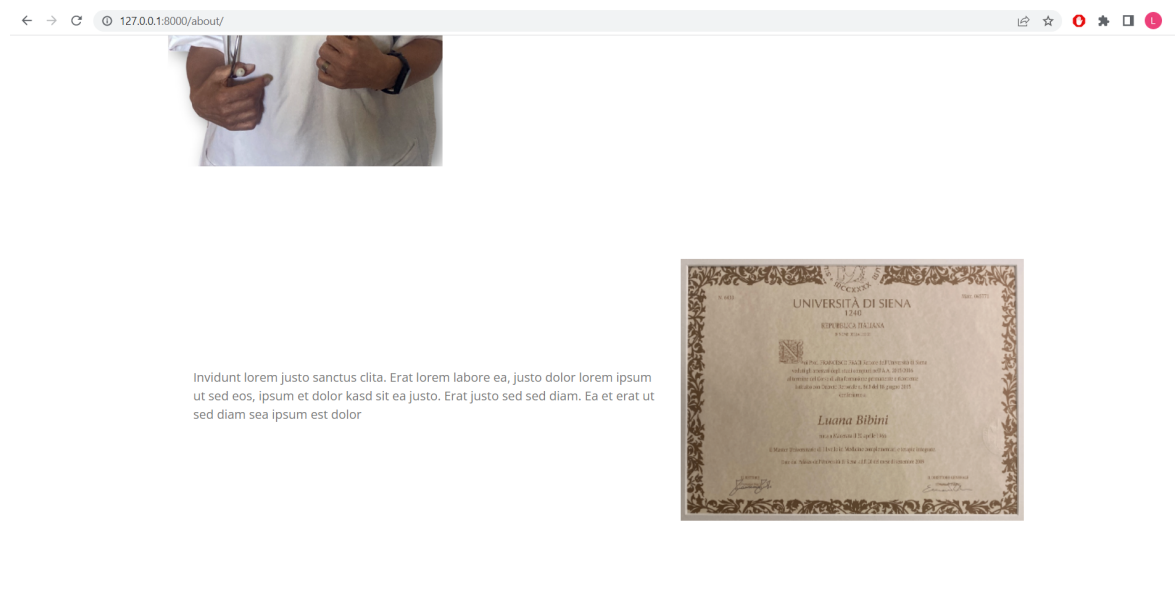


Figura 5.37: Mockup della schermata relativa a *Chi siamo*

5.2.4 FAQ

La pagina delle *FAQ* è accessibile attraverso la rotta `faq/` e mostra tutte le FAQ create dall'amministratore, come evidenziato in Figura 5.37.

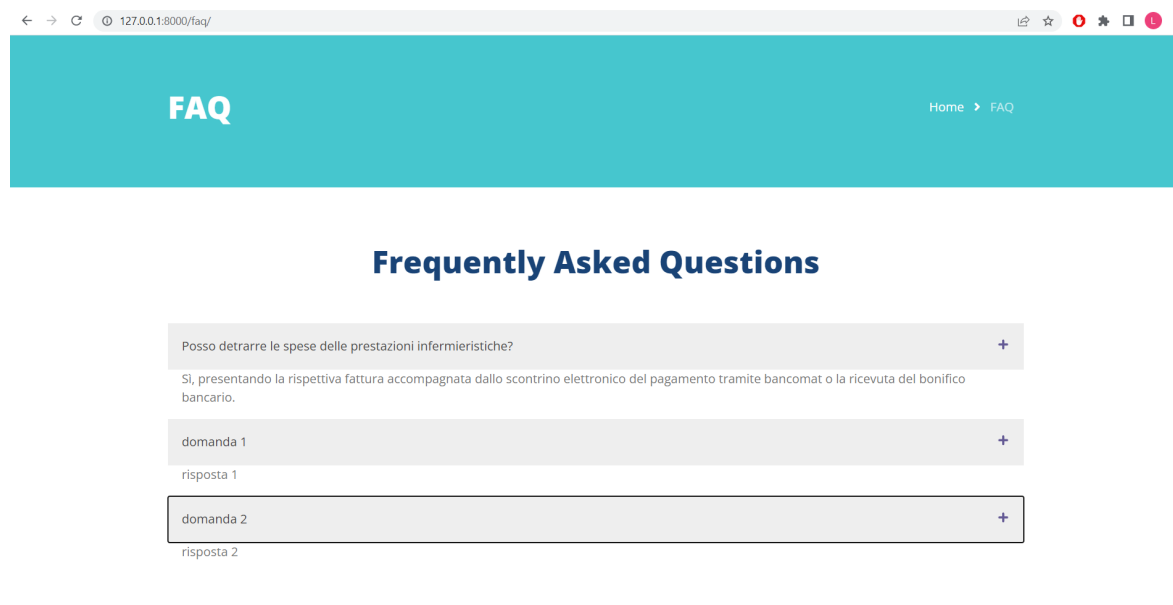


Figura 5.38: Mockup della schermata relativa alle *FAQ*

In questo capitolo analizzeremo il framework utilizzato per lo sviluppo del nostro software ed il relativo linguaggio; per ogni sezione analizzata nei precedenti capitoli mostreremo, poi, alcuni test di interesse. I test ci consentono di dimostrare che il nostro programma svolge i compiti per il quale è stato realizzato e ci consentono di identificare eventuali errori, per questo motivo sono di fondamentale importanza.

6.1 Framework e linguaggio utilizzati

In questo progetto è stato utilizzato Django, un framework composto da librerie scritte in Python, un linguaggio multiplatforma che sfrutta il paradigma orientato agli oggetti. Ne consegue, quindi, che sarà proprio Python il linguaggio di programmazione utilizzato per lo sviluppo del software.

Django utilizza il pattern MVC (Model-View-Controller), infatti, ogni applicazione che scriveremo sarà composta da un Model, che consente di rappresentare il modello dei dati, da una View, che svolge il ruolo del Controller e che gestirà la logica delle interazioni, e da un Template che svolge il ruolo della View e rappresenta la struttura e lo stile delle pagine che verranno visualizzate.

Una volta che abbiamo installato Django, possiamo creare un progetto lanciando il comando `django-admin.py startproject` seguito dal nome del nostro progetto.

L'applicazione principale contiene i parametri di configurazione e gli `url` gestiti dal progetto. Per ogni funzionalità creeremo un'applicazione specifica che sarà caratterizzata da file preimpostati nei quali inseriremo le classi e i metodi di interesse.

L'IDE di riferimento per lo sviluppo di questo software è VSCode. Il progetto è stato collegato ad una repository Github a cui manca il file `.env` contenente informazioni riservate, come la chiave segreta di progetto. Il codice di progetto è accessibile all'indirizzo URL `https://github.com/LorenzoLongarini/Progetto-di-ingegneria-del-software.git`.

6.2 Unit testing

Il testing in Django viene effettuato con l'utilizzo di `Unit tests`, che ci consentono di verificare il comportamento dei componenti, quali classi e funzioni. Ogni applicazione dispone della sua sezione dedicata ai test; per questo motivo, nelle prossime sezioni li analizzeremo nello specifico.

6.2.1 Test relativo all'agenda

In questa sezione esamineremo i test relativi all'agenda. In primo luogo definiremo la classe relativa ai test ed effettueremo il setup del modello in questione come mostrato in Figura 6.1.

```
class AppuntamentoTestCase(TestCase):
    @classmethod
    def setUpTestData(cls):
        Appuntamento.objects.create(id = 1, nome = "", day = "2022-05-31", orario_inizio = "18:00", orario_fine = "20:00", note = "")
```

Figura 6.1: Test di *Agenda*: creazione della classe e setup dei dati

Il primo test ci consente di verificare che l'etichetta relativa al nome dell'appuntamento sia corretta, come mostrato in Figura 6.2.

```
def test_nome_label(self):
    appuntamento = Appuntamento.objects.get(id=1)
    field_label = appuntamento._meta.get_field('nome').verbose_name
    self.assertEqual(field_label, 'Nome appuntamento')
```

Figura 6.2: Test di *Agenda*: etichetta relativa al nome

Il codice in Figura 6.3 ci consente di testare la funzione `check_overlap`, la quale verifica che non ci siano due appuntamenti creati nelle stesse fasce orarie. In questo caso, in base ai dati inseriti, ci aspettiamo un fallimento a conferma del suo corretto funzionamento.

```
#FAIL
def test_check_overlap(self):
    assert(Appuntamento.check_overlap(self, '18:00', '20:00', '14:00', '15:00', 2))
```

Figura 6.3: Test di *Agenda*: funzione `check_overlap`

Il codice in Figura 6.4 ci consente di testare la funzione `clean`, la quale verifica che l'orario del termine dell'appuntamento non sia antecedente a quello di inizio. Anche in questo caso, in base ai dati inseriti, ci aspettiamo un fallimento a conferma del suo corretto funzionamento.

```
#FAIL
def test_clean(self):
    appuntamento = Appuntamento.objects.create(id = 3, nome = "", day = "2022-05-31", orario_inizio = "18:00",
                                                orario_fine = "15:00", note = "")
    assert(Appuntamento.clean(appuntamento))
```

Figura 6.4: Test di *Agenda*: funzione `clean`

Infine, il codice in Figura 6.5 ci consente di testare la funzione `get_absolute_url`, la quale effettua un override dell'indirizzo URL nella pagina di amministratore per permetterci di manipolare i dati all'interno del layout.

```
def test_get_absolute_url(self):
    appuntamento = Appuntamento.objects.get(id=1)
    self.assertEqual(appuntamento.get_absolute_url(), '<a href="/admin/agenda/appuntamento/1/change/"> 18:00:00-20:00:00</a>')
```

Figura 6.5: Test di *Agenda*: funzione `get_absolute_url`

6.2.2 Test relativo alle FAQ

In questa sezione esamineremo i test relativi alle FAQ. Come prima cosa definiamo la classe relativa ai test e inseriamo il setup del modello in questione, come mostrato in Figura 6.6.

```
class FAQTestCase(TestCase):  
  
    @classmethod  
    def setUpTestData(cls):  
        FAQ.objects.create(nome = 'esempio1', domanda = "domanda1", risposta= "risposta1")
```

Figura 6.6: Test di FAQ: creazione della classe e setup dei dati

Il test in Figura 6.7, ci consente di verificare che l’etichetta relativa al nome della FAQ sia scritta correttamente.

```
def test_nome_label(self):  
    faq = FAQ.objects.get(id=1)  
    field_label = faq._meta.get_field('nome').verbose_name  
    self.assertEqual(field_label, 'nome')
```

Figura 6.7: Test di FAQ: etichetta relativa al nome della FAQ

In Figura 6.8 viene mostrato il test che verifica l’effettiva lunghezza massima delle risposte delle FAQ.

```
#FAIL  
def test_risposta_max_length(self):  
    faq = FAQ.objects.get(id=1)  
    max_length = faq._meta.get_field('risposta').max_length  
    self.assertEqual(max_length, 300)
```

Figura 6.8: Test di FAQ: verifica del max_length

Infine, in Figura 6.9 viene mostrato il test che verifica che la rotta relativa alla pagina delle FAQ, nella vista cliente, sia corretta.

```
def test_faqqpage(self):  
    response = self.client.get('/faq/')  
    self.assertEqual(response.status_code, 200)
```

Figura 6.9: Test di FAQ: rotta /faq/

6.2.3 Test relativo alle fatture

In questa sezione esamineremo i test relativi alle fatture. In primo luogo definiamo la classe relativa ai test ed inseriamo il setup del modello in questione come mostrato in Figura 6.10.

```
class FatturaTestCase(TestCase):

    @classmethod
    def setUpTestData(cls):
        Fattura.objects.create(id=1, nome="Mario", cognome="Rossi", nome_ditta="", citta="Macerata", via="Rossini", num_civico=8,
                               cap=62100, partita_iva=1111, cod_fiscale="BBBBB7865YFYF", cod_cliente="1234567", numero=1,
                               data = "2022-05-31", descrizione_pag = "descrizione", banca = "IT00000123")
        TariffaFatt.objects.create(descrizione_prod="descrizione1", quantita=3, prezzo=10.50, fatt=Fattura.objects.get(id=1))
```

Figura 6.10: Test di *Fattura*: creazione della classe e setup dei dati

Il primo test che effettuiamo riguarda l’etichetta del codice cliente. Il test fallirà e a riga di comando verrà restituito il valore corretto, ossia `Codice Cliente`. Il test è mostrato in Figura 6.11.

```
#FAIL
def test_cod_cliente_label(self):
    fattura = Fattura.objects.get(id=1)
    field_label = fattura._meta.get_field('cod_cliente').verbose_name
    self.assertEqual(field_label, 'cod_cliente')
```

Figura 6.11: Test di *Fattura*: etichetta relativa al codice cliente

Questo test verifica l’etichetta relativa alla quantità nelle potenziali tariffe (o prodotti) che possono essere inserite all’interno della fattura. Il test è mostrato in Figura 6.12.

```
def test_quantita_label(self):
    fatt_prod = TariffaFatt.objects.get(id=1)
    field_label = fatt_prod._meta.get_field('quantita').verbose_name
    self.assertEqual(field_label, 'Quantità')
```

Figura 6.12: Test di *Fattura*: etichetta relativa alla quantità

Questo test verifica che il numero di caratteri del codice fiscale non sia superiore a 16, come mostrato in Figura 6.13.

```
def test_cod_fiscale(self):
    fattura = Fattura.objects.get(id=1)
    max_length = fattura._meta.get_field('cod_fiscale').max_length
    self.assertEqual(max_length, 16)
```

Figura 6.13: Test di *Fattura*: verifica del `max_length`

Il codice in Figura 6.14 testa che la funzione `validate_price`, la quale verifica che un prezzo inserito non sia negativo, operi correttamente; al test viene passato come argomento il prezzo dell’oggetto definito nel setup.

```
def test_validate_price(self):
    fattura = Fattura.objects.get(id=1)
    prezzo = fattura.prezzo
    assert(models.validate_price(prezzo), True)
```

Figura 6.14: Test di *Fattura*: funzione `validate_price`

6.2.4 Test relativo alle prenotazioni

I test relativi alla prenotazione non necessitano un setup, come nei casi precedenti, poichè il modello relativo è una form.

Il primo test è mostrato in Figura 6.15 e verifica che l'etichetta relativa alla prescrizione sia corretta.

```
class PrenotazioneTestCase(TestCase):  
  
    def test_prescrizione_label(self):  
        form = Prenotazione()  
        self.assertTrue(form.fields['prescrizione'].label is None or  
                        form.fields['prescrizione'].label == 'Possiedo la prescrizione del medico:')
```

Figura 6.15: Test di *Prenotazione*: etichetta relativa alla prescrizione

Il test in Figura 6.16 verifica che la rotta che consente di visualizzare la pagina al cliente sia corretta.

```
def test_prenotazionepage(self):  
    response = self.client.get('/contact/')  
    self.assertEqual(response.status_code, 200)
```

Figura 6.16: Test di *Prenotazione*: rotta /contact/

Infine, il codice in Figura 6.17 testa la funzione `clean_data`, la quale controlla che la data inserita nella form non sia antecedente alla data odierna. Inserendo come valore una data passata, il test fallirà, a conferma del corretto funzionamento della funzione.

```
#FAIL  
def test_clean_data(self):  
    assert(form.clean_data('2022-01-24'))
```

Figura 6.17: Test di *Prenotazione*: funzione `clean_data`

6.2.5 Test relativo al prodotto

In questa sezione esamineremo i test relativi ai prodotti. In primo luogo definiamo la classe relativa ai test ed inseriamo il setup del modello in questione come mostrato in Figura 6.18.

```
class ProdottoTestCase(TestCase):  
  
    @classmethod  
    def setUpTestData(cls):  
        Prodotto.objects.create(nome = 'prodotto1', descrizione = "descrizione generica", marca= "marca1", prezzo = -3)
```

Figura 6.18: Test di *Prodotto*: creazione della classe e setup dei dati

Il test in Figura 6.19 verifica che l'etichetta relativa alla marca venga scritta correttamente. Il test fallirà, mostrando a riga di comando l'etichetta corretta.

```
#FAIL
def test_marca_label(self):
    prodotto = Prodotto.objects.get(id=1)
    field_label = prodotto._meta.get_field('marca').verbose_name
    self.assertEqual(field_label, 'marca')
```

Figura 6.19: Test di *Prodotto*: etichetta relativa alla marca

In Figura 6.20 viene testata la funzione `validate_price` presente nel prodotto, la quale verifica che il prezzo non assuma valori negativi.

```
def test_validate_price(self):
    prodotto = Prodotto.objects.get(id=1)
    prezzo = prodotto.prezzo
    assert(models.validate_price(prezzo), False)
```

Figura 6.20: Test di *Prodotto*: funzione `validate_price`

Infine, in Figura 6.21, viene testata la funzione `get_absolute_url`, la quale effettua un override dell'indirizzo URL nella pagina di amministratore per permetterci di manipolare i dati all'interno del layout.

```
def test_get_absolute_url(self):
    prodotto = Prodotto.objects.get(id=1)
    self.assertEqual(prodotto.get_absolute_url(), '<a href="/admin/prodotto/prodotto/1/change/">prodotto1</a>')
```

Figura 6.21: Test di *Prodotto*: funzione `get_absolute_url`

6.2.6 Test relativo alle tariffe

In questa sezione esamineremo i test relativi alle tariffe. In primo luogo definiamo la classe relativa ai test ed inseriamo il setup del modello in questione come mostrato in Figura 6.22.

```
class TariffaTestCase(TestCase):
    @classmethod
    def setUpTestData(cls):
        Tariffa.objects.create(nome = 'tariffa1', descrizione = "descrizione", prezzo= 9.99)
```

Figura 6.22: Test di *Tariffa*: creazione della classe e setup dei dati

Il test in Figura 6.23 verifica che l'etichetta relativa alla descrizione sia scritta correttamente.

```
def test_descrizione_label(self):
    tariffa = Tariffa.objects.get(id=1)
    field_label = tariffa._meta.get_field('descrizione').verbose_name
    self.assertEqual(field_label, 'Descrizione')
```

Figura 6.23: Test di *Tariffa*: etichetta relativa alla descrizione

In Figura 6.24, il test verifica che il numero massimo di caratteri per la descrizione non superi 300.

```
def test_descrizione_max_length(self):
    tariffa = Tariffa.objects.get(id=1)
    max_length = tariffa._meta.get_field('descrizione').max_length
    self.assertEqual(max_length, 300)
```

Figura 6.24: Test di *Tariffa*: verifica del `max_length`

Infine, il codice in Figura 6.25 ci consente di testare la funzione `get_absolute_url`, che, come nei casi precedenti effettua un override dell'indirizzo URL nella pagina di amministratore per permetterci di modificare il layout.

```
#FAIL
def test_get_absolute_url(self):
    tariffa = Tariffa.objects.get(id=1)
    self.assertEqual(tariffa.get_absolute_url(), 'href="/admin/prodotto/prodotto/1/change/')
```

Figura 6.25: Test di *Tariffa*: funzione `get_absolute_url`

Conclusioni e sviluppi futuri

Nei precedenti capitoli abbiamo sviluppato un software in grado di semplificare le operazioni lavorative di un'infermiera libera professionista. Siamo partiti da un'intervista alla diretta interessata, dalla quale poi abbiamo ricavato i requisiti. Avendo definito, dunque, cosa doveva garantire il nostro programma, abbiamo potuto determinare le classi di interesse e gli scenari, per poi modellare le viste con i mockup. Una volta implementato il codice, lo abbiamo testato per verificare che funzionasse correttamente.

I punti di forza da noi individuati per il nostro software sono i seguenti:

- La completezza con cui il lavoro è stato svolto e documentato. In primo luogo, il software consente di avere una visione completa di ciò che è stato fatto e, in secondo luogo, ne facilita una potenziale modifica futura.
- Una grande semplicità ed intuitività per l'utilizzatore, come previsto nei requisiti. Il software consente, dopo qualche utilizzo, di avere subito una grande familiarità con esso e una conseguente velocità nello svolgere le operazioni.
- Una grande modularità delle classi, dovuta al pattern MVC che ci consente di modellare i dati e la logica di qualsiasi componente senza dover compromettere la struttura generale.
- La modularità del template, che facilita la modifica del design senza dover ricostruire da zero nessun componente.
- Una grande sicurezza relativa ai dati personali e ai dati del progetto, grazie all'utilizzo del file `.env`.

Oltre agli aspetti positivi, è bene mettere in evidenza alcuni punti di debolezza che sono presenti; questi riguardano, in particolare:

- La scarsa personalizzazione del lato amministratore, poichè frutto di file generati direttamente da Django.
- La mancanza di un livello di utenza per il cliente, che avrebbe semplificato la memorizzazione di alcuni dati per l'amministratore.

In conclusione è, inoltre, importante mettere in risalto i possibili sviluppi futuri applicabili al nostro software; tra questi citiamo:

- La possibilità di implementare la generazione della fattura in XML, il formato richiesto dall'agenzia delle entrate.

- La possibilità di sviluppare, nelle sezioni di prodotto e tariffa, la funzione di generazione del PDF per averne una versione cartacea usufruibile.
- La possibilità di creare un livello di utenza relativa al cliente, che potrebbe facilitare l'amministratore nella memorizzazione di alcuni dati.

Gli sviluppi futuri potrebbero dipendere da eventuali richieste dell'infermiera, dovute a nuove esigenze pratiche del proprio lavoro. Risulta, quindi, opportuno consultare la diretta interessata prima di procedere con l'implementazione di nuove funzionalità.

- BOSCAINI, M. (2020), *Imparare a programmare in Python*, Apogeo.
- BUTTU, M. (2014), *Programmare con Python, Guida completa*, Edizioni Lswr.
- C. GHEZZI, D. M., M. JAZAYERI (2021), *Ingegneria del software, Fondamenti e principi*, Pearson.
- GASTON, P. (2011), *CSS3, Guida completa per lo sviluppatore*, Hoepli.
- GIGLIOTTI, G. (2011), *HTML5 e CSS3*, Apogeo.
- H. BERSINI, G. D., P. ALEXIS (2015), *Apprendere la programmazione web Python&Django*, Eyrolles.
- HAVERBEKE, M. (2019), *Javascript, Guida completa per lo sviluppatore*, Hoepli.
- MELÈ, A. (2020), *Django 3 By Example*, Packt.
- P. ATZENI, S. C. (2018), *Basi di dati*, McGraw-Hill.
- P. DEITEL, H. D. (2020), *Introduzione a Python, Per l'informatica e la data science*, Pearson.
- P. FOGGIA, M. V. (2011), *Algoritmi e strutture dati*, McGraw-Hill.
- PRESSMAN, R. S. (2008), *Principi di Ingegneria del software*, McGraw-Hill Education.
- RUBIO, D. (2017), *Beginning Django, Web Application Development and Deployment with Python*, Apress.
- SHVETS, A. (2019), *Dive into Design Patterns*, Ebook.
- SOMMERVILLE, I. (2017), *Ingegneria del Software*, Pearson Education.

Websites consulted

- Django, the web framework for perfectionists with deadlines - <https://www.djangoproject.com/>

- Real Python - <https://realpython.com/>
- HTML.it - <https://www.html.it/>
- StackOverflow - <https://stackoverflow.com/>
- Python - <https://www.python.org/>
- W3School <https://www.w3schools.com/>
- OrdinaryCoders - <https://ordinarycoders.com/>
- HTML - <https://www.w3.org/Style/CSS/>
- W3 - <https://html.spec.whatwg.org/multipage/>
- Mdn web docs -
<https://developer.mozilla.org/en-US/docs/Web/JavaScript/>
- Github - <https://github.com/>

Ringraziamenti

Vorrei dedicare questo spazio alle persone che, con il loro supporto, mi hanno aiutato in questo meraviglioso percorso di acquisizione delle conoscenze durante gli anni universitari.

Un ringraziamento particolare va al mio relatore Ursino Domenico, che mi ha seguito, con la sua infinita disponibilità, in ogni step della realizzazione dell'elaborato.

Grazie per aver accresciuto le mie conoscenze, le mie competenze ed avermi insegnato un metodo di lavoro che sicuramente replicherò in futuro.

Non posso non menzionare la mia famiglia che, da sempre, mi sostiene nella realizzazione dei miei progetti, ed, in particolare, mia sorella Alice per avermi aiutato con lo studio.

Non finirò mai di ringraziarvi per avermi permesso di arrivare fin qui.

Soldi non ne ho, quindi spero che la mia tesi possa bastarvi.

Grazie ai miei amici Michele, Leonardo, Andrea, Luigi, Luca e Leonardo, per essere stati sempre presenti durante il mio percorso di studi. Grazie per aver ascoltato i miei sfoghi, per avermi dato consigli, per avermi incoraggiato e motivato ad andare sempre avanti. Siete come fratelli per me.

Grazie ai miei amici e coinquilini Michael e Simone, per non avermi fatto sentire inadeguato, per avermi insegnato a cucinare, per avermi fatto scoprire cose nuove.

Grazie alla mia collega e compagna di progetto Chiara, per esserci stata dal primo giorno, e alla mia fidanzata e collega Margherita, per avermi aiutato anche in quest'ultima parte del mio percorso. Grazie per aver creduto in me.

Grazie ai miei amici e colleghi Alex, Michele, Diego e Gianmatteo, per aver condiviso con me questa esperienza; grazie per le risate, grazie per la tranquillità e la serenità che mi avete trasmesso. Un ringraziamento speciale ad Alex, per avermi trasmesso la sua motivazione e tenacia.

Infine, dedico questa tesi a me stesso, ai miei sacrifici e alla mia perseveranza, per non aver mollato anche quando sembrava la cosa più semplice da fare.