UNIVERSITÀ POLITECNICA DELLE MARCHE
FACOLTÀ DI INGEGNERIA
Dipartimento di Ingegneria dell'Informazione
Corso di Laurea in Ingegneria Informatica e dell'Automazione

TESI DI LAUREA

Progettazione e implementazione di un sistema informativo per la gestione di una società di rugby

Design and implementation of an information system for the management of a rugby club

Relatore

Prof. Domenico Ursino

Candidato

Leonardo Bordoni

ANNO ACCADEMICO 2022-2023

La potenza è nulla senza il controllo

Ronaldo ″il Fenomeno″ Luís Nazário de Lima

Sommario

Al giorno d'oggi nessuna società o azienda può esimersi dall'utilizzare un sistema gestionale informatizzato per controllare ed organizzare i dati di cui usufruisce nel quotidiano. La gestione dei dati è la base per poter costruire una società robusta ed efficiente, a prescindere dal contesto in cui essa opera. In questa tesi vengono descritti la progettazione e la creazione di un sitema gestionale per la società Rugby Falconara Dinamis ASD. In particolare, per realizzare questa attività è stata eseguita una prima fase di raccolta e analisi dei requisiti; poi si è progettata una base di dati opportuna. A seguire, si è progettata l'applicazione usando diversi diagrammi e UML. Infine, il sistema è stato implementato ed è stata effettuata un'analisi qualitativa dello stesso.

Keyword: Sistema Gestionale Sportivo, Analisi dei Requisiti, Diagrammi UML, Diagrammi ER, Architettura Software, Interfaccia Utente

Indice

Introduzione

Ad oggi ogni tipo di associazione, sia lavorativa che non, possiede una grande mole di dati che, oltre a necessitare di uno spazio fisico in cui risiedere, hanno bisogno di essere organizzati in una qualche modalità. Una volta che i dati vengono archiviati e sistemati possono essere usati per molteplici scopi, ad esempio, per effettuare scelte consapevoli, organizzare lavori futuri, o, semplicemente, tenere sotto controllo i vari aspetti dell'associazione.

Non è possibile definire un sistema gestionale come una tecnologia dedicata solo ad un certo ambito, in quanto esso è applicabile in più contesti. Sicuramente nel campo lavorativo, un gestionale completo ed efficiente fornisce un vantaggio importante nei confronti dei competitor su più fronti, tanto che oramai la maggior parte delle aziende, grandi o piccole che siano, hanno a disposizione ed investono in sistemi gestionali che possano agevolare il lavoro umano e, in certi casi, sostituirlo.

Tuttavia possiamo applicare le tecnologie di un gestionale anche in altri contesti, come, ad esempio, in un associazione non a scopo di lucro, come una società sportiva o una società a scopo benefico. Ognuna di queste si interessa di diversi settori, focalizzandosi su vari aspetti; dietro alle attività principali, ci sono sempre, però, delle figure che si occupano dell'organizzazione e della gestione dell'organizzazione. Avere un sistema articolato di gestione dei dati, qualunque essi siano, che comprenda, eventualmente, una loro analisi, fornisce un supporto importante nell'alleggerire i carichi di lavoro e rendere più agevole il controllo dei vari settori, sopratutto nei casi di grandi società. Dunque, è impossibile pensare di poter fare tutto ciò senza dotarsi di un programma digitale che supporti questa mole di lavoro; oramai è impensabile trovare una struttura che svolga queste operazioni di gestione senza utilizzare un software, piuttosto che un'app mobile o un sito web.

Nel caso di una società sportiva, un ruolo centrale lo assume la responsabilità del club nella gestione dei documenti e dei dati anagrafici di ogni iscritto; sia per una questione di privacy che per l'immagine della società, non gestire correttamente questo tipo di dati o, peggio, richiedere ripetutamente ai diretti interessati i dati necessari alle operazioni quotidiane è, certamente, un grande problema, oltre che un lavoro aggiuntivo sulle spalle di chi si occupa di questi settori. Risulta fondamentale essere organizzati e saper gestire questi dati delicati in maniera opportuna. Inoltre, una società di questo tipo avrà a disposizione molte attrezzature per effettuare allenamenti e partite; non tenere sotto controllo questo materiale porterebbe a molti problemi, come, ad esempio, acquisti superflui o confusione nel magazzino. Anche qui, avere un metodo o, meglio, uno strumento per la gestione di queste problematiche risulterebbe di grande aiuto sia per le piccole che le grandi società. Si potrebbe continuare descrivendo molte altre motivazioni per cui un'azienda, di qualunque ambito essa si occupi, dovrebbe adottare un gestionale per controllare i dati più disparati; ci si limita alle spiegazioni

già date in quanto risultano esaustive e provano questa pratica che è, anche, dettagliatamente documentata online.

Il sistema gestionale oggetto della presente tesi è dedicato alla società sportiva Rugby Falconara Dinamis ASD. Il sistema si compone di una schermata inziale che permette l'accesso degli utenti al gestionale tramite credenziali. Ogni utente, a seconda del suo ruolo nella società, avrà a disposizione funzionalità diverse.

Una volta autenticati si entra nell'area funzionale del software che, a sua volta, si compone di tre macro sezioni.

La prima, che viene subito visualizzata sullo schermo, è dedicata agli iscritti; in questa parte l'utente potrà accedere alle schede di tutti gli iscritti, divisi per categoria di appartenenza, e potrà inserire nuovi dati, modificare quelli già esistenti o rimuoverli.

La seconda parte, invece, è dedicata all'inventario. Qui l'utente potrà gestire tutte le attrezzature sportive, e non solo, divise, come in precedenza, per le categorie di appartenenza. Per ogni attrezzo si potranno inserire vari attributi, come quantità o prezzo di acquisto; chiaramente, tali attributi, potranno essere modificati o rimossi. In questa sezione, una parte è dedicata anche al merchandising della società; in questa si potranno inserire degli oggetti dedicati alla vendita dovendo, però, aggiungere, degli attributi specifici rispetto all'attrezzatura sportiva, come, ad esempio, il link del sito per poter acquistare la merce.

L'ultima sezione, invece, è dedicata agli eventi delle categorie della società; si potranno inserire gli eventi, classificati in allenamenti o partite, e, di questi, si potranno inserire delle valutazioni tecniche tramite gli appositi indicatori. Per ogni partita è, poi, possibile inserire una formazione i cui componenti vengono suggeriti in automatico dal sistema a seconda della categoria che effettua la gara.

Sulla destra dello schermo è presente, e rimane fissa, una colonna che indica tutti gli iscritti aventi i certificati medici scaduti o assenti; l'utente potrà decidere o meno se visualizzare, e dunque rimuovere, gli avvisi. Per ogni sezione sono disponibili delle funzionalità specifiche di ordinamento degli elementi ed, inoltre, è possibile utilizzare l'apposita barra di ricerca per navigare nel gestionale utilizzando, però, opportune parole chiave identificative.

Di seguito viene fornito un elenco dei capitoli contenuti nella tesi:

- *I sistemi gestionali*: viene fornita una descrizione dettagliata di questi sistemi, esaminandone vantaggi pratici e complessità nella loro implementazione.
- *Specifica e analisi dei requsiti*: vengono descritte le fasi che hanno portato alle specifiche richieste dalla società di rugby e viene fatta l'analisi dei requisiti del software, derivanti da esse.
- *Progettazione del database sottostante*: vengono descritti tutti i passaggi di analisi e di creazione del database che supporta i dati del programma, a partire dal primo schema base fino all'implementazione vera e propria.
- *Progettazione della componente applicativa*: questo è il capitolo centrale della trattazione; vengono forniti dettagli riguardo il programma dal punto di vista architetturale, di desgin e funzionale.
- *Implementazione e manuale utente*: vengono descritti gli strumenti che sono stati utli all'implementazione del programma, inoltre, viene fornito un manuale utente come guida pratica per l'utilizzo del gestionale.
- *Discussione in merito al lavoro svolto*: viene fatta un'analisi qualitativa del software usando una tecnica di analisi aziendale e viene comparato il gestionale creato con altri disponibili sul mercato.

CAPITOLO 1

I sistemi gestionali

In questo primo capitolo viene fornita una descrizione generale dei i sistemi gestionali. Una volta introdotti i vari tipi esistenti, si pone il focus sui vantaggi che una tecnologia simile può conferire, si eseguirà una veloce panoramica su questo aspetto toccando vari lati di un'azienda, partendo da quello pratico e lavorativo, per arrivare poi a quello più umano. Verranno,poi,elencate le caratteristiche principali che un gestionale possiede solitamente, per poi definirne, nel dettaglio, il ruolo che può occupare, una volta messo a regime. Infine, dopo aver discusso di eventuali problemi che possono sorgere progettando e usando un gestionale, si concluderà parlando di come effettivamente, al giorno d'oggi, il concetto di "sistema gestionale" e "informatizzazione e digitalizzazione", inevitabilmente, si intrecciano.

1.1 Introduzione ai sistemi gestionali

Un sistema gestionale è un insieme di tecnologie, processi e persone che lavorano insieme per supportare le attività di un'azienda. I sistemi gestionali possono aiutare le aziende a migliorare la loro efficienza, produttività e competitività.

Esistono diversi tipi di sistemi gestionali, ognuno dei quali è progettato per supportare un particolare tipo di attività aziendale. Alcuni dei tipi più comuni di sistemi gestionali includono:

- sistemi di contabilità;
- sistemi di gestione delle relazioni con i clienti (CRM);
- sistemi di gestione della catena di fornitura (SCM);
- sistemi di gestione dei progetti (PMS);
- sistemi di gestione dei contenuti (CMS);

Come anche il nome lascia intuire, di solito, le aziende si affidano a sistemi gestionali per avere una maggiore organizzazione in diversi ambiti. Si tratta di un investimento, che spesso può essere importante, fatto per avere dei vantaggi organizzativi importanti che verranno elencati in seguito.

Questi sistemi gestionali molto spesso vengono completamente digitalizzati e l'azienda stessa si appogia su un'azienda specializzata per il loro monitoraggio, la loro gestione ed eventualmente, la loro analisi, a seconda dell'utilizzo che se ne fa.

Al giorno d'oggi non esiste azienda affermata che non ne usufuisce, anzi molto spesso le grandi aziende tendono ad investire su tali sistemi; in particolare nella loro completa automatizzazione tramite software. Inoltre, si stanno iniziando ad utilizzare le AI (Artificial Inttelligence) anche in questo settore, alcuni esempi sono Google, Amazon e Facebook; tale approccio sicuramente in breve tempo rivoluzionerà diversi aspetti dei sistemi gestionali e il modo stesso con cui ci interfacceremo con loro.

1.2 Vantaggi dell'utilizzo dei sistemi gestionali

I vantaggi nell'utilizzo di sistemi gestionali sono molteplici e si riflettono positivamente in diverse sfere di un'organizzazione.

Ci riferiamo ad esempio, alla gestione delle risorse di un'azienda, come l'inventario o il personale, ai processi operativi, dove essi contribuiscono ad accellerare la velocità delle attività quotidiane riducendo gli errori umani, all'analisi dei dati, così da prendere decisioni informate e strategiche basate su dati.

Questi strumenti tecnologici, infatti, sono progettati per semplificare e ottimizzare il processo decisionale, migliorare l'efficienza operativa e fornire una visione più completa e dettagliata dell'andamento aziendale.

Grazie all'automazione di compiti ripetitivi e alla centralizzazione dei dati, i sistemi gestionali consentono di risparmiare tempo e risorse, permettendo ai dipendenti di concentrarsi su compiti di maggior valore aggiunto.

Inoltre, la gestione integrata delle informazioni facilita il monitoraggio delle performance aziendali, favorendo una maggiore trasparenza e la possibilità di individuare eventuali criticità o opportunità di miglioramento.

La riduzione degli errori e la possibilità di accedere ai dati in tempo reale rendono le decisioni più tempestive ed efficaci, promuovendo un ambiente di lavoro più collaborativo e ottimizzato.

Vanno, poi, tenuti in considerazione un'altra serie di vantaggi che un sistema gestionale fornisce, che non sono fondamentali per lo sviluppo di un'azienda, ma sicuramente possono essere facilmente allineabili con certe filosofie aziendali. Ad esempio, utilizzando un sistema gestionale si riduce l'impatto ambientale dell'azienda, in quanto si eliminano tantissimi fogli di carta dei registri; ancora, i sistemi gestionali permettono di avere un servizio clienti migliore in quanto permettono di rendere più veloce ed efficiente questo settore garantendo una maggiore soddisfazione del cliente. Per ultimo, l'utilizzo di una tecnologia di questo tipo permette una maggiore mobilità in quanto i sistemi gestionali consentono ai dipendenti addetti di accedere ad essi anche da remoto o da dispositivi mobili.

In definitiva, l'utilizzo di sistemi gestionali si traduce in una maggiore competitività aziendale, una migliore soddisfazione del cliente e nella creazione di un solido fondamento per la crescita e lo sviluppo sostenibile dell'organizzazione.

1.3 Caratteristiche e tipologie dei sistemi gestionali

I gestionali possono essere più o meno complicati a seconda dei loro utilizzi, tuttavia presentano una serie di caratteristiche che, più o meno sviluppate, sono ricorrenti:

- *Centralizzazione dei dati*: i sistemi gestionali consentono di memorizzare e centralizzare tutte le informazioni aziendali in un unico luogo accessibile. Ciò facilita l'accesso ai dati, riduce la duplicazione delle informazioni e garantisce la coerenza dei dati aziendali. Spesso i dati vengono salvati in basi di dati, ma non è escluso che possano essere salvati anche in file locali, in caso di software per realtà aziendali minori.

- *Automazione dei processi*: grazie alla capacità di automatizzare compiti ripetitivi, i sistemi gestionali migliorano l'efficienza operativa, riducendo il carico di lavoro manuale e prevenendo errori umani.
- *Integrazione dei moduli*: i sistemi gestionali integrano vari moduli funzionali, come finanza, contabilità, risorse umane, inventario e vendite. Questa integrazione permette ai dipendenti di accedere a tutte le informazioni necessarie in un'unica piattaforma. I vari moduli poi possono essere più o meno accessibili a seconda dell'utente che utilizza il software
- *Reporting e analisi*: i sistemi gestionali offrono funzionalità di reporting e analisi avanzate, consentendo alla direzione di ottenere una panoramica completa delle prestazioni aziendali e prendere decisioni basate sui dati.
- *Sicurezza dei dati*: la sicurezza dei dati è una caratteristica fondamentale dei sistemi gestionali. Essi includono controlli di accesso, crittografia dei dati e funzionalità di backup per proteggere le informazioni aziendali da accessi non autorizzati o perdite. Nel caso di utilizzo di una base di dati, la sicurezza è sicuramente maggiore, ed in parte questo aspetto viene affrontato direttamente nel database piuttosto che solo nel software.

Nel mondo lavorativo, in particolare in ambito aziendale, ci sono diversi ambiti che potrebbero essere digitalizzati, e in parte, automatizzati tramite software come i sistemi gestionali; tuttavia, ogni caso è a sè, ed ogni gestionale deve essere realizzato ad hoc quasi in ogni situazione. Bisogna considerare, però, che, anche se ognuna con le sue differenze, le aziende possono essere raggrupate in base al loro settore lavorativo ed in base all'utilizzo che intendono fare del gestionale. Dunque, molto spesso, è possibile partire da una certa tipologia di gestionale piuttosto che da un'altra prendendo in considerazione tali parametri. Come già accennato nell'introduzione, esistono diversi tipi di sistemi gestionali che svolgono compiti diversi e hanno scopi diversi. Volendo generalizzare, potremmo indicare i seguenti come i più utilizzati:

- *Enterprise Resource Planning (ERP)*: gli ERP sono sistemi gestionali che integrano tutte le funzioni aziendali in un'unica piattaforma. Essi coprono processi come finanza, risorse umane, gestione della produzione, vendite e distribuzione. L'integrazione dei processi permette una visione completa delle operazioni aziendali, migliorando l'efficienza e la collaborazione tra i dipartimenti.
- *Customer Relationship Management (CRM)*: i CRM sono sistemi focalizzati sulla gestione delle relazioni con i clienti. Essi aiutano a tracciare le interazioni con i clienti, gestire il marketing e migliorare il servizio clienti. Grazie alla raccolta di dati sui clienti, le aziende possono personalizzare le offerte e fidelizzare la clientela.
- *Supply Chain Management (SCM)*: i sistemi SCM si concentrano sulla gestione e l'ottimizzazione della catena di approvvigionamento, migliorando l'efficienza nella produzione e distribuzione dei prodotti. Monitorando il flusso delle merci, gli SCM consentono di ridurre i tempi di consegna e i costi logistici.
- *Human Resource Management System (HRMS)*: gli HRMS si dedicano alla gestione delle risorse umane, inclusi processi come assunzioni, paghe, formazione e valutazione delle prestazioni. Facilitando la gestione del personale, gli HRMS migliorano la produttività e il benessere dei dipendenti.

- *Inventory Management System*: questi sistemi gestionali sono specificamente progettati per gestire l'inventario aziendale, permettendo di tenere traccia delle scorte e di ottimizzare gli approvvigionamenti. Mantenendo il giusto livello di inventario, le aziende possono evitare esaurimenti o sprechi di magazzino.

1.4 Ruolo dei sistemi gestionali nella gestione aziendale

Il ruolo dei sistemi gestionali nella gestione aziendale è fondamentale per il successo e la crescita delle organizzazioni moderne.

Questi sistemi forniscono un'infrastruttura tecnologica completa e integrata che agevola la pianificazione, l'organizzazione, il monitoraggio e il controllo di tutte le attività aziendali.

Grazie alla centralizzazione dei dati, i sistemi gestionali consentono agli amministratori e ai responsabili di avere una visione globale e tempestiva delle operazioni aziendali, facilitando la presa di decisioni informate e strategiche. La raccolta e l'analisi dei dati aziendali permettono di individuare tendenze, identificare opportunità di miglioramento e prevenire potenziali criticità.

Inoltre, l'automazione dei processi riduce la dipendenza da attività manuali, aumentando l'efficienza operativa e liberando risorse umane per compiti di maggiore valore aggiunto. I sistemi gestionali migliorano anche la comunicazione interna ed esterna, facilitando la collaborazione tra i dipendenti e con i clienti.

La sicurezza dei dati è garantita da controlli di accesso, crittografia e procedure di backup, proteggendo le informazioni sensibili dell'azienda da minacce esterne e interne.

Nel complesso, i sistemi gestionali hanno un ruolo di supporto per ogni aspetto della gestione aziendale, consentendo alle organizzazioni di ottimizzare le risorse, migliorare l'efficienza e mantenere un vantaggio competitivo nel mercato globale.

1.5 Problematiche e sfide nell'implementazione di sistemi gestionali

L'implementazione di sistemi gestionali può presentare diverse problematiche e sfide per le aziende. Una delle principali è rappresentata dalla complessità dell'implementazione stessa.

Integrare e personalizzare un sistema gestionale per soddisfare le specifiche esigenze aziendali richiede tempo, risorse e competenze tecniche adeguate.

Inoltre, la migrazione dei dati da sistemi precedenti può rivelarsi complicata e rischiosa, richiedendo un'attenta pianificazione per evitare perdita o corruzione di dati critici.

Un'altra problematica comune è la resistenza al cambiamento da parte dei dipendenti.

L'introduzione di un nuovo sistema gestionale può suscitare preoccupazioni riguardo alla formazione, alla curva di apprendimento e alla perdita di familiarità con i vecchi processi.

Per affrontare questa sfida, è fondamentale coinvolgere i dipendenti fin dalle prime fasi dell'implementazione, offrendo formazione e supporto costante per garantire una transizione senza intoppi.

La sicurezza dei dati è una preoccupazione cruciale durante l'implementazione di sistemi gestionali.

La gestione e la protezione delle informazioni sensibili richiedono rigorosi controlli di accesso, monitoraggio costante e procedure di backup affidabili per prevenire violazioni della sicurezza e perdite di dati.

Molto spesso, proprio per questi motivi, ci si appoggia su servizi e DBMS (Database Management System) esterni che conferiscono, almeno, delle sicurezze di base per i dati e le risorse in generale.

Inoltre, il costo dell'implementazione può essere una sfida significativa, soprattutto per le piccole e medie imprese.

Gli investimenti in hardware, software, formazione e assistenza tecnica possono rappresentare una barriera finanziaria per alcune aziende.

È importante valutare attentamente i costi e i benefici dell'implementazione di un sistema gestionale e pianificare un budget adeguato.

Infine, la scelta del sistema gestionale più adatto alle esigenze aziendali può essere complessa, considerando la vasta gamma di soluzioni disponibili sul mercato.

La valutazione delle diverse opzioni richiede una ricerca accurata e una valutazione delle funzionalità, delle capacità di scalabilità e dell'affidabilità dei fornitori.

Nonostante queste sfide, superare con successo l'implementazione di un sistema gestionale porta notevoli benefici a lungo termine, migliorando l'efficienza operativa, la produttività e la competitività aziendale.

1.6 Ruolo dell'IT nei sistemi gestionali

L'Information Technology (IT) svolge un ruolo fondamentale nell'implementazione e nella direzione dei sistemi gestionali all'interno delle aziende. Gli specialisti IT sono responsabili della scelta, dell'installazione e della configurazione del DBMS e del sistema gestionale più adatto alle esigenze aziendali. Essi collaborano con i reparti aziendali per comprendere le esigenze funzionali e definire le specifiche del sistema da implementare.

Durante l'implementazione, l'IT è coinvolto nella creazione dello schema del database, nella definizione delle tabelle, degli attributi e dei vincoli, garantendo una struttura dati coerente e ben organizzata. Gli specialisti IT sono responsabili anche della migrazione dei dati da sistemi precedenti, assicurandosi che i dati siano trasferiti correttamente nel nuovo sistema gestionale.

Inoltre, l'IT è responsabile della sicurezza del sistema gestionale. Gli specialisti IT implementano controlli di accesso e meccanismi di crittografia per proteggere i dati aziendali da accessi non autorizzati e garantire la conformità alle normative sulla privacy.

Dopo l'implementazione, l'IT monitora e gestisce il sistema gestionale per garantire un funzionamento efficiente e affidabile. Gli specialisti IT eseguono backup regolari dei dati e pianificano operazioni di manutenzione per prevenire eventuali guasti o perdite di dati.

Inoltre, l'IT fornisce formazione e supporto agli utenti interni, aiutandoli a utilizzare correttamente il sistema gestionale e risolvendo eventuali problemi o difficoltà incontrate durante l'utilizzo.

Nel complesso, il ruolo dell'IT nei sistemi gestionali è cruciale per garantire un'implementazione e un funzionamento efficace e sicuro del sistema. Gli specialisti IT lavorano in sinergia con gli altri reparti aziendali per massimizzare i benefici del sistema gestionale e migliorare l'efficienza, la produttività e la competitività dell'azienda.

CAPITOLO 2

Specifica e Analisi dei requisiti

Nel secondo capitolo viene descritta nel dettaglio la struttura della società per cui è stato creato il gestionale, analizzandone i problemi e le eventuali soluzioni che sarebbero già disponibili sul mercato. Successivamente si spiega come sono stati raccolti i dati per la progettazione del software, in particolare come si sono acquisiti i requisiti e come questi poi sono stati trasformati in casi d'uso.

2.1 Descrizione della società

2.1.1 Struttura della società

La società sportiva Rugby Falconara Dinamis ASD è stata fondata nel 2006 da un piccolo gruppo di appassionati che intravidero nella pratica del rugby la possibilità di promuovere all'interno della città quel complesso di valori sociali ed etici che sono propri di questo sport e fondamento della società civile. L'idea era quella di proporre lo sport nella città di Falconara a partire dai bambini di 6 anni fino ad arrivare agli adulti e gli "Old", considerando che nel rugby l'età massima per giocare a livello agonistico è 42 anni. Ogni squadra è definita in base alla categoria di appartenenza; vengono definite "Under", e vengono divisi i ragazzi e le ragazze in fasce di età dispari. Ogni categoria racchiude due annate a partire dall'under 7 fino ad arrivare all'under 19 per poi proseguire con la "Senior", ovvero la prima squadra, sia maschile che femminile, e la old, ovvero gli over 42 anni.

La società mette a disposizione di ogni squadra almeno un allenatore e un accompagnatore, ovvero colui che segue la squadra durante gli eventi della stagione; inoltre, vengono forniti tutti i materiali e le attrezzature che sono neccessarie ad ogni squadra per allenarsi al meglio, come conetti, pettorine e ovviamente palloni.

É presente una figura che si occupa specificatamente del tesseramento degli atleti e dei collaboratori in campo; questa è la segretaria. Essa richiede ad ognuno una serie di documenti necessari all'iscrizione; il principale è il "modello12". Poi, ovviamente, ognuno deve pagare la quota di iscrizione e può scegliere se diventare socio versando una cifra simbolica ulteriore che gli permette di diventare sostenitore attivo e di particiapare alle due riunioni annuali che vengono determinate dal presidente della società e dai suoi consiglieri; quest'utlimi di norma variano da 5 agli 8.

La società poi dispone di un piccolo settore dedicato al merchandising in cui vengono venduti, ai tesserati e ai semplici tifosi, l'abbigliamento ufficiale del club ed eventuali attrezzature dedicate ai giocatori. Sostanzialmente questa parte si compone di un piccolo negozio

che è presente in sede in cui è possibile vedere e provare l'abbigliamento in vendita, successivamente però, salvo rare eccezioni, vengono fatti degli ordini sul sito internet della azienda Macron, sponsor tecnico della società, che riserva una sezione del sito all'abbigliamento del Falconara.

Per quanto riguarda l'organizzazione delle attività sportive, di solito ogni categoria svolge almeno 2 allenamenti settimanali, che salgono a seconda della fascia di età, e solitamente la partita di domenica. Per le categorie dall'Under 17 in su, negli eventi ufficiali, l'accompagnatore deve stilare un foglio, chiamato Lista gara, in cui viene ratificata la formazione all'arbitro della partita in questione; tale lista è necessaria per il corretto svolgimento della manifestazione.

2.1.2 Problematiche e difficoltà organizzative rilevate

La società si trova spesso in difficoltà quando c'è bisogno di rintracciare coloro che hanno problemi a livello di documentazione. In particolare, risulta difficoltoso ogni volta controllare tutte le scadenze dei certificati, in particolare quello medico, e tenere traccia di tutti gli altri moduli che ogni atleta deve avere con sè; questo perchè ogni documento viene archiviato in formato cartaceo, o al più ne viene segnata la corretta compilazione su un foglio excel relativo a tutti gli iscritti. Inoltre non sembra esserci un organizzazione complessiva degli iscritti, degli atleti e delle figure esterne, rendendo poi difficoltosi i vari lavori di segreteria.

Questo discorso, essendo una piccola società, è allargabile a tutto il resto. Risulta non esserci una divisione chiara degli attrezzi che ogni categoria dovrebbe avere e, in questo contesto, anche il merchandising funziona scrivendo gli ordini che spesso, passando di mano in mano, riusltano poco chiari o si perdono del tutto.

Inoltre emerge anche la necessità di tenere traccia sia delle partite che degli allenamenti, in quanto, essendo ogni categoria a sè, spesso ci sono settimane in cui ogni giorno si allena almeno una squadra. Questo genera caos a livello organizzativo, sopratutto perchè risulta poi complicato controllare l'operato dello staff tecnico e le presenze degli atleti sul campo. Oltretutto una non chiara schedulazione degli eventi comporta anche scompiglio tra tutti coloro che si occupano dell'organizzazione degli stessi, come accoglienza e ristoro.

2.1.3 Identificazione principali concorrenti

Sarebbe errato dire che non esiste, per una società di rugby, un modo per organizzare i dati provenienti dalle iscrizioni o quelli derivanti da altri aspetti societari. Ci sono due principali servizi web che eseguono questo compito, entrambi hanno pregi e difetti. Per un'ottimale sviluppo di progetto si ritiene opportuno analizzare il funzionamento di questi e trarne eventuali spunti. Questi ultimi vengono descritti di seguito:

Rugby Manager 2.0

Questo è il principale competitor nel web. Si tratta di un gestionale online che è stato ideato e poi progettato da un team di sviluppatori del Rugby Perugia con a capo il loro direttore tecnico. Dunque si tratta di un software costruito su misura per il rugby in particolare per la società perugina. Permette di salvare le informazioni anagrafiche, ma non solo, di tutti gli atleti dando poi la possibilità di dividere ognuno in base alla categoria di appartenenza. Esiste poi una funzione per gli allenamenti, mettendo a disposizione una serie di strumenti per la valutazione degli stessi e costruendo, poi, sulla base di questi, grafici settimanali e mensili. Infatti il software presenta diverse interfacce che consentono di inserire dati importanti per il tracciamento delle sedute settimanali, come, ad esempio, le presenze o la qualità di vari aspetti tecnici. Oltre a queste che sono le sue principali funzioni, esso contiene anche delle linee guida

e dei documenti pre-compilati per agevolare il compito degli accompagnatori delle squadre agli eventi. Ovviamente il gestionale gestisce gli accessi con un classico login, permettendo ad un account amministratore di creare account per gli allenatori con funzionalità limitate, in base ai loro compiti. Il servizio è a pagamento su base annua o mensile.

Questo sistema gestionale è complessivamente completo e ricco di funzionalità. Tuttavia quello che dovrebbe essere un pregio può essere, però, anche un difetto. Infatti ci sono una serie di funzioni che, seppur interessanti, risultano inefficaci in piccoli club come quello di Falconara. Funzionale poi è il metodo in cui vengono salvati e organizzati i dati degli iscritti, anche se l'interfaccia è un po' antiquata e non così intuitiva, cosa che potrebbe risultare fondamentale quando si devono eseguire operazioni semplici e veloci. La limitazione delle funzionalità per gli allenatori è, sicuramente, molto utile, perchè permette ai tecnici stessi di gestire le loro squadre, e anche le linee guida per gli accompagnatori sono funzionali. Ciò nonostante va sicuramente segnalata la mancanza di area riservata alle attrezzature delle diverse squadre e del merchandising, che nel caso del rugby Falconara è parte integrante. Come ultima cosa va anche considerato il costo, che seppur basso, risulta essere una spesa che, se possibile, viste e considerate le dimensioni del club, e, di conseguenza, la forza economica, dovrebbero essere evitate.

Gestionale FIR: firweb.gisa.net

Questo è un'altro strumento organizzativo, creato appositamente per il rugby, messo a disposizione gratuitamente dalla FIR (Federazione Italiana Rugby). In realtà questa piattaforma va utlizzata obbligatoriamente da ogni società rugbistica, in quanto viene usata a livello nazionale per tracciare le iscrizioni nei vari club; per le funzionalità che offre potrebbe essere considerata come una sorta di gestionale. Questo servizio permette di registrare tutti gli atleti, e più in generale i tesserati, della società; per ognuno vanno immessi, oltre a tutte le informazioni anagrafiche e altre informazioni di vario tipo, i documenti obbligatori per l'iscrizione; il principale è il modello 12 e per gli atleti è allo stesso modo importante il certificato medico agonistico. Tutti i tesserati vengono divisi nelle loro categorie di appartenenza e, nel sito, è possibile ricercare gli iscritti in base a queste.

Il servizio risulta completo per quanto riguarda il lato degli iscritti; non sembrano esserci parti mancanti o poco chiare; tuttavia viene complemtamente a mancare una parte dedicata agli eventi e all'inventario. Nel caso preso in esame, ovvero quello del rugby Falconara, probabilmente questa soluzione risolverebbe solo un terzo dei problemi rilevati e, poi, costringerebbe la società a virare su opzioni complementari a questa per gestire le altre due macro aree.

2.1.4 Acquisizione dei requisiti e metodologie usate

Dopo aver analizzato le varie parti che caratterizzano il funzionamento della società, si è proceduto con la fase di raccolta dei requisiti e, di conseguenza, delle specifiche di progetto. Questa è una fase fondamentale per il corretto sviluppo, in quanto costituisce la base da cui partire per fare l'analisi e la progettazione. Per raccogliere i requisiti è stata fatta un'intervista al presidente della società chiedendo esplicitamente quali fossero le problematiche del club e le parti necessarie che avrebbero dovuto comporre il gestionale per risultare efficace. Inoltre, è stata interpellata anche la segretaria per avere un quadro più specifico di quale fosse la situazione lavorativa attuale e quale fosse il grado di dettaglio che richiedeva quel settore. Infine, per quanto riguarda l'area relativa all'inventario e agli eventi sportivi, è stata chiesta una testimonianza al direttore tecnico della società, che ha espresso quali fossero le necessità dando anche qualche spunto su quali potevano essere delle funzionalità, utili ai tecnici di campo, da implementare. Unendo le informazioni sulla società, riportate all'inizio del

capitolo, e quelle acquisite con queste interviste, si è stilata una lista di specifiche che, poi, si trasformeranno in requisiti.

2.1.5 Specifiche di progetto

Unendo le informazioni sulla società, riportate all'inizio del capitolo, e quelle acquisite con le interviste si è stilata una lista di specifiche che poi si trasformeranno in requisiti. Da queste, in particolare, si andranno a prendere quei requisiti definiti come funzionali, ovvero che comprendono le funzionalità che il progetto dovrà effettivamente contenere e implementare.

La segreteria ha bisogno di un sitema semplice ma, allo stesso tempo, efficente che permetta di tenere organizzati tutti i tesserati del club. Per ognuno ci sarà bisogno di tenere traccia, oltre dei classici dati, dei documenti. Visto che, dalle interviste, è emerso che i documenti sono sempre tenuti in formato cartaceo per via delle direttive FIR, non c'è necessità di salvarli in formato digitale, ma deve essere indicato se il documento è presente o meno in archivio. Inoltre, visto che il problema principale è ricordarsi ogni volta quando scadono i certificati medici, è fondamentale avere un sistema di notifiche che invii un avviso quando un certificato è scaduto.

Seguendo, invece, le direttive del direttore tecnico, ogni categoria deve avere una sua specifica area in cui indicare la tipologia e la quantità di oggetti che dispone in modo tale dà non avere mai carenze di attrezzature. Eventualmente andrebbe indicato il negozio o, in generale, la provenienza di ogni strumento, così da poter poi ricomprare il materiale in caso di necessità. Fondamentale, poi, una piccola area dedicata al merchandising in cui si potranno aggiungere gli ordini ricevuti così da tenerne traccia.

Inoltre, andranno salvati tutti gli allenamenti e le partite fatte. Per quanto riguarda gli eventi ufficiali, a partire dalla categoria Under 17, andrà anche tenuta in considerazione la formazione titolare che scenderà in campo, così da agevolare il lavoro degli accompagnatori e il loro dialogo con i rispettivi tecnici. Inoltre, sempre seguendo le esigenze del direttore tecnico, è emerso il desiderio di avere un semplice sistema di analisi degli allenamenti e delle partite, così da educare gli allenatori a porre maggiore attenzione al lavoro da eseguire sul campo; non è un aspetto prioritario ma una sorta di avviamento, che, eventualmente, poi porterà all'utilizzo di software di analisi specifici e dedicati.

Ci sono, poi, altre due specifiche che sono state fatte, che, però, non sono delle vere e proprie funzionalità, ma più che altro dei vincoli a cui il software dovrà possibilmente sottostare, ovvero è stato esplicitamente richiesto da quasi tutte le figure interpellate che il sistema abbia un'interfaccia grafica intuitiva e il meno complessa possibile. Inoltre il presidente della società ha richiesto che ci fosse una schermata di autenticazione in modo tale da poter differenziare e scandire tutti gli utenti, andando possibilmente a limitare le funzionalità a seconda di chi userà il gestionale.

2.2 Analisi dei requisiti

2.2.1 Divisione del progetto in macroaree

Analizzando i pochi dati ottenuti fino ad adesso, risulta subito visibile una divisione netta di tre settori di lavoro in base alle funzionalità che andranno implementate. Queste sono:

- *Iscritti*: in questa macroarea andranno tutti gli iscritti di cui la segreteria si dovrà occupare. Verrà fatta in modo tale che sia separata dal resto, così da non confondersi durante il lavoro. In particolare, questa area del gestionale, oltre che dall'amministratore, sarà accessibile soltanto dalla segreteria.

- *Inventario*: qui andranno raccolte tutte le informazioni riguardanti le attrezzature di tipo tecnico e non; in più, ci sarà una sezione dedicata al merchandising separata dal resto. Quest'area sarà di uso comune sia a tecnici che alla segreteria perchè in questa si intrecciano gli interessi di ambo le parti.
- *Eventi*: con questo nome si intende riassumere tutti le manifestazioni ufficiali della società, ovvero partite e allenamenti. In quest'ultimo caso, l'accesso è permesso soltanto ai tecnici, non essendo un'area di competenza della segreteria.

2.2.2 Individuazione ed enumerazione dei requisiti

Si procede enumerando i requisiti individuati a seguito delle ricerche e delle conseguenti preliminari analisi svolte sopra

- *Gestione Iscritti*
  - *CRUD Iscritti*: il sistema dovrà gestire le attività CRUD sugli iscritti.
  - *Apri scheda iscritti*: il sistema dovrà gestire la visualizzazione delle singole schede degli iscritti.
  - *Visualizza iscritti per categoria*: il sistema dovrà gestire la visualizzazione degli iscritti in base alla categoria di appartenenza.
  - *Ordina per nome*: il sistema dovrà gestire l'ordinamento degli iscritti in base al loro nome.
  - *Ordina per data di nascita*: il sistema dovrà gestire l'ordinamento degli iscritti in base alla loro data di nascita.
  - *Ricerca iscritti per nome*: il sistema dovrà gestire la ricerca di un iscritto per nome e visualizzare la lista di risultati corrispondenti.
  - *Ricerca per codice fiscale*: il sistema dovrà gestire la ricerca di un iscritto per codice fiscale e visualizzare la lista di risultati corrispondenti.
- *Gestione inventario*
  - *CRUD oggetti*: il sistema dovrà gestire le attività CRUD sugli oggetti.
  - *Ordina oggetto* per nome: il sistema dovrà gestire l'ordinamento degli oggetti in base al loro nome.
  - *Ordina per quantità*: il sistema dovrà gestire la visualizzazione della lista degli oggetti in base alla loro quantità.
  - *Visualizza oggetti per categoria*: il sistema dovrà gestire la visualizzazione della lista degli oggetti in base alla categoria di appartenenza.
  - *Ricerca oggetti per nome*: il sistema dovrà gestire la ricerca di un oggetto per nome e visualizzare la lista di risultati corrispondenti.
  - *Ricerca oggetti per quantità*: il sistema dovrà gestire la ricerca di un oggetto per quantità e visualizzare la lista di risultati corrispondenti.
  - *Apri scheda oggetti*: il sistema dovrà gestire la visualizzazione delle singole schede degli oggetti.
- *Gestione eventi*
  - *CRUD Eventi*: il sistema dovrà gestire le operazioni CRUD sugli eventi.

  - *Visualizza allenamenti e partite*: il sistema dovrà gestire la visualizzazione degli allenamenti e delle partite registrati in base alla categoria.
  - *Apri scheda evento*: il sistema dovrà gestire la visualizzazione delle singole schede degli allenamenti e delle partite.
  - *Inserisci formazione*: il sistema dovrà permettere di inserire la formazione di una partita.
  - *Modifica formazione*: il sistema dovrà permettere di modificare la formazione di una partita.
  - *Ordina per tipo*: il sistema dovrà gestire l'ordinamento degli eventi in base al loro tipo.
  - *Ordina per data*: il sistema dovrà gestire l'ordinamento degli eventi in base alla data.
  - *Ricerca per data*: il sistema dovrà gestire la ricerca degli eventi in base alla loro data e, poi, visualizzare i risultati.
  - *Ricerca per note*: il sistema dovrà gestire la ricerca degli eventi in base alla loro data e, poi, visualizzare i risultati.

- *Generali*
  - *Notifiche pop up*: il sistema dovrà gestire un sistema di notifiche pop up nello schermo.
  - *Aggiunta utenti*: il sistema dovrà permettere a nuovi utenti di registrarsi e accedere.

2.2.3 Requisti funzionali e non funzionali

La divisione in requisti funzionali e non funzionali serve, a livello di analisi, a capire quali sono funzionalità da implementare e quali sono, invece,, i vincoli costruttivi da rispettare. In questo caso, tutti i requisti sopra enumerati sono funzionali. Ciò era comunque evidente visto che si trattavano di elementi che il sistema dovrà implementare. I requisiti non funzionali, invece, sono i seguenti:

- *Phyton3*: il sistema dovrà essere implementato in Phyton 3.
- *GUI*: il sistema dovrà essere dotato di un'interfaccia grafica.
- *Login*: il sistema dovrà permettere l'accesso attraverso l'utilizzo di un nome utente e una password.
- *Univocità*: il sistema dovrà catalogare ogni entità in maniera univoca e senza generare ambiguità.

Questi, che sono a tutti gli effetti vincoli da rispettare, sono stati estratti dalle richieste degli intervistati della società; ne sono stati, poi, aggiunti altri per far sì che il sistema gestionale sia completo, corretto ed efficiente.

2.2.4 Casi d'uso e Attori

Una volta estrapolati i requisiti si passa alla creazione dei casi d'uso. Questi sono una tecnica di analisi e documentazione utilizzata nell'ambito dell'ingegneria del software e nella progettazione dei sistemi. Essi rappresentano situazioni o scenari specifici in cui un sistema o un'applicazione deve essere utilizzato per soddisfare le esigenze degli utenti o raggiungere

determinati obiettivi. Sostanzialmente è una descrizione narrativa che illustra come un utente interagisce con il sistema in una determinata situazione. È focalizzato su "cosa" fa il sistema, anziché su "come" lo fa.

Prima di ciò, è importante, però, definire gli attori, ovvero quelle figure che, poi, andranno ad usare il gestionale. Questi sono la segretaria, che si occuperà, ovviamente, del lato burocratico della società, i vari tecnici, che invece dovranno avere accesso alla parte degli eventi e dell'inventario, ed, infine, l'utente amministratore, che, in questo caso, coincide con il presidente della società, che avrà accesso ad ogni funzionalità del software.

Ora vengono enumerati i casi d'uso, seguiti dal loro normale flusso decrittivo, composto da: *Attori, Descrizione, Flusso principale e eventuali sequenze alternative*.

Gestione iscritti:

- *CRUDIscritti*
  - *Descrizione:* Il caso d'uso CRUDIscritti consente l'inserimento, la modifica e la rimozione di Iscritti.
  - *Attori Primari:* Amministratore, Segretaria
  - *Attori Secondari:* Nessuno
  - *Precondizioni:*
    1. L'utente deve avere un account per accedere ai dati.
  - *Postcondizioni:* Nessuna
  - *Sequenza degli eventi principale:*
    1. Il caso d'uso inizia quando l'attore primario vuole effettuare un'operazione CRUD relativa ad un iscritto.
    2. `if` l'attore vuole inserire un nuovo iscritto:
       (a) L'attore inserisce tutti i dati relativi al nuovo iscritto.
    3. `else if` l'attore vuole modificare un dato di un iscritto già esistente:
       (a) L'attore inserisce dei parametri per ricercare l'iscritto da modificare.
       (b) Il sistema mostra l'iscritto trovato.
       (c) L'attore modifica i dati desiderati dell'iscritto.
       (d) Il sistema mostra un messaggio di informazione che conferma l'esito positivo dell'operazione.
    4. `else if` l'attore vuole rimuovere un iscritto:
       (a) L'attore inserisce dei parametri per ricercare l'iscritto da rimuovere.
       (b) Il sistema rimuove l'iscritto dalla lista.
       (c) Il sistema mostra un messaggio di informazione che conferma l'esito positivo dell'operazione.
  - *Sequenza degli eventi alternativa:* Nessuna

- *VisualizzaIscritti*
  - *Descrizione:* Il caso d'uso VisualizzaIscritti permette di visualizzare la lista di iscritti raggruppati in base alla loro categoria di appartenenza. Oltre a questa catalogazione è possibile anche ordinare gli iscritti di ogni categoria in base al codice fiscale o al nome.
  - *Attori Primari:* Amministratore, Segretaria
  - *Attori Secondari:* Nessuno

  - *Precondizioni:*
    1. L'utente deve avere un account per accedere ai dati.
  - *Postcondizioni:* Nessuna
  - *Sequenza degli eventi principale:*
    1. Il caso d'uso inizia quando un attore vuole visualizzare la lista degli iscritti.
    2. `if` l'attore vuole visualizzare la lista in base al nome:
       (a) L'attore inserisce il senso di ordinamento.
       (b) `include`(OrdinaIscritti).
    3. `else if` l'attore vuole visualizzare la lista degli iscritti in base al codice fiscale:
       (a) L'attore inserisce il senso di ordinamento.
       (b) `include`(OrdinaIscritti).
    4. Il sistema mostra sullo schermo la lista ordinata.
  - *Sequenza degli eventi alternativa:* Nessuna

- *OrdinaIscritti*
  - *Descrizione:* Il caso d'uso OrdinaIscritti consente di ordinare la lista degli iscritti in base al nome o al codice fiscale.
  - *Attori Primari:* Nessuno.
  - *Attori Secondari:* Nessuno.
  - *Precondizioni:* Nessuna.
  - *Postcondizioni:* Nessuna.
  - *Sequenza degli eventi principale:*
    1. Il caso d'uso inizia quando il sistema deve effettuare un ordinamento della lista.
    2. Il sistema rileva il tipo di ordinamento richiesto.
    3. Il sistema ordina la lista come richiesto.
  - *Sequenza degli eventi alternativa:* Nessuna

- *ApriSchedaIscritti*
  - *Descrizione:* Il caso d'uso ApriSchedaIscritti permette di visualizzare specifici dati appartenenti ad ogni iscritto.
  - *Attori Primari:* Amministratore, Segretaria
  - *Attori Secondari:* Nessuno
  - *Precondizioni:*
    1. L'utente deve avere un account per accedere ai dati.
    2. L'utente sta visualizzando la lista di iscritti.
  - *Postcondizioni:* Nessuna
  - *Sequenza degli eventi principale:*
    1. Il caso d'uso inizia quando l'utente seleziona un iscritto.
    2. Il sistema visualizza sullo schermo la scheda informativa relativa all'iscritto selezionato.
  - *Sequenza degli eventi alternativa:* Nessuna

Gestione inventario:

- *CRUDOggetti*
  - *Descrizione:* Il caso d'uso CRUDOggetti consente l'inserimento, la modifica e la rimozione di oggetti dell'inventario.
  - *Attori Primari:* Amministratore, Segretaria
  - *Attori Secondari:* Nessuno
  - *Precondizioni:*
    1. L'utente deve avere un account per accedere ai dati.
  - *Postcondizioni:* Nessuna
  - *Sequenza degli eventi principale:*
    1. Il caso d'uso inizia quando l'attore primario vuole effettuare un'operazione CRUD relativa ad un oggetto.
    2. `if` l'attore vuole inserire un nuovo oggetto:
       (a) L'attore inserisce tutti i dati relativi al nuovo oggetto.
    3. `else if` l'attore vuole modificare un dato di un oggetto già esistente:
       (a) L'attore inserisce dei parametri per ricercare l'oggetto da modificare.
       (b) Il sistema mostra l'oggetto trovato.
       (c) L'attore modifica i dati desiderati dell'oggetto.
       (d) Il sistema mostra un messaggio di informazione che conferma l'esito positivo dell'operazione.
    4. `else if` l'attore vuole rimuovere un oggetto:
       (a) L'attore inserisce dei parametri per ricercare l'oggetto da rimuovere.
       (b) Il sistema rimuove l'oggetto dall'inventario.
       (c) Il sistema mostra un messaggio di informazione che conferma l'esito positivo dell'operazione.
  - *Sequenza degli eventi alternativa:* Nessuna
- *VisualizzaOggetti*
  - *Descrizione:* Il caso d'uso VisualizzaOggetti permette di visualizzare la lista di oggetti e il loro relativo stato. Oltre a questa catalogazione è possibile anche ordinare gli oggetti di ogni categoria in base alla loro quantità o al nome.
  - *Attori Primari:* Amministratore, Segretaria
  - *Attori Secondari:* Nessuno
  - *Precondizioni:*
    1. L'utente deve avere un account per accedere ai dati.
  - *Postcondizioni:* Nessuna
  - *Sequenza degli eventi principale:*
    1. Il caso d'uso inizia quando un attore vuole visualizzare la lista degli oggetti.
    2. `If` l'attore vuole visualizzare la lista in base al nome:
       (a) L'attore inserisce il senso di ordinamento.
       (b) `include`(OrdinaOggetti).
    3. `else if` l'attore vuole visualizzare la lista degli oggetti in base alla quantità:
       (a) L'attore inserisce il senso di ordinamento.

(b) `include`(OrdinaOggetti).

4. Il sistema mostra sullo schermo la lista ordinata.

– *Sequenza degli eventi alternativa:* Nessuna

- *OrdinaOggetti*
  - *Descrizione:* Il caso d'uso OrdinaOggetti consente di ordinare la lista degli oggetti in base alla quantità o al nome.
  - *Attori Primari:* Nessuno.
  - *Attori Secondari:* Nessuno.
  - *Precondizioni:* Nessuna.
  - *Postcondizioni:* Nessuna.
  - *Sequenza degli eventi principale:*
    1. Il caso d'uso inizia quando il sistema deve effettuare un ordinamento della lista.
    2. Il sistema rileva il tipo di ordinamento richiesto.
    3. Il sistema ordina la lista come richiesto.
  - *Sequenza degli eventi alternativa:* Nessuna
- *ApriSchedaOggetti*
  - *Descrizione:* Il caso d'uso ApriSchedaOggetti permette di visualizzare specifici dati appartenenti ad ogni oggetto.
  - *Attori Primari:* Amministratore, Segretaria
  - *Attori Secondari:* Nessuno
  - *Precondizioni:*
    1. L'utente deve avere un account per accedere ai dati.
    2. L'utente sta visualizzando la lista di oggetti.
  - *Postcondizioni:* Nessuna
  - *Sequenza degli eventi principale:*
    1. Il caso d'uso inizia quando l'utente seleziona un oggetto.
    2. Il sistema visualizza sullo schermo la scheda informativa relativa all'oggetto selezionato.
  - *Sequenza degli eventi alternativa:* Nessuna

Gestione eventi:

- *CRUDEventi*
  - *Descrizione:* Il caso d'uso CRUDEventi consente l'inserimento, la modifica e la rimozione di eventi.
  - *Attori Primari:* Amministratore, Allenatore
  - *Attori Secondari:* Nessuno
  - *Precondizioni:*
    1. L'utente deve avere un account per accedere ai dati.
  - *Postcondizioni:* Nessuna

  - *Sequenza degli eventi principale:*
    1. Il caso d'uso inizia quando l'attore primario vuole effettuare un'operazione CRUD relativa ad un evento.
    2. `if` l'attore vuole inserire un nuovo evento:
       (a) L'attore inserisce tutti i dati relativi al nuovo evento.
    3. `else if` l'attore vuole modificare un dato di un evento già esistente:
       (a) L'attore seleziona l'evento da modificare.
       (b) L'attore modifica i dati desiderati dell'oggetto.
       (c) Il sistema mostra un messaggio di informazione che conferma l'esito positivo dell'operazione.
    4. `else if` l'attore vuole rimuovere un evento:
       (a) L'attore seleziona l'evento da rimuovere.
       (b) Il sistema rimuove l'evento dall'inventario.
       (c) Il sistema mostra un messaggio di informazione che conferma l'esito positivo dell'operazione.
  - *Sequenza degli eventi alternativa:* Nessuna

- *CRUDFormazione*
  - *Descrizione:* il caso d'uso CRUDFormazione permette di compiere operazioni CRUD sulla formazione relativa ad una partita
  - *Attori Primari:* Amministratore,Allenatore
  - *Attori Secondari:* Nessuno
  - *Precondizioni:*
    1. l'utente deve avere un account per accedere ai dati
  - *Postcondizioni:* Nessuna
  - *Sequenza degli eventi principale:*
    1. Il caso d'uso inizia quando l'attore primario vuole effettuare un'operazione CRUD relativa ad una formazione di una partita.
    2. `if` l'attore vuole inserire una nuova formazione:
       (a) L'attore inserisce i giocatori.
    3. `else if` l'attore vuole modificare una formazione di una partita già esistente:
       (a) L'attore seleziona la partita da modificare.
       (b) L'attore modifica i giocatori desiderati.
       (c) Il sistema mostra un messaggio di informazione che conferma l'esito positivo dell'operazione.
    4. `else if` l'attore vuole rimuovere una formazione:
       (a) L'attore seleziona la formazione da rimuovere.
       (b) Il sistema rimuove la formazione.
       (c) Il sistema mostra un messaggio di informazione che conferma l'esito positivo dell'operazione.
  - *Sequenza degli eventi alternativa:*

- *VisualizzaEventi*
  - *Descrizione:* Il caso d'uso VisualizzaEventi permette di visualizzare la lista di eventi, sia partite che allenamenti, divisi per categorie. Inoltre, è possibile ordinare gli eventi in base al tipo e alla data di svolgimento.

  - *Attori Primari:* Amministratore, Allenatore
  - *Attori Secondari:* Nessuno
  - *Precondizioni:*
    1. L'utente deve avere un account per accedere ai dati.
  - *Postcondizioni:* Nessuna
  - *Sequenza degli eventi principale:*
    1. Il caso d'uso inizia quando un attore vuole visualizzare la lista degli eventi.
    2. `If` l'attore vuole visualizzare la lista degli allenamenti in base alla data di svolgimento:
       (a) L'attore inserisce il senso di ordinamento.
       (b) `include`(OrdinaEventi).
    3. `else if` l'attore vuole visualizzare la lista degli allenamenti in base al tipo:
       (a) L'attore inserisce il senso di ordinamento.
       (b) `include`(OrdinaEventi).
    4. Il sistema mostra sullo schermo la lista ordinata.
  - *Sequenza degli eventi alternativa:* Nessuna

- *OrdinaEventi*
  - *Descrizione:* Il caso d'uso OrdinaEventi consente di ordinare la lista degli eventi in base al tipo o alla data di svolgimento.
  - *Attori Primari:* Nessuno.
  - *Attori Secondari:* Nessuno.
  - *Precondizioni:* Nessuna.
  - *Postcondizioni:* Nessuna.
  - *Sequenza degli eventi principale:*
    1. Il caso d'uso inizia quando il sistema deve effettuare un ordinamento della lista.
    2. Il sistema rileva il tipo di ordinamento richiesto.
    3. Il sistema ordina la lista come richiesto.
  - *Sequenza degli eventi alternativa:* Nessuna

- *ApriSchedaEventi*
  - *Descrizione:* Il caso d'uso OrdinaEventi consente di ordinare la lista degli eventi in base al tipo o alla data di svolgimento.
  - *Attori Primari:* Nessuno.
  - *Attori Secondari:* Nessuno.
  - *Precondizioni:* Nessuna.
  - *Postcondizioni:* Nessuna.
  - *Sequenza degli eventi principale:*
    1. Il caso d'uso inizia quando il sistema deve effettuare un ordinamento della lista.
    2. Il sistema rileva il tipo di ordinamento richiesto.
    3. Il sistema ordina la lista come richiesto.

  - *Sequenza degli eventi alternativa:* Nessuna

Generali:

- *ScadenzePopUp*
  - *Descrizione:* Il caso d'uso ScadenzePopUp permette di visualizzare sullo schermo delle notifiche riguardanti delle scadenze dei certificati degli iscritti.
  - *Attori Primari:* Tempo.
  - *Attori Secondari:* Nessuno.
  - *Precondizioni:*
    1. Deve essere scaduto almeno un certificato.
  - *Postcondizioni:* Nessuna
  - *Sequenza degli eventi principale:*
    1. Il caso d'uso inizia quando scade un certificato.
    2. Il sistema fa comparire sullo schermo un messaggio di notifica.
  - *Sequenza degli eventi secondaria:* Nessuna.
- *AggiungiUtenti*
  - *Descrizione:* Il caso d'uso AggiungiUtenti permette all'utilizzatore di aggiungere un utente utilizzando un codice.
  - *Attori Primari:* Allenatore, Segretaria, Amministratore
  - *Attori Secondari:* Nessuno.
  - *Precondizioni:* Nessuna
  - *Postcondizioni:* Nessuna
  - *Sequenza degli eventi principale:*
    1. Il caso d'uso inizia quando l'attore primario vuole aggiungere un utente.
    2. `If` l'attore vuole inserire un nuovo utente:
       (a) L'attore inserisce tutti i dati relativi al nuovo utente.
       (b) L'attore inserisce il codice di iscrizione.
       (c) `If` l'attore inserisce un codice valido:
           i. Il sistema aggiunge un utente del tipo specificato dal codice.
           ii. Il sistema informa della completata aggiunta.
       (d) `else if` l'attore inserisce un codice non valido:
           i. Il sistema mostra un messaggio di errore.
  - *Sequenza degli eventi alternativa:* Nessuna.
- *Ricerca*
  - *Descrizione:* Il caso d'uso Ricerca permette di cercare e visualizzare tutti gli elementi del gestionale.
  - *Attori Primari:* Allenatore, Segretaria, Amministratore
  - *Attori Secondari:* Nessuno.
  - *Precondizioni:*
    1. L'utente deve avere un account per accedere ai dati.

- *Postcondizioni:* Nessuna.
- *Sequenza degli eventi principale:*
  1. L'utente immette nell'apposita barra di ricerca i parametri.
  2. `If` i parametri immessi sono giusti:
     (a) Il sistema mostra i risultati.
  3. `else` il sistema non mostra nulla.
- *Sequenza degli eventi alternativa:* Nessuna.

CAPITOLO 3

Progettazione del database sottostante

In questo capitolo vengono descritte le fasi che hanno portato alla progettazione del database per il sistema gestionale. La creazione è stata strutturata ad hoc sia sulle esigenze strutturali del software sia sulle reali esigenze della società di Falconara. Di fatto, la base di dati, è stata creata in due fasi: la prima in cui sono state riorganizzate tutte le informazioni possedute in modo da creare uno schema di partenza, la seconda in cui, invece, si sono svolte delle analisi sulla consistenza e la chiarezza dello schema, per poi a arrivare allo schema finale e, di conseguenza, al modello relazionale.

3.1 Fasi della progettazione del database

Il database su cui si appoggia il gestionale è stato progettato appositamente, su misura per le problematiche che deve risolvere. In una prima fase di brainstorming si sono raccolti tutti i dati provenienti dalle interviste e le informazioni raccolte dalla società di rugby; dopo averle selezionate e sistemante si è proceduto con la creazione di un primo schema concettuale o schema entità-relazione. Ovviamente, essendo uno schema iniziale, questo non era ottimale; dunque, si è svolta una fase di progettazione logica in cui si sono raffinati i dati utilizzati in precedenza rendendo lo schema corretto e completo. In ultima istanza, si è proceduto ad eseguire quella che viene definita traduzione, ovvero la creazione, a partire dalle risorse ottenute, di un modello di dati composto da entità e relazioni interconnesse tra loro mediante tabelle.

Le fasi sopra descritte vengono trattate più nel dettaglio nelle seguenti sezioni.

3.2 Modellazione dei dati

A partire dalle interviste e dalle informazioni raccolte sul campo si sono iniziati a raggruppare i primi dati; in particolare si è scelto di creare il database seguendo la traccia usata per la ricerca dei requisiti e dei casi d'uso del gestionale, cioè considerando l'intera società divisa nelle tre macro aree: gestione degli iscritti, gestione dell'inventario e gestione degli eventi sportivi.

Avendo già molti dati a disposizione, si è cominciato a vedere quali entità fossero di fondamentale importanza e, sopratutto, potessero essere considerate come entità, ovvero concetti talmente importanti per il gestionale che posseggono attributi, proprietà e interagiscono, a loro volta, con altre entità. Analizzando le informazioni a disposizione si sono identificate le seguenti entità fondamentali:

- *Merc*: tutti gli oggetti che la società vende e possono essere prenotati dagli iscritti e non.
- *Attrezzatura*: tutti gli oggetti che vengono usati durante gli allenamenti.
- *Partita*: rappresenta gli eventi sportivi a cui ogni squadra della società prende parte.
- *Allenamento*: rappresenta le sedute di allenamento che ogni squadra svolge durante la settimana, con le relative differenze dovute all'età.
- *Categoria*: rappresenta tutti di gruppi, giocatori e non, che fanno parte della società. Vengono considerati solo gruppi di iscritti che svolgono un ruolo attivo nella società.
- *Iscritto*: rappresenta un tesserato della società.

Oltre a queste concetti, sono stati presi in considerazione anche le seguenti entità, che non hanno un ruolo principale, ma sono complementari e di supporto a quelle già esposte:

- *Evento*: generico evento della società; è una generalizzazione di partita e allenamento.
- *Formazione*: schieramento della squadra in una partita.
- *Certificato medico*: certificato che ogni atleta deve avere per giocare e allenarsi.
- *Oggetto*: tutti gli oggetti che la società ha divisi. in merc e attrezzatura, dei quali è una generalizzazione.

A questo punto, tutti i dati necessari per la base di dati sono stati modellati; ovviamente nel corso della progettazione alcune entità verranno modificate e altre magari aggiunte in base a dei criteri che poi verranno esaminati; tuttavia l'ossatura fondamentale sarà composta dalle sopracitate entità.

3.3 Schema concettuale

Trovati i dati necessari, bisogna organizzarli in uno schema concettuale; questo viene così definito perchè cerca di modellare attraverso collegamenti dotati di cardinalità, detti relazioni, tutti i collegamenti logici che nella realtà i concetti descritti sopra, definiti come entità, possono avere. Infatti questo schema si chiama entità-relazione perchè con queste due componenti fondamentali, descrive e modella tutta la logica del database.

Sono stati dati dei criteri per la creazione dello schema così da avere una traccia da seguire e una realizzazione migliore, tale attività viene, anche, definita come analisi di qualità perchè si vanno a descrivere quali proprietà deve avere lo schema. Queste ultime sono:

- *Correttezza*: non sono presenti errori sintattici o semantici, lo schema utilizza correttamente le proprie entità e le proprie relazioni.
- *Completezza*: sono stati trattati con efficacia tutti gli aspetti derivanti dalla descrizione della società e dalle funzionalità richieste dal funzionale.
- *Leggibilità*: si è cercato di rendere leggibile lo schema mettendo al centro l'entità fondamentale, ovvero le *categorie*. Successivamente si è pensato di dividere lo schema in diverse zone in base alle funzionalità, raggruppando, così, di conseguenza, le entità correlate e/o simili.
- *Minimalità*: lo schema risulta minimale, dato che non presenta cicli o ridondanze.

Prima di creare lo schema vero e prorpio si sono realizzate due tabelle che, nel loro complesso, formano il dizionario dei dati. Nella prima si sono espresse tutte le entità coinvolte, dandone una breve descrizione, ma, soprattuto, definendone, per ognuna, gli attributi e le chiavi. Le chiavi sono delle particolari proprietà delle entità che permettono di identificarle e, dunque, devono essere uniche. Il Dizionario delle Entità è riportato nella Tabella 3.1.

Tabella 3.1: Dizionario delle Entità

CONCETTO	DESCRIZIONE	ATTRIBUTI	IDENTIFICATORI
Evento	Generico evento della società	data(data), note(testo), codice(stringa)	codice
Allenamento	Evento in cui le squadre si allenano	presenze(numerico), preparazione atletica(numerico), attacco(numerico), difesa(numerico), morale(numerico)	codice(Evento)
Partita	Evento in cui le squadre giocano contro altre squadre avversarie	punteggio(numerico), tempo effettivo(numerico), placcaggi(numerico), palloni recuperati(numerico), palloni persi(numerico), passaggi totali(numerico), passaggi efficaci(numerico)	codice(Evento)
Formazione	Schieramento della squadra in una partita	numero di maglia(numerico)	-
Iscritto	Regolarmente tessarato con la società	assicurazione(bool), quota(bool), modello 12(bool), blsd(bool), socio(bool), email(stringa), num. di telefono(numerico), data di nascita(data), nome(stringa), cognome(stringa), codice fiscale(stringa)	codice fiscale
Certificato medico	Certificato che ogni atleta deve avere per giocare e allenarsi	data di scadenza(data)	-
Categoria	Gruppo di ragazzi che, per età, giocano nella stessa squadra, o gruppo di persone con lo stesso ruolo nella società	nome(numerico), numero iscritti(numerico)	nome
Attrezzatura	Tutto ciò che è usato negli allenamenti	-	codice(Oggetto)
Merc	Tutto ciò che è venduto dalla società	link(stringa), prezzo(numerico)	codice(Oggetto)
Oggetto	Tutti gli oggetti che la società ha, divisi in Merc e Attrezzatura da campo	codice(stringa), quantità(numerico), nome(stringa)	codice

L'altra tabella, invece, è dedicata alle relazioni che coinvolgono le entità. Si compone di quattro colonne che sono: nome della relazione, descrizione, entità che partecipano alla relazione e eventuali attributi che le relazioni hanno. Il Dizionario delle Relazioni è riportato nella Tabella 3.2

Tabella 3.2: Dizionario delle Relazioni

RELAZIONE	DESCRIZIONE	ENTITÀ COINVOLTE	ATTRIBUTI
Usata da	Lega l'utilizzo delle attrezzature alle categorie di appartenenza	Attrezzatura, Categoria	-
Appartiene	Lega un iscritto alla categoria di appartenenza	Iscritto, Categoria	-
Possiede	Lega un iscritto al suo certificato medico	Iscritto, Certificato medico	-
Gioca	Lega una formazione alla relativa partita evidenziando il ruolo in campo di ogni giocatore tramite la maglia	Partita, Iscritto	num. maglia
Partecipa	Lega una categoria alla possibile partecipazione ad un generico evento	Evento, Categoria	-

Adesso sono state introdotte tutte le informazioni necessarie per la creazione dello schema concettuale. Nelle creazione dello stesso si è proceduto a inserire le cardinalità nelle relazioni, queste indicano il numero massimo e minimo di istanze di una certa entità che possono partecipare in una relazione. Le cardinalità si dividono in tre: *uno a uno*, *uno a molti*, e *molti a molti*, a volte compare anche lo 0 e sta ad indicare che una certa entità può anche non prendere parte a quella relazione.

Lo schema concettuale ultimato viene riportato in figura 3.1

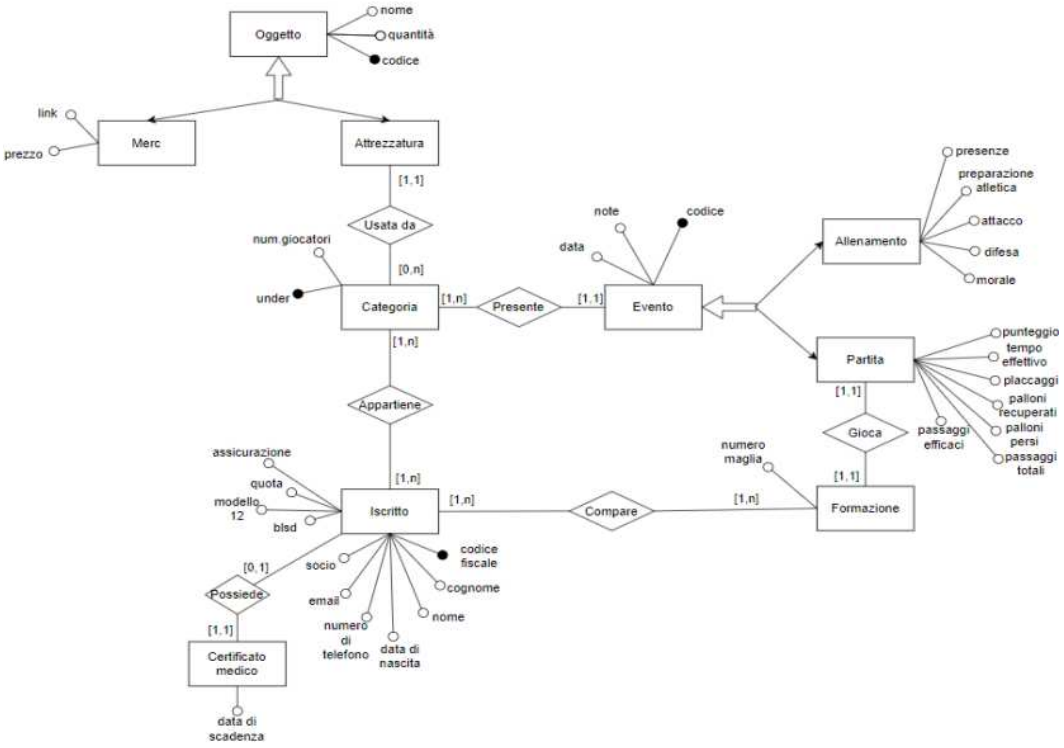

Figura 3.1: Schema concettuale

Tabella 3.3: Tabella delle operazioni

ID	OPERAZIONE	FREQUENZA(settimanale)
1	Inserimento nuovo allenamento	36
2	Inserimento nuova partita	9
3	Inserimento nuovo iscritto	1 volte al mese
4	Aggiunta iscritti nelle varie categorie	1
5	Modifica certificati	1
6	Aggiunta Attrezzature	1 volta al mese
7	Aggiunta partite	9
8	Aggiunta merc	1
9	Cancellazione Iscritti	1 volta al mese
10	Cancellazione attrezzature	1 volta al mese
11	Cancellazione merc	1 volta al settimana
12	Modifica merc	1 volta al mese
13	Modifica Attrezzature	1 volta al mese
14	Calcolo iscritti tot	1
15	Calcolo medie punteggi allenamenti	1
16	Calcolo medie stats partite	1 volta al mese
17	Controllo numeri di maglia nelle partite	5
18	Calcolo giocatori tot. nella società	1
19	Controllo quantità di oggetti della società	1 volta al mese
20	Controllo di quanti iscritti non hanno pagato la quota o non hanno il modello 12	1
21	Controllo numero di giocatori per categoria	1

3.4 Progettazione logica

Questa fase di progettazione della base di dati consiste in una rifinitura di ciò che è stato elaborato in precendenza. In particolare, viene esaminata la struttura dello schema concettuale e si verifica che non ci siano informazioni incomplete o, al contrario, troppo complesse o ridondanti e se ci sono parti che possono essere rielaborate per ottenere poi un modello relazionale più efficace. Tutto ciò viene verificato attraverso 4 fasi fondamentali su cui ci si soffermerà nel dettaglio.

3.4.1 Analisi ridondanze

In questa fase si controlla se ci sono attributi che risultano ridondanti, ovvero non necessari, o comunque non rappresentano la soluzione più efficace per eseguire certi gruppi di operazioni del database. Per eseguire questi controlli si fa uso di due tabelle: la Tabella delle operazioni (Tabella 3.3), in cui vengono riassunte tutte le azioni che il database può effettuare e le sue relative frequenze, e la tabella dei volumi (Tabella 3.4) , in cui, invece, si enumerano tutti i concetti esemplificati dalla base di dati con i corrispettivi volumi e la loro tipologia (entità o relazione).

Usando queste tabelle si svolgono una serie di operazioni di verifica per controllare se ci sono attributi ridondanti. A seconda del caso si possono avere due scenari differenti:

1. Si prende in considerazione l'attributo che si suppone ridondante e si controlla quanti accessi in lettura e/o scrittura devono essere fatti al database per eseguire quelle operazioni che coinvolgono l'entità scelta. Fatto questo si controlla quanti accessi

Tabella 3.4: Tabella dei volumi

ID	CONCETTO	TIPO	VOLUME
1	Merc	E	200
2	Attrezzatura	E	34
3	Categoria	E	12
4	Iscritto	E	150
5	Certificato medico	E	121
6	Partita	E	155
7	Allenamento	E	620
8	Usata da	R	34
9	Appartiene	R	159
10	Possiede	R	121
11	Partecipa	R	775
12	Gioca	R	2904

Tabella 3.5: Operazione 14 (con ridondanza)

Concetto	Volume	Accessi	Tipo
Categoria	12	1	L

servirebbero se l'entità ridondante non ci fosse. Nei calcoli si considera sempre che un accesso in scrittura equivale a 2 in lettura. Alla fine si confrontano i risultati ottenuti e si verifica quale delle due soluzioni è più efficace e veloce.

2. Si procede al contrario, ovvero si controlla se, per un certo numero di operazioni, potrebbe essere più semplice aggiungere un attributo ad un'entità o ad una relazione piuttosto che evitare di averla.

Nel caso di questo gestionale si sono controllate due diverse situazioni appartenenti ognuna ad un caso.

Inanzitutto si è voluto controllare che l'attibuto *numero iscritti* dell'entità *Categoria* non fosse ridondante. Di seguito vengono riportati i calcoli effettuati nelle Tabelle 3.5-3.10.

Costo totale operazioni con ridondanza = 3 L/settimana
Costo totale operazioni senza ridondanza = 6 L/settimana
Dunque è chiaro che convenga mantenere l'attributo numero iscritti.

L'altra analisi effettuata, che ricade nel secondo caso possibile, riguarda l'entità *Evento*. Si è pensato che potrebbe essere conveniente aggiungere un attributo *under* per indicare quale categoria partecipa a quell'evento. Di seguito vengono riportati i calcoli effettuati nelle Tabelle 3.11-3.14

Costo totale operazioni con ridondanza = 1643 L/settimana
Costo totale operazioni senza ridondanza = 659 L/settimana
L'operazione 16 è stata convertita da L/mese a L/settimana facendo il totale diviso quattro in entrambi i casi.

Tabella 3.6: Operazione 14 (senza ridondanza)

Concetto	Volume	Accessi	Tipo
Iscritto	150	1	L
Appartiene	159	1	L

Tabella 3.7: Operazione 18 (con ridondanza)

Concetto	Volume	Accessi	Tipo
Categoria	12	1	L

Tabella 3.8: Operazione 18 (senza ridondanza)

Concetto	Volume	Accessi	Tipo
Iscritto	150	1	L
Appartiene	159	1	L

Tabella 3.9: Operazione 20 (con ridondanza)

Concetto	Volume	Accessi	Tipo
Categoria	12	1	L

Tabella 3.10: Operazione 20 (senza ridondanza)

Concetto	Volume	Accessi	Tipo
Iscritto	150	1	L
Appartiene	159	1	L

Tabella 3.11: Operazione 15 (con ridondanza)

Concetto	Volume	Accessi	Tipo
Allenamento	620	1	L

Tabella 3.12: Operazione 15 (senza ridondanza)

Concetto	Volume	Accessi	Tipo
Categoria	12	1	L
Partecipa	775	1	L
Allenamento	620	1	L

Tabella 3.13: Operazione 16 (con ridondanza)

Concetto	Volume	Accessi	Tipo
Partita	155	1	L

Tabella 3.14: Operazione 16 (senza ridondanza)

Concetto	Volume	Accessi	Tipo
Categoria	12	1	L
Partecipa	775	1	L
Partite	155	1	L

Anche in questo caso risulta che l'opzione con la ridondanza è più efficace; questa volta, però, dobbiamo aggiungerle l'attributo analizzato. Si può notare che le operazioni che coinvolgevano *under* erano anche altre, ovvero quelle di aggiunta, ma risulta evidente che l'attributo, seppur ridondante, è sicuramente utile ai fini dell'efficienza del database.

3.4.2 Eliminazione delle gerarchie

Nel creare uno schema concettuale si creano delle interdipendenze tra entità, in particolare possono esserci delle *generalizzazioni*. Queste si creano quando ci sono diverse entità che condividono le stesse proprietà e quindi, concettualmente, si crea un' entità *genitore* che comprende gli attributi comuni delle *figlie*. Ciò a livello logico aiuta la comprensione dello schema, tuttavia non è traducibile in un modello relazionale e quindi bisogna eliminare questi tipi di gerarchie. Per fare ciò ci sono 3 possibilità:

1. Se le entità *figlie* sono indipendenti e hanno senso da sole allora si mantengono e viene eliminata la generalizzazione. Spesso si utilizza questa soluzione quando ogni entità *figlia* prende parte ad operazioni diverse.
2. Se l'entità *genitore* è il fulcro di tutte le operazioni che coinvolgono le ramificazioni, allora si mantiene solo la generalizzazione eliminando le *figlie*
3. Se sia la generalizzazione che le ramificazioni sono importanti, cioè ci sono operazioni che devono essere svolte dalle specifiche entità e delle altre invece che possono essere agilmente svolte dall'entità *genitore*, si può decidere di sostituire la generalizzazione con delle relazioni uno a uno tra il *genitore* e le *figlie*

Nel caso di questo gestionale, lo schema concettuale contiene solo due generalizzazioni, ovvero *Oggetto* e *Evento*. Esse sono state gestite in questo modo:

- *Oggetto*: si è diviso in *Merc* e *Attrezzatura*. Si deciso di eliminare l'entità padre perchè entrambe le entità *figlie* hanno un significato proprio anche senza la generalizzazione. Inoltre, ci si trova in una situazione in cui *Attrezzatura* non ha degli attributi aggiuntivi rispetto ad *Oggetto* e anche *Merc*, comunque, ne presenta solo 2 in più.
- *Evento*: originariamente era diviso in *Allenamento* e *Partita*, ma, in questo caso, non può essere applicato il ragionamento precedente. L'ipotetica eliminazione dell'attributo *padre* porterebbe ad un'eredità eccessiva, sia in termini di relazioni che in termini di attributi, mentre, invece, l'eliminazione delle entità figlie porterebbe ad un'ambiguità nel senso della parola *Evento*. Dunque si è deciso di mantenere la struttura invariata aggiungendo delle relazioni aggiuntive di "specificazione".

3.4.3 Accorpamento e partizionamento

Controllando lo schema concettuale, risulta inopportuno accorpare entità o relazioni perchè non ce n'è nessuna che risulta troppo divisa nei concetti.

Viceversa è opportuno partizionare alcune parti dello schema; in particolare alcuni attributi di *Partita* sono molto specifici e non vengono usati in ogni istanza dell'entità; dunque si è modificata l'entità come mostrato in Figura 3.2

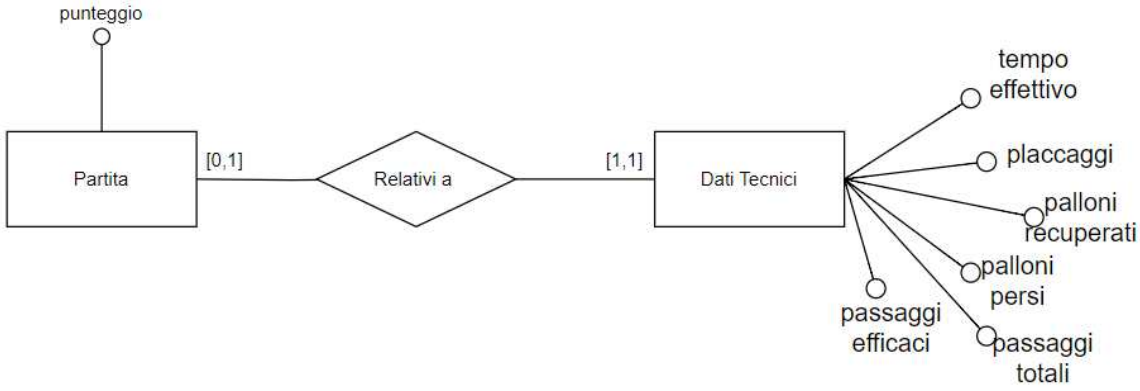

Figura 3.2: Partizionamento di *Partita*

Anche in *Iscritto* ci sono attributi che vengono usati sporadicamente e a fini esclusivamente informativi; questo vengono partizionati come in Figura 3.3.

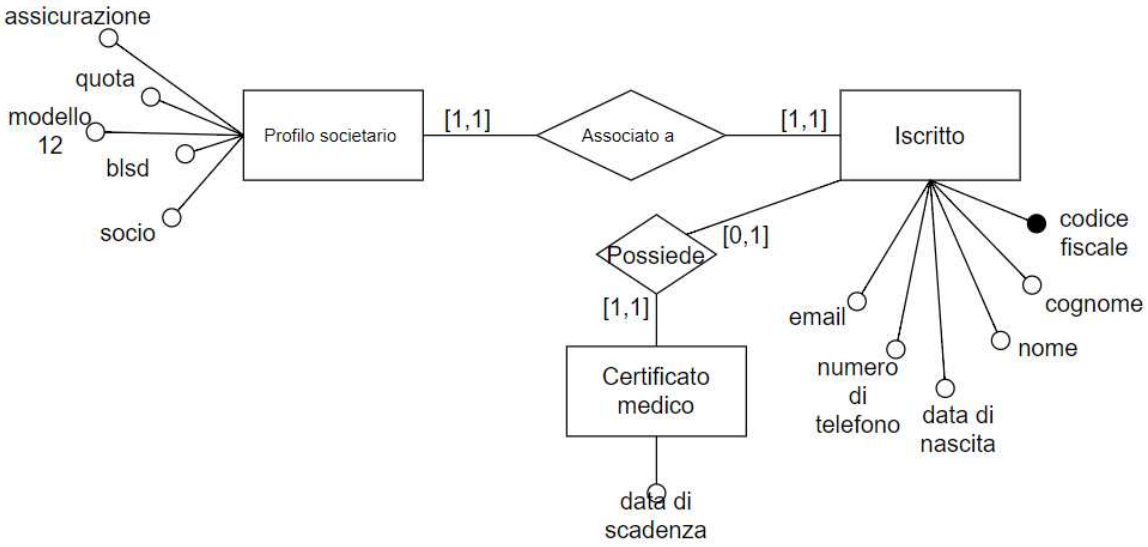

Figura 3.3: Partizionamento di *Iscritto*

3.4.4 Eliminazione degli attributi multivalore

Spesso nella creazione di schemi concettuali si creano delle situazioni per cui risulta logico formare degli attributi che possono avere più di un valore contemporaneamente; per esempio gli indirizzi di una stessa persona possono essere molteplici. Questi, seppur semplificano la comprensione del funzionamento della base di dati, non possono essere tradotti in un modello relazionale. Nel caso della base di dati che si sta analizzando non ci sono attributi di questo tipo.

3.5 Traduzione verso il modello relazionale

Una volta completata questa analisi dello schema concettuale si ottiene lo schema completo, che è quello mostrato in Figura 3.4

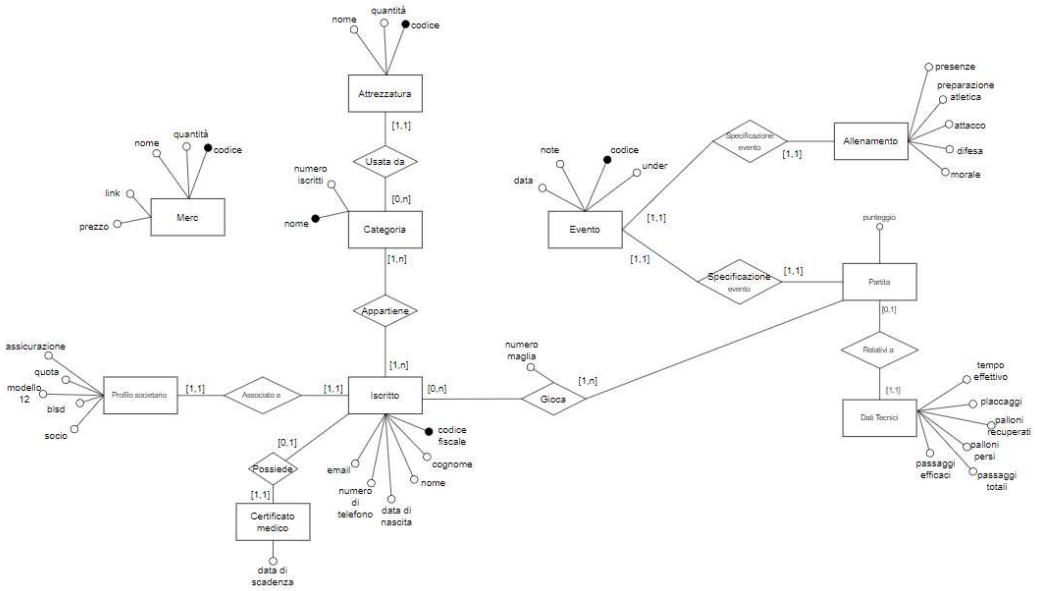

Figura 3.4: Schema completo

Il modello relazionale, non è altro che una mappatura delle entità e delle relazioni trovate fino ad ora, in delle tabelle. Le colonne di queste rappresentano gli attributi, e le righe invece sono gli elementi. La particolarità è che ogni tabella deve avere almeno una chiave, ovvero un attibuto univoco tramite il quale è possibile riasalire specificatamente ad ogni elemento. Le chiavi possono essere anche esterne, ovvero condividono il valore con altre tabelle creando, così, degli inter vincoli tra le tabelle e gli stessi elementi. Nel tradurre in modello relazionale bisogna fare attenzione alle relazioni, perchè, a seconda della cardinalità, si può pensare o meno di mapparle in tabelle specifiche. Si usa fare questa operazione quando una relazione tra due o più entità è di tipo molti a molti, mentre, invece, nel caso di relazione uno a molti si utilizzano delle chiavi esterne solo nell'enità con cardinalità massima pari a molti. Nelle Tabelle 3.15 e 3.16 vengono mostrate le traduzioni effettuate per lo schema fino ad ora descritto.

Tabella 3.15: Schema logico del database

Entità/Relazione	Attributi
Merc	codice, quantità, nome, link, prezzo
Attrezzatura	codice, quantità, nome
Categoria	nome, numero iscritti
Iscritto	codice fiscale, nome, cognome, data di nascita, telefono, email
Certificato medico	codice fiscale, data di scadenza
Profilo societario	codice fiscale, assicurazione, quota, modello 12, BLSD, socio
Evento	codice, under, note, data
Allenamento	codice, under, presenze, preparazione atletica, attacco, difesa, morale
Partita	codice, punteggio
Dati tecnici	codice, tempo effettivo, placcaggi, palloni recuperati, palloni persi, passaggi totali, passaggi efficaci

Tabella 3.16: Traduzione e Vincoli di Riferimento

Traduzione	Vincoli di Riferimento
Merc(codice, quantità, nome, link, prezzo)	-
Attrezzatura(codice, quantità, nome)	-
Categoria(nome, numero iscritti)	-
Iscritto(codice fiscale, nome, cognome, data di nascita, telefono, email)	-
Certificato medico(codice fiscale, data di scadenza)	codice fiscale → Iscritto.codice fiscale
Profilo societario(codice fiscale, assicurazione, quota, modello 12, BLSD, socio)	codice fiscale → Iscritto.codice fiscale
Evento(codice, under, note, data)	-
Allenamento(codice, under, presenze, preparazione atletica, attacco, difesa, morale)	codice → Evento.codice
Partita(codice, punteggio)	codice → Evento.codice
Dati tecnici(codice, tempo effettivo, placcaggi, palloni recuperati, palloni persi, passaggi totali, passaggi efficaci)	codice → Partita.codice
Gioca(numero di maglia, codice fiscale, codice)	codice → Partita.codice, codice fiscale → Iscritto.codice fiscale
UsataDa(nome, codice)	nome → Categoria.nome, codice → Attrezzatura.codice

3.6 Conclusioni

La sfida di progettazione proposta dalla realtà modellata dal gestionale ha presentato diverse situazioni complesse, o comunque da risolvere, per avere una corretta modellazione della base di dati. Nella fase di progettazione logica, sopratutto, non è stato semplice risolvere i problemi delle gerarchie e i partizionamenti. Una corretta e approfondita analisi dello schema concettuale aiuterà, poi, a sviluppare in maniera agile ed efficace il software con la certezza di avere un database su cui appoggiarsi che modella in maniera efficente la schema dei dati.

Nel dettaglio, per tale sistema gestionale è stato utilizzato un database MySql appoggiandosi su dei server creati localmente, poichè la realtà sportiva in cui doveva essere applicato lo consentiva e non era opportuno appesantire la società con dei costi, quali server esterni, sia virtuali che fisici.

CAPITOLO 4

Progettazione della componente applicativa

In questo capitolo vengono descritte le parti che compongono il programma. Nello specifico, si fa una panoramica piuttosto dettagliata del percorso che ha portato alle scelte architetturali, funzionali e, anche, di design. Si parte col parlare dell'architettura, per poi esporre una serie di diagrammi, fondamentali in fase di progettazione, che hanno portato alla costruzione delle componenti applicative del programma; infine, si descrive l'interfaccia utente e le sue componenti. Ogni scelta, di ognuna delle tre tipologie elencate, è stata commentata e descritta per farne comprendere meglio lo scopo. Alla fine del capitolo l'attenzione sarà, invece, posta su come si è garantita la sicurezza e l'affidabilità del sistema, quindi si parlarà di gestione delle eccezioni e prevenzione degli errori.

4.1 Architettura del sistema

Il software è stato creato, ideato, e progettato, su misura, in base alle richieste della società di Falconara. L'architettura, dunque, rispecchierà le esigenze riscontrate in fase di acquisizione delle informazioni e durante la creazione della base di dati.

L'idea, che è stata perseguita durante la progettazione e l'implementazione, è stata quella di ideare un software che, dal punto di vista strutturale, non risultasse troppo ingombrante o, comunque, di difficile comprensione e manutenzione, considerato che, una volta messo a regime, il software, essendo creato per una società sportiva e non per un'azienda, non avrà controlli troppo frequenti.

Per costruire un software pulito ed efficente, si è optato per dividere le classi in diverse *directory* in base al loro scopo. Nel programma troviamo una *directory* per le classi, chiamata *Class*, al cui interno troviamo mappate tutte le tabelle che sono presenti nel database, sia entità vere e proprie sia le relazioni trasformate in tabelle durante la traduzione al modello relazionale; è presente, poi, una *directory* per i gestori, chiamata *Gestori*, al cui interno troviamo tre classi, una per macro area del sistema, che si occupano di raggruppare tutte le funzioni relative a quella specifica parte di sistema. Infine è presente la *directory*, denominata *GUI*, in cui sono raggruppati tutti quei file che modellano l'interfaccia grafica del gestionale; in questi file possono possono essere presenti metodi che non implementano nessuna funzionalità particolare, ma svolgono delle azioni utili al corretto ed efficente funzionamento dell'interfaccia utente. Abbiamo poi una *directory* per i test e, infine, non raggruppati in delle directory specifiche, le classi *main* e *database*.

In questa sezione si porrà l'attenzione sulle scelte fatte dal punto di vista architetturale, ma in parte anche funzionale, e, in particolar modo, dei motivi che hanno portato a prendere queste decisioni.

4.1.1 Analisi delle funzionalità del software

Il sistema implementa diverse funzioni, tutte create seguendo i casi d'uso precedentemente descritti. Nelle funzionalità implementate possono esserci, rispetto alle considerazioni fatte durante la scrittura dei casi d'uso, alcune sottili differenze dovute, però, a problemi di natura prettamente strutturale e architetturale; ciò significa che, appunto, tutto ciò che il sistema gestionale può fare è descritto nei casi d'uso, poi la creazione dei singoli metodi può aver subito leggere modifiche dovute a situazioni incontrate in fase di progettazione o implementazione.

Le funzionalità principali, che il sistema offre, sono implementate tramite delle chiamate, ovvero delle *query*, al database. Queste vengono raggruppate nelle classi contenute nella cartella *Gestori*.

La scrittura, e la conseguente implementazione, sono frutto di una serie di analisi sui singoli metodi, fatte in fase di progettazione, volte a descrivere, in maniera chiara e corretta, il loro funzionamento. Nello specifico, sono stati usati 3 tipi di diagrammi per le funzioni e un diagramma per le classi; questa tipologia di progettazione, basata sul principio *Plan-do-check-act*, usufruisce di questi schemi per avere una visione completa di ciò che si sta sviluppando, mettendone in risalto quindi eventuali incongruenze o errori, nella fase di *plan*; questi diagrammi, inoltre, nel corso dello sviluppo dell'app permettono di avere una sorta di "traccia" su cui basarsi, per verficare l'andamento del progetto e la coerenza con ciò che era stato stabilito in fase di pianificazione.

I diagrammi vengono spiegati nel dettaglio di seguito; si fa notare che non sono stati creati schemi per ogni singolo metodo del gestionale per motivi principalmente di utilità, in quanto, seppur divisi nelle varie aree, molti metodi eseguono le stesse azioni ma su differenti tipi di oggetti, dunque ci sarebbe stata un'inutile ridondanza. Si è, pertanto deciso di dare maggior attenzione ai metodi più ostici ad essere implementati, evitando le ripetizioni.

Diagrammi di sequenza

La prima tipologia di diagrammi utilizzata è quella dei *diagrammi di sequenza*; questi servono a mostrare come le varie entità, sia interne sia esterne al programma, come la base di dati, collaborano e comunicano durante l'esecuzione di un particolare scenario, in questo caso, di un caso d'uso.

In breve, questi diagrammi rappresentano lo scambio di comunicazioni tra diversi *oggetti*, ognuno dei quali ha una propria *linea di vita*, definita da una linea verticale tratteggiata; essa rappresenta il tempo di vita di un oggetto nel diagramma. Gli oggetti comunicano tra loro usando dei messaggi che sono definiti usando linee orizzontali continue o tratteggiate, a seconda che si tratti di un messaggio *sincrono*, e quindi l'oggetto si blocca fino al ricevimento di una risposta, o *asincrono*, dove, invece, l'oggetto continua la sua esecuzione senza attendere. In diversi diagrammi compaiono dei rettangoli con scritto *alt* oppure *opt*, che simboleggiano rispettivamente una sequenza di eventi alternativa e una sequenza condizionale. Ci sono anche altri tipi di strumenti utilizzabili in questi diagrammi che, tuttavia, nella presente trattazione risultano inutilizzati o di semplice comprensione.

Si noti che i nomi utilizzati nei vari scambi di messaggi tra entità non sempre corrispondono esattamente a quelli poi utilizzati nel programma; questo perchè si è preferito usare delle espressioni, magari più generiche, ma che spiegassero chiaramente la funzione di riferimento.

Inoltre, come già spiegato in precedenza, e, quindi, per gli stessi motivi, gli scenari descritti non sono tutti, ma solo i principali o più complessi da comprendere.

Di seguito una lista di riferimenti ai vari diagrammi utilizzati per la progettazione:

- la Figura 4.1 rappresenta *CRUDIscritti*;

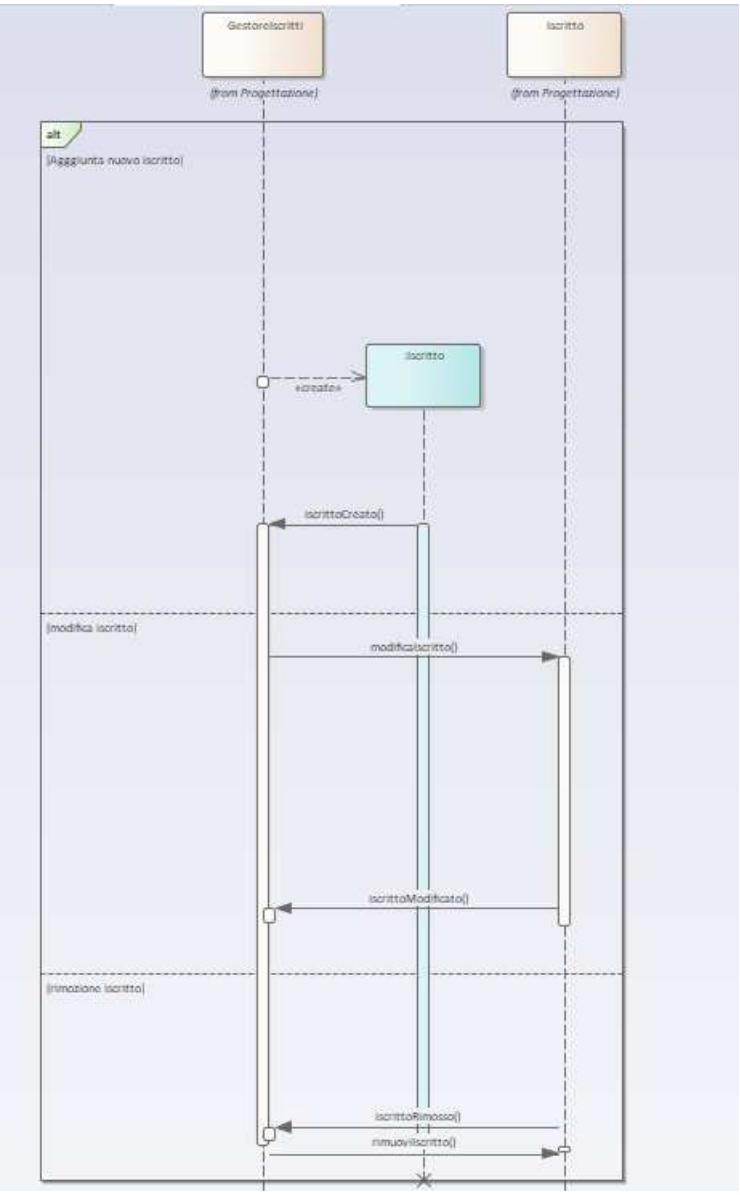

Figura 4.1: Diagramma di sequenza di CRUDIscritti

- la Figura 4.2 rappresenta *Ricerca*;
- la Figura 4.3 rappresenta *ScadenzePopUp*;
- la Figura 4.4 rappresenta *VisualizzaEvento*;
- la Figura 4.5 rappresenta *OrdinaEventi*.

Diagrammi di attività

Un'altra tipologia di diagramma utilizzata in fase di progettazione è quella dei *diagrammi di attività*. Questi servono a spiegare il comportamento e i possibili scenari dei vari processi del software, ovvero scandiscono l'ordine di operazioni che, a seconda delle scelte fatte dall'utente, alla fine, portano alla conclusione del flusso di lavoro e alla eventuale produzione di un risultato.

Per produrre questi diagrammi vengono utilizzati diversi elementi:

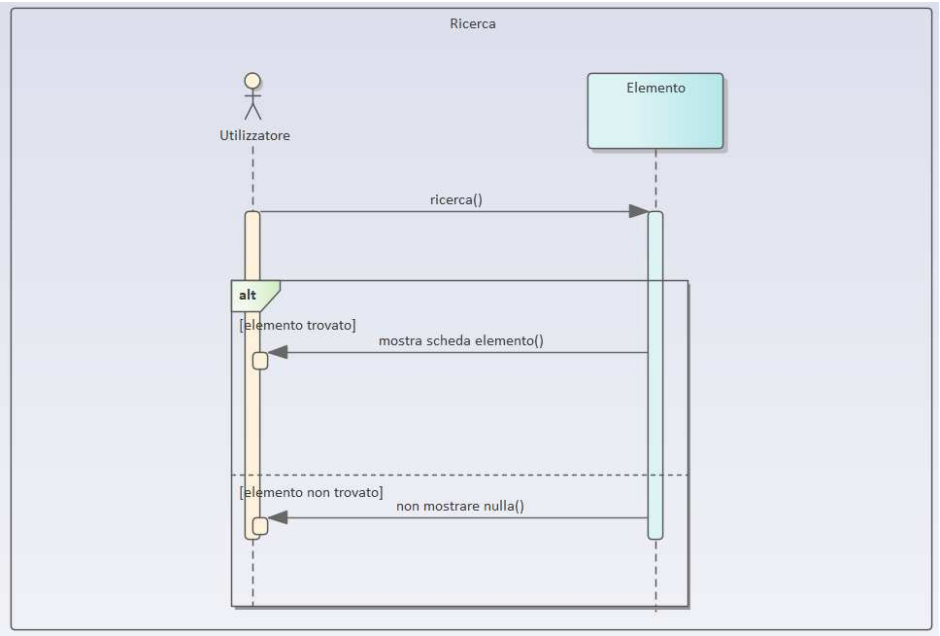

Figura 4.2: Diagramma di sequenza di Ricerca

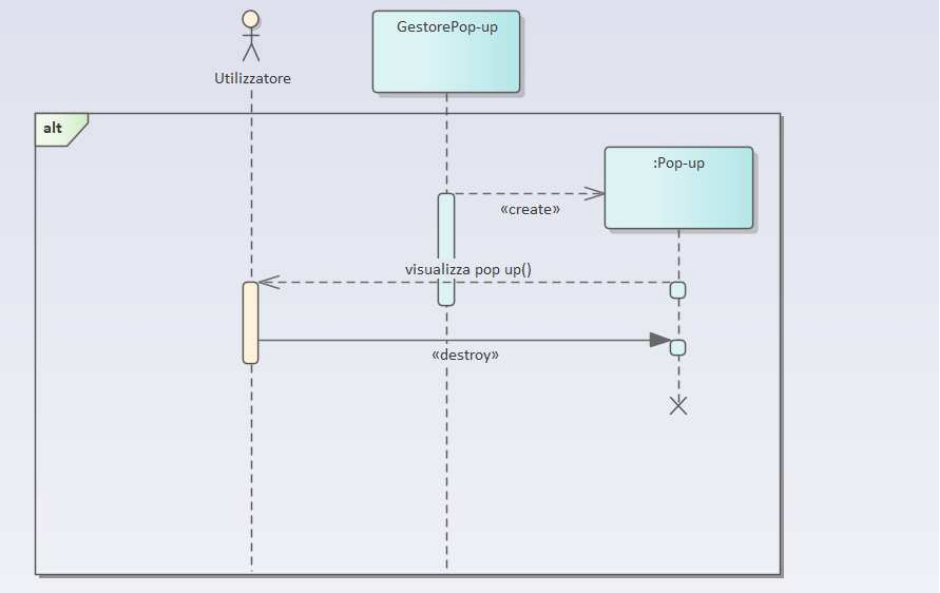

Figura 4.3: Diagramma di sequenza di ScadenzePopUp

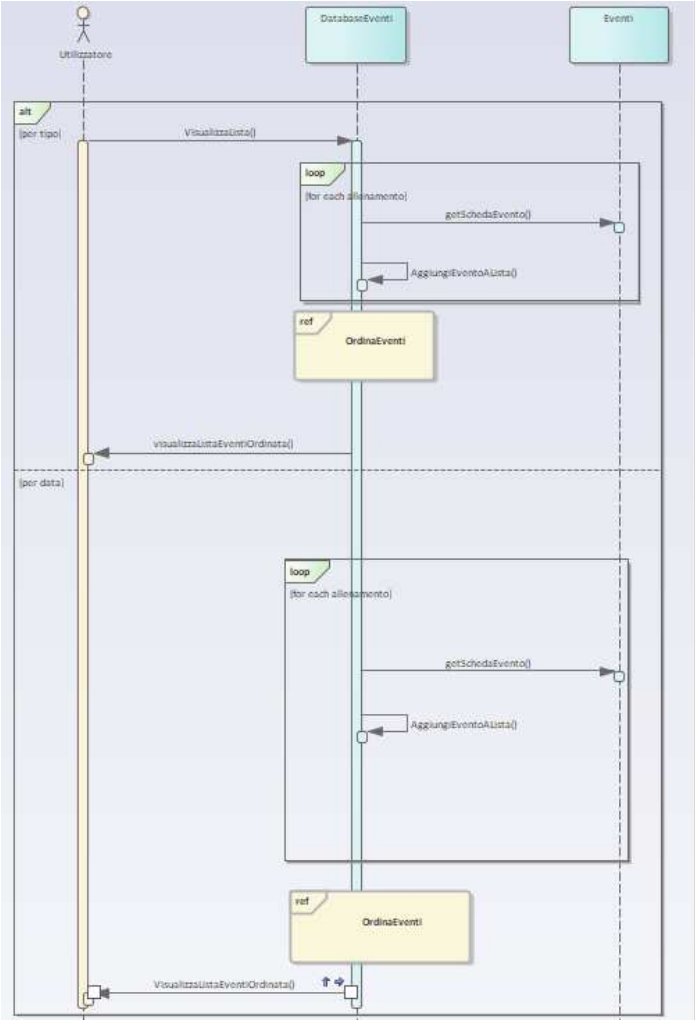

Figura 4.4: Diagramma di sequenza di VisualizzaEvento

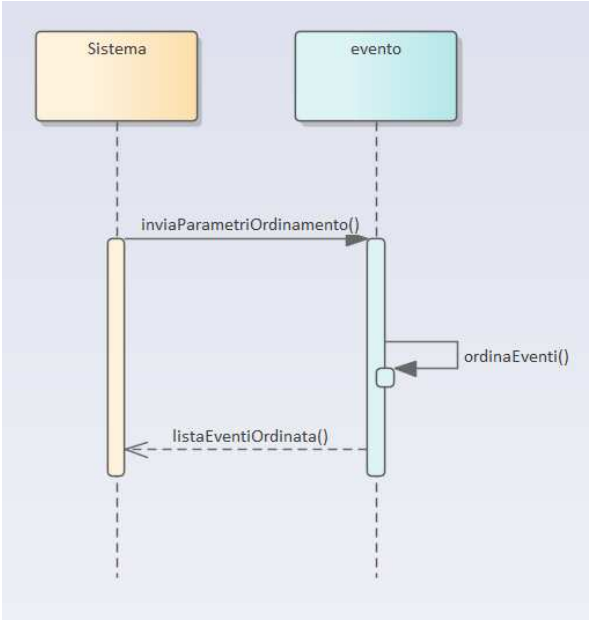

Figura 4.5: Diagramma di sequenza di OrdinaEventi

- *Attività (Activity)*: rappresenta un'azione o un'operazione che viene eseguita all'interno del processo. Può essere qualsiasi azione o passo che contribuisce al raggiungimento dell'obiettivo del processo. L'*attività* viene indicata con un rettangolo di colore arancione.
- *Stato iniziale (Initial state)* e *Stato finale (Finale state)*: il primo indicato da un pallino nero e l'altro da un pallino verde con all'interno un cerchio nero, rappresentano il punto di inizio e di fine del flusso di lavoro.
- *Transizione (Transition)*: rappresenta il passaggio da un'attività a un'altra. Viene indicato con una freccia che collega l'attività di partenza con quella di destinazione.
- *Decisione (Decision)*: rappresenta una scelta che deve essere fatta tra diverse opzioni. Ogni opzione è rappresentata come una freccia che parte dalla decisione; il nodo decisonale, invece, è un rombo verde.
- *Merge (Merge Node)*: indica il punto in cui più percorsi si uniscono in uno solo. È il complemento di una decisione.

Ci sono poi altri diversi elementi utilizzabili, ma, per questa trattazione, quelli elencati sono sufficienti.

Anche in questo caso, esattamente come nel precedente, non si sono fatti i diagrammi per ogni metodo ma si è continuata la progettazione di quelli visti in precedenza:

- la Figura 4.6 rappresenta *CRUDIscritti*;
- la Figura 4.7 rappresenta *Ricerca*;
- la Figura 4.8 rappresenta *ScadenzePopUp*;
- la Figura 4.9 rappresenta *VisualizzaEvento*;
- la Figura 4.10 rappresenta *OrdinaEventi*.

Diagramma delle classi di progettazione

Questo diagramma è stato utilizzato per avere una traccia da seguire durante l'implementazione del software. Infatti, questa tipologia di schema viene usata in fase di progettazione per mostrare le classi facenti parte del software, i loro attributi, di cui si specificano visibilità e tipo, e i loro metodi, di cui, invece, oltre a tipo e alla visibilità, si scrivono i parametri. Inoltre, è possibile descrivere le associazioni tra le varie classi definendone nome e cardinalità. Sostanzialmente, dunque, servono ad avere uno schema, seppur semplice, completo e corretto per poi procedere all'implementazione senza possibilità di errori.

Bisogna specificare, però, che questo diagramma non descrive esattamente tutte le classi che poi verranno implementate nel software, ma la maggior parte. Viene fatta una selezione per due motivi: il primo, più semplice, è per la leggibilità, in quanto questo diagramma poi, potrebbe, essere esposto al cliente che avrà interesse a capire con facilità quali classi comporanno il programma, così come altri, eventuali, collaboratori del progetto potrebbero voler vedere lo schema; il secondo motivo, invece, è che poi, in fase di implementazione, potrebbero essere necessari dei costrutti, che potrebbero essere classi, piuttosto che metodi, piuttosto che qualsiasi altro tipo di elemento, che non hanno un ruolo centrale nella descrizione dell'argomento, in questo caso un gestionale per una società sportiva, ma sono utili al corretto funzionamento del software.

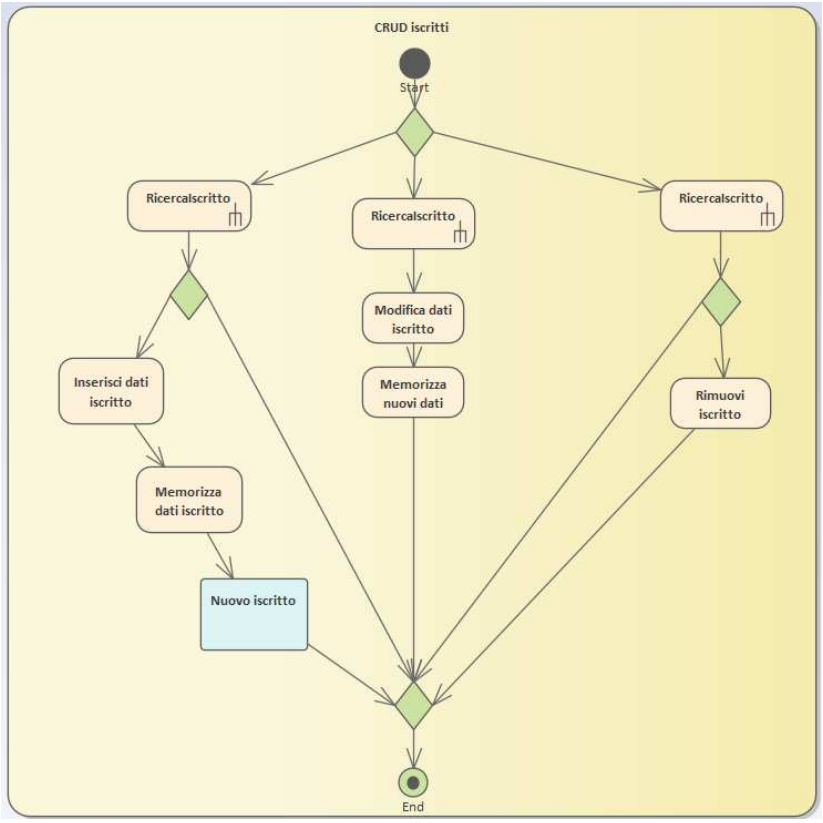

Figura 4.6: Diagramma di attività di CRUDIscritti

In questo caso, infatti, si sono omesse alcune classi, che sono presenti nel modello relazionale descritto nel Capitolo 3, in quanto ritenute superflue per capire il funzionamento del software.

Nelle Figure 4.11 e 4.12 viene presentato il *diagramma delle classi di progettazione*.

Diagrammi delle macchine a stati

Quest'utlima tipologia di diagramma serve a definire il comportamento di quelle componenti software, sia funzioni che oggetti, che possono cambiare il loro stato in risposta ad eventi o condizioni. Oltre, ovviamente, ad aiutare la progettazione e la comunicazione con gli stakeholder del progetto, servono a chiarire il comportamento delle varie parti del software in modo tale da non commettere errori di incongruenza durante la fase di implementazione.

Il diagramma si compone delle seguenti parti:

- *Stati*: gli stati rappresentano le condizioni o le fasi che un oggetto o un sistema può assumere durante il suo ciclo di vita. Ogni stato è rappresentato da un rettangolo con un nome all'interno.
- *Transizioni*: le transizioni rappresentano il passaggio da uno stato all'altro in risposta a un evento o una condizione. Sono rappresentate da frecce che collegano gli stati.
- *Eventi*: gli eventi sono stimoli o condizioni che possono scatenare una transizione da uno stato all'altro. Esse sono spesso associati alle frecce che rappresentano le transizioni.
- *Stato iniziale e finale*: rappresenta lo stato iniziale e finale del sistema che si sta esaminando.

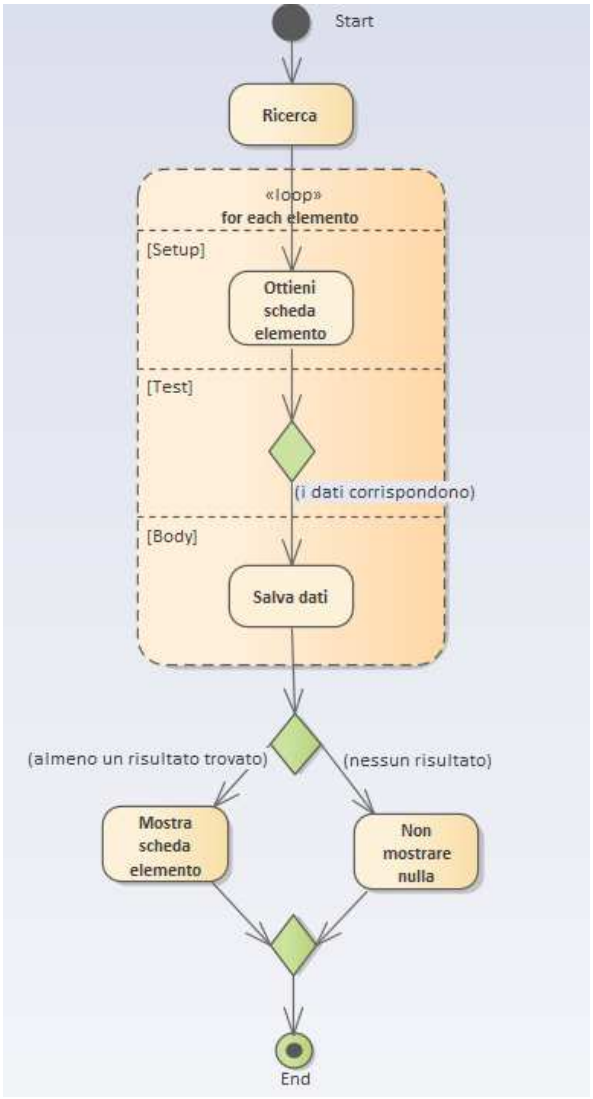

Figura 4.7: Diagramma di attività di Ricerca.

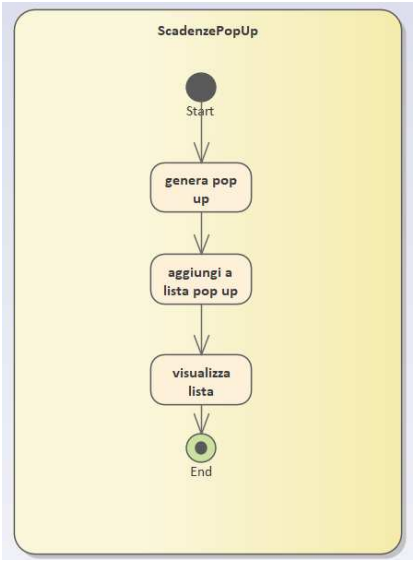

Figura 4.8: Diagramma di attività di ScadenzePopUp.

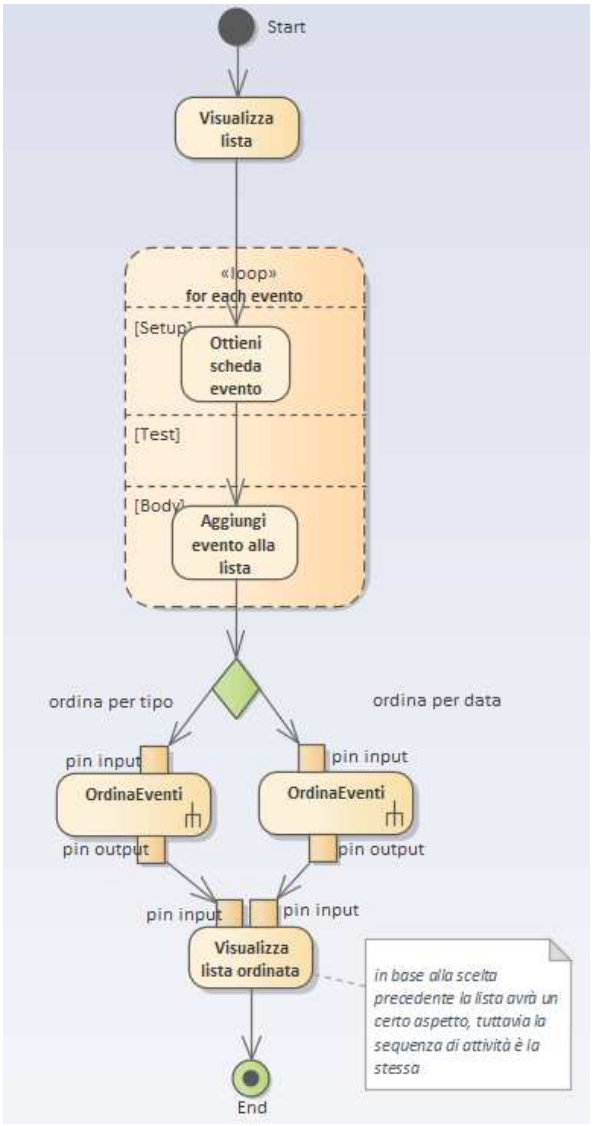

Figura 4.9: Diagramma di attvità di VisualizzaEvento.

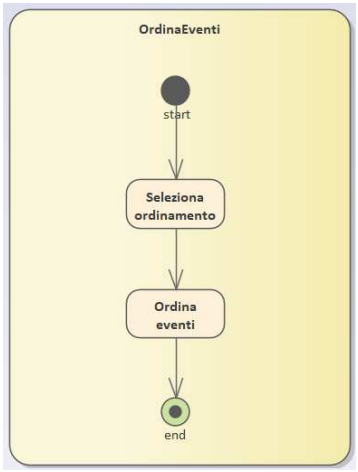

Figura 4.10: Diagramma di attività di OrdinaEventi.

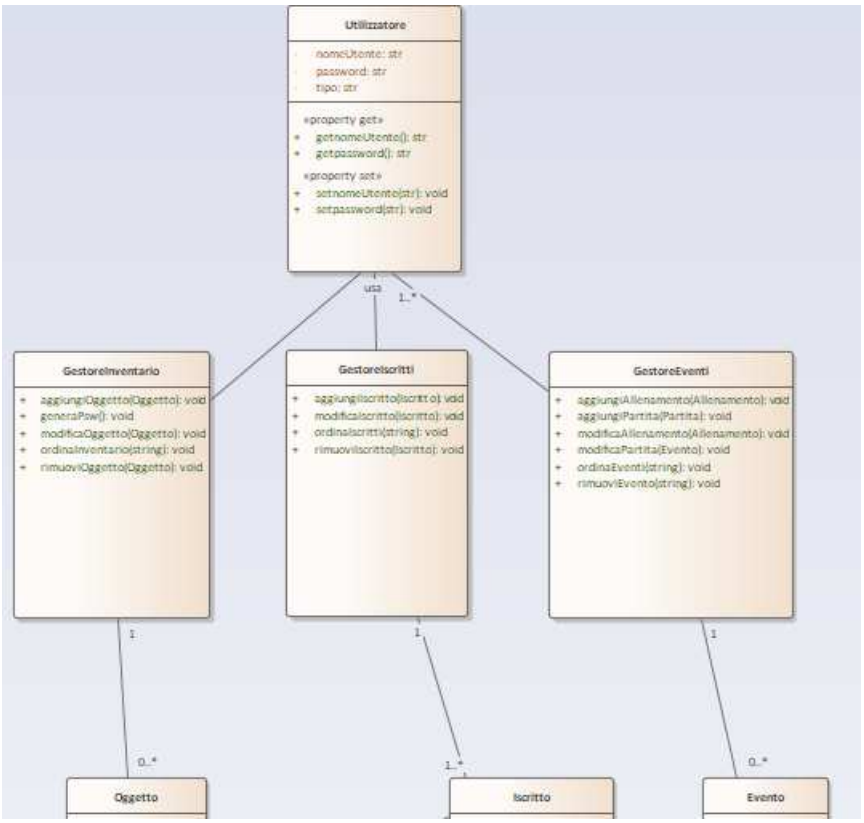

Figura 4.11: Diagramma delle classi di progettazione(prima parte)

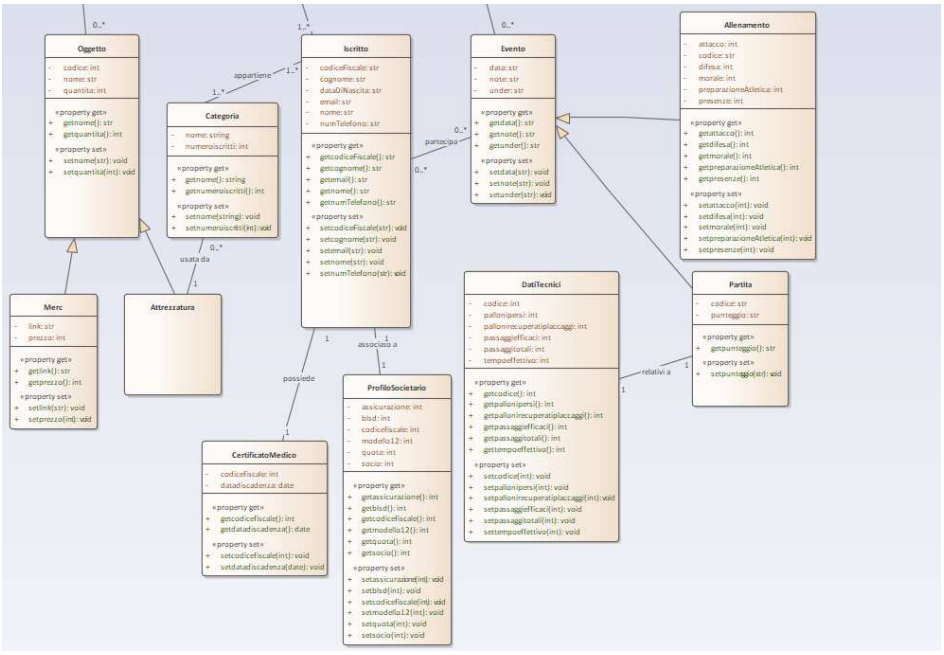

Figura 4.12: Diagramma delle classi di progettazione(seconda parte)

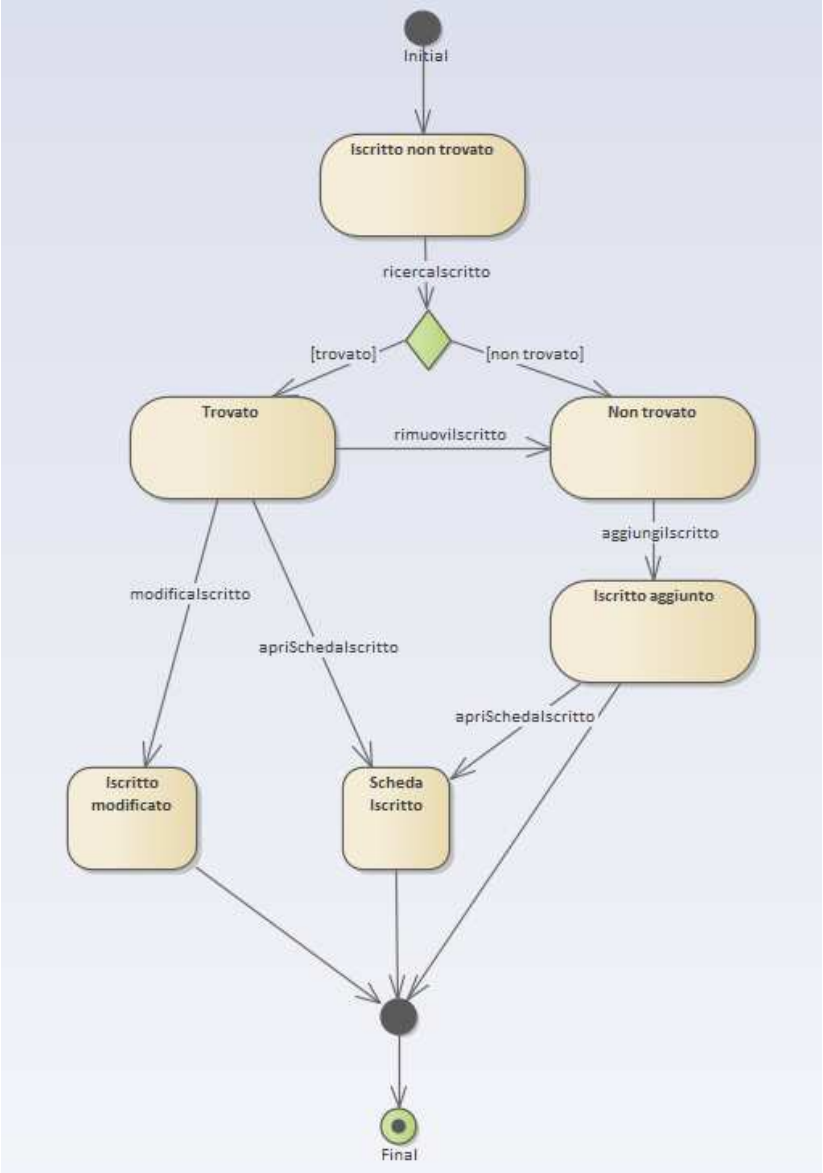

Figura 4.13: Diagramma delle macchine a stati di iscritto

- *stato di scelta*: rappresenta uno stato in cui il sistema prende una decisione; il sistema può passare a uno dei diversi stati in base a condizioni specifiche.
- *Stato di giunzione*: indica il punto in cui più transizioni si uniscono, consentendo al sistema di proseguire in uno stato successivo.

Nel caso che si sta trattando, si è deciso di utilizzare questo specifico diagramma solo per due particolari oggetti: *iscritto* e *oggetto*, come è possibile vedere, rispettivamente, dalle Figure 4.13 e 4.14. Questa scelta è dovuta al fatto che non sono presenti altri oggetti che sono tanto complessi da necessitare una descrizione più ampia utilizzando, appunto, i diagrammi degli stati. Un particolare, di cui si avrà la verifica in seguito, è che, in fase di implementazione, l'oggetto denominato *oggetto*, non verrà utilizzato, viste le analisi fatte nel Capitolo 3 di questa trattazione; tuttavia, per il compito che deve eseguire il diagramma degli stati, ovvero utilizzare questa generalizzazione per gli oggetti *merc* e *attrezzatura*, risulta essere più efficace e meno ridondante.

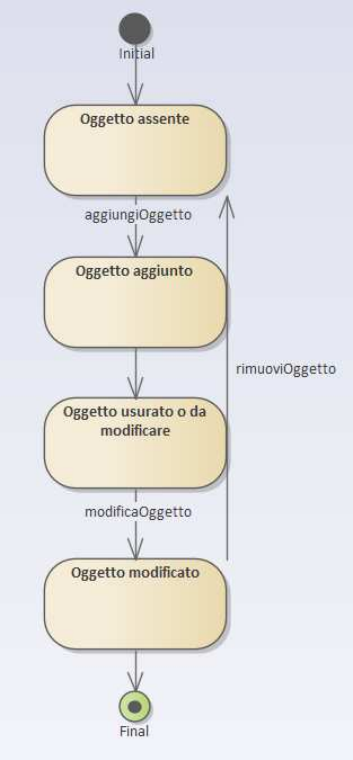

Figura 4.14: Diagramma delle macchine a stati di oggetto

4.1.2 Discussione delle scelte architetturali

Nel costruire il programma, come accennato in precedenza, si è deciso di separare le varie componenti del software, unendo quelle di una stessa categoria o raggruppando funzionalità comuni, in diverse directory, così da avere un codice strutturalmente più pulito, chiaro e facilmente manutenibile.

Si è, inoltre, cercato di costruire il programma adottando un pattern architetturale classico, definito *Model View Controller*, abbreviato in *MVC*. Questo pattern, come dice il nome stesso, divide il software in tre parti, dove, ognuna, ha un ruolo specifico. Le tre parti sono:

- *Model*: qui vengono raggruppate tutte quelle componenti che si occupano di definire i dati. In particolare, in questo caso, si è deciso di raggruppare insieme, sotto la directory *Class*, tutte quelle classi che modellano la realtà di questo sistema gestionale.
- *View*: comprende tutte quelle componenti che, invece, modellano l'interfaccia grafica del programma; nel gestionale, viene definito dalla directory *gui*.
- *Controller*: questa ultima parte è come se facesse da collante per le altre due; cioè, serve sia a far funzionare correttamente l'interfaccia con le funzioni implementate ma, soprattutto, si occupa di far cambiare i dati, ovunque essi siano memorizzati, in base alle scelte che l'utente effettua, utilizzando metodi implementati in essa. Nella nostra applicazione questa parte si rispecchia con la directory *Gestori*.

Bisogna notare, dunque, che, seppur divise in sezioni diverse, le tre aree non sono isolate, ma anzi collaborano per il corretto funzionamento del programma. Il pattern architetturale *MVC* comporta anche altri vantaggi dal punto di vista dell'efficienza, della chiarezza e della manutenibilità.

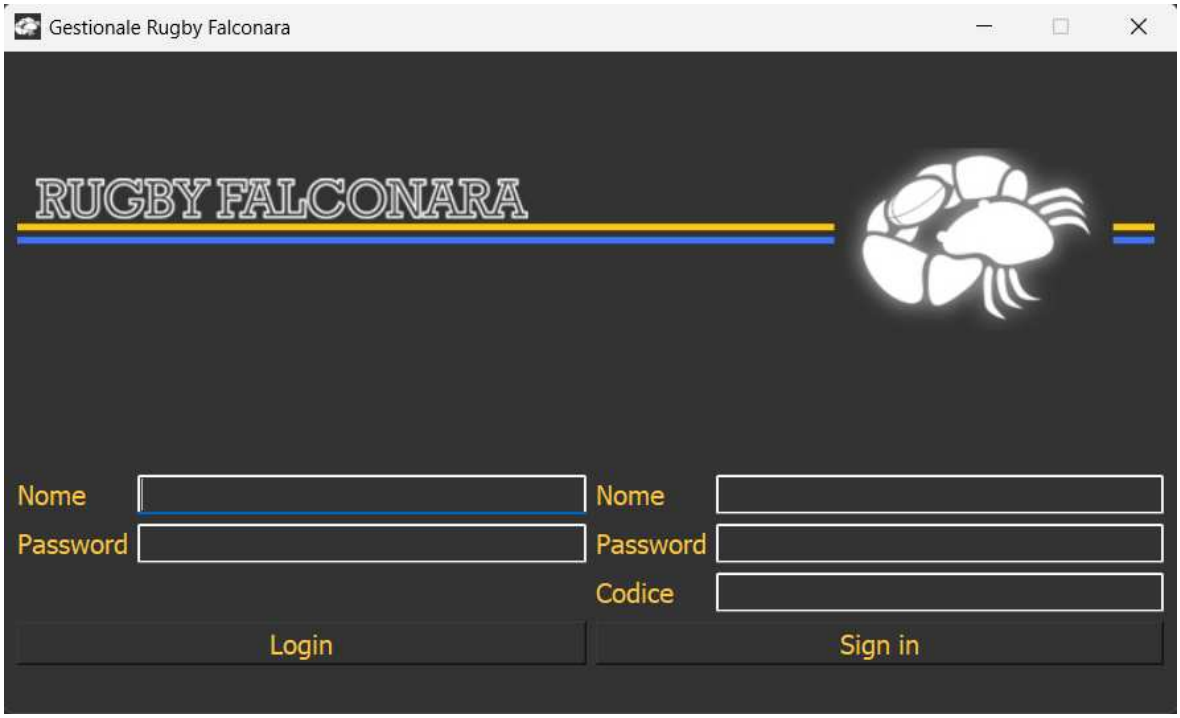

Figura 4.15: Login

Ci sono molti altri pattern che si sarebbero potuti utilizzare, come il *Factory method* o il pattern *Observer*; tuttavia si è ritenuto sufficiente l'utilizzo dell' *MVC*, visto che, di per sè, organizza e divide abbastanza bene il gestionale. Il codice, infatti, in questa maniera, risulta semplice e coerente con quanto richiesto dalla società di rugby, visto che quest'ultima, come anche affermato in precedenza, non avendo a disposizione troppe risorse per la gestione del programma stesso, necessitava di un gestionale che potesse essere facilmente comprensibile da, quasi, tutti e, soprattutto, non necessitasse di una manutenzione troppo complessa.

4.2 Interfaccia utente

In questa sezione, dopo aver analizzato le scelte architetturali prese, si descrive la composizione del *front-end*, ovvero tutto ciò del gestionale che l'utente finale visualizza e con cui può interagire.

L'interfaccia utente, seguendo il pattern architetturale descritto nella sezione precedente, verrà gestita da tutte quelle componenti che rientrano nella directory *gui*; bisognerà, poi, fare in modo che, ad ogni azione dell'utente, vi sia una conseguente risposta del sistema, sia positiva, che negativa. Si tenga, fin da subito, presente che il funzionamento, nel particolare, di ogni area descritta in seguito, verrà trattato in uno dei successivi capitoli della trattazione.

4.2.1 Descrizione dell'interfaccia utente e dei suoi componenti

Seguendo le specifiche ottenute nelle fasi inziali di progetto, si è cercato di rendere il programma il più semplice ed intuitivo possibile, in modo tale da permettere l'utilizzo senza avere problemi di comprensione delle funzionalità.

All'apertura del software viene subito mostrata una schermata che permette il login nel gestionale, mostrata nella Figura 4.15, utilizzabile attraverso opportune credenziali; ciò è stato fatto per dividere le diverse aree di competenza di ogni utilizzatore.

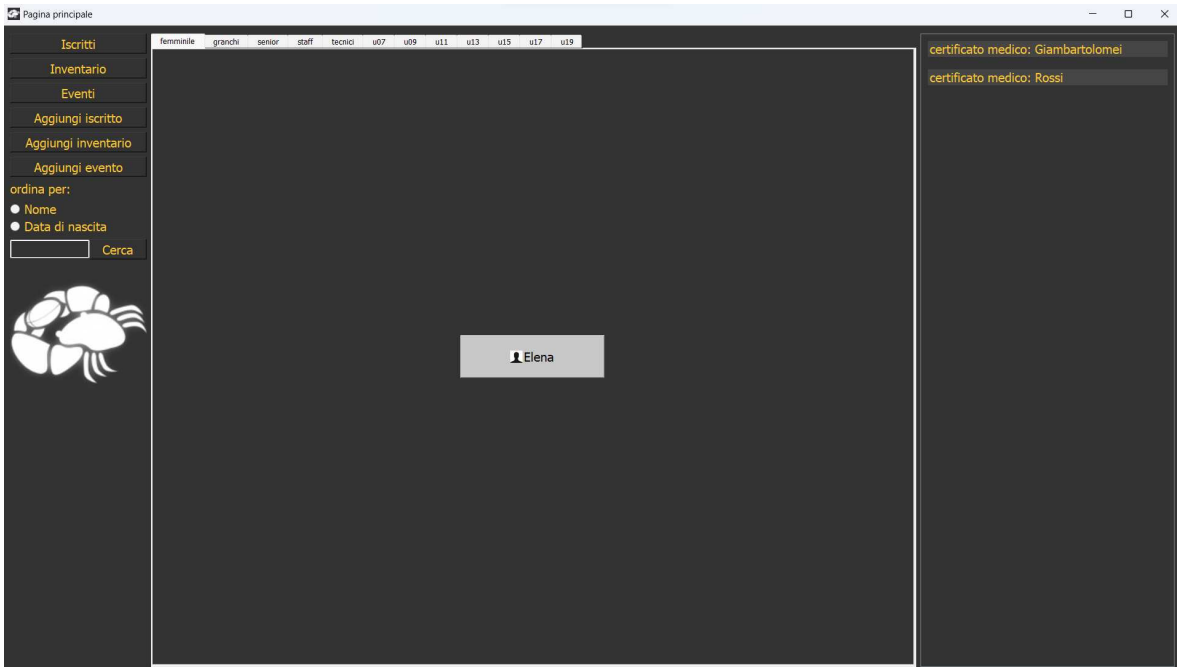

Figura 4.16: Home page

Una volta autenticato, l'utente si trova di fronte alla home page del gestionale, mostrata in Figura 4.16. Questa viene divisa in 3 aree: *iscritti*, *inventario* e *eventi*. Ognuna di queste aree viene, a sua volta, divisa in varie schede, una per ogni *categoria*, facilmente individuabili nella parte superirore della schermata. A completamento di questa sezione troviamo, poi, due bande laterali, una a sinistra e una a destra; nella prima troviamo una serie di pulsanti e una barra di ricerca, che consentono di spostarsi agilmente tra le varie aree del gestionale e sono il fulcro della navigazione del programma; nella seconda, invece, troviamo tutti gli avvisi di certificati scaduti relativi ai vari iscritti alla società.

Con queste scelte di design si riescono a rispettare diversi requisiti espressi dalla società, tra i quali la divisione delle macro aree in base alle categorie o, anche, la necessità di avere dei promemoria riguardanti le scadenze dei certificati medici, oltre che la semplicità di navigazione nel gestionale.

Come si vede sulla sinistra, possiamo aggiungere elementi in ogni area, a seconda di dove ci si trova. Una volta cliccato il relativo pulsante, comparirà nello schermo un form di inserimento che, chiaramente, si diversifica nei campi da compilare in base a che cosa si vuole inserire nel gestionale. Tutti i form, anche quelli utilizzati per la modifica e la visualizzazione di un elemento, sono stati creati con la medisima struttura per permettere agli utilizzatori di abituarsi al funzionamento del sistema senza dover ogni volta diversificare il loro comportamento a seconda di che cosa si vuole fare.

Nelle Figure 4.17, 4.18, 4.19 e 4.20, è possibile vedere i form di inserimento, rispettivamente, di *iscritto*, *oggetto*, *allenamento* e *partita*.

Si fa notare che non esistono due form diversi per *attrezzatura* e *merc*; questo perchè il primo non ha nulla di diverso dal secondo, ma, semplicemente, non contiene i valori *"link"* e *"prezzo"*; in particolare, queste due voci sono visibili nel form, ma poi, in fase di visualizzazione, non vengono considerati. Questa scelta è stata fatta per non appesantire troppo l'interfaccia e il funzionamento della sezione *inventario*, in quanto la parte relativa al merchandising non è così grande da necessitare un form a parte; dunque si è ritenuto opportuno unificare l'inserimento con gli oggetti *attrezzatura*.

Figura 4.17: Form di inserimento di un iscritto

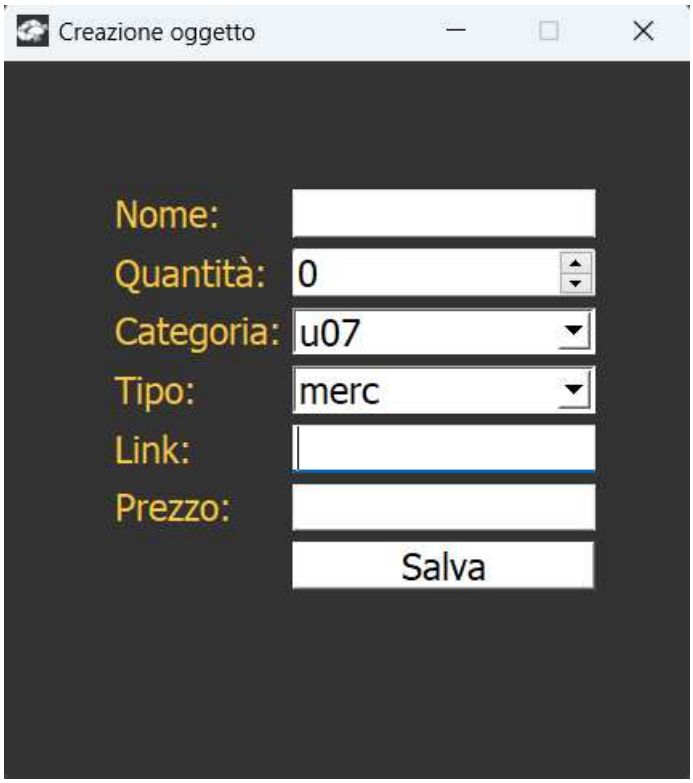

Figura 4.18: Form di inserimento di un oggetto nell'inventario.

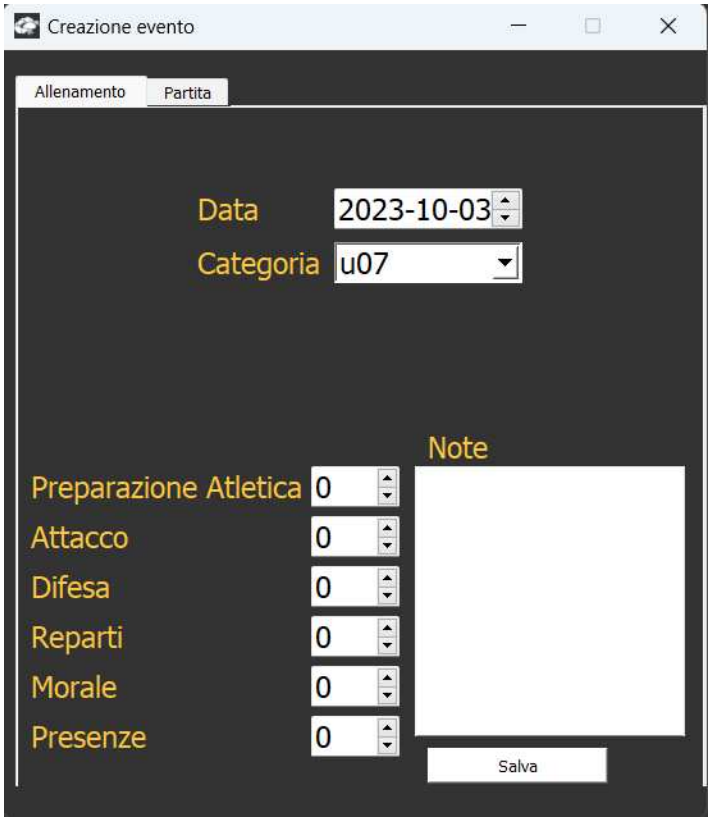

Figura 4.19: Form di inserimento di un allenamento.

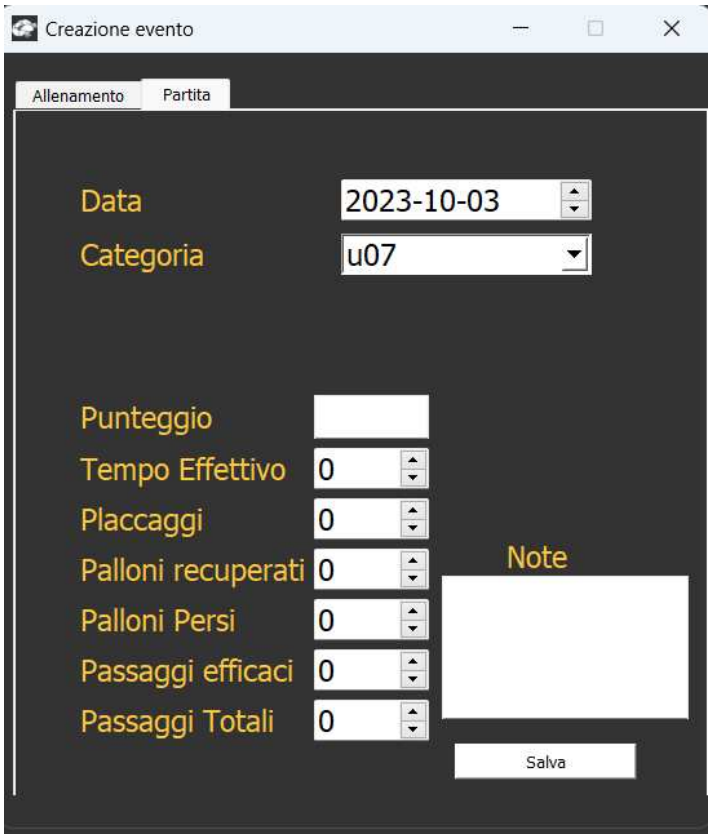

Figura 4.20: Form di inserimento di una partita.

4.2.2 Spiegazione delle decisioni di design prese per l'usabilità

In parte le scelte sono già state discusse in precedenza. In particolare, si è già parlato della scelta della divisione in tre macroaree e della questione riguardante il form di inserimento di un oggetto dell'*inventario*.

Inoltre, la scelta di avere le notifiche riguardanti i certificati medici sempre visibili è stata fatta in quanto, da come si legge nei requisiti del gestionale, questi documenti sono di fondamentale importanza per il corretto svolgimento delle partite e degli allenamenti da parte degli atleti.

Altra scelta di design, che è stata fatta per l'usabilità del programma, è quella di utilizzare dei rettangoli, di grandi dimensioni, per ogni elemento del sistema. Si è deciso di utilizzare questo stile, piuttosto che una semplice lista, in quanto questo gestionale è pensato per un utilizzo prolungato, sopratutto da parte della segretaria della società di Falconara, dunque, visivamente, a distanza nel tempo, queste componenti di grandi dimensioni dovrebbero, insieme alle funzionalità di ordinamento e ricerca, facilitare la coretta e rapida selezione degli elementi sullo schermo.

Un'altra scelta di cui bisogna parlare è quella di dividere in un'unica categoria gli allenatori e lo staff. Potrebbe risultare più logico, in un primo momento, inserire nelle varie categorie di squadre i tecnici relativi; tuttavia, dovendo questo gestionale aiutare la società nella gestione degli iscritti, si è ritenuto opportuno effettuare questa divisione per dare la possibilità alla segretaria di avere una vista generale su tutti gli iscritti e, dunque, anche sui tecnici e sullo staff. Ad esempio, in questa maniera, la segretaria, o l'amministratore, avrà la possibilità di consultare subito le schede relative ai tecnici senza dover ogni volta controllare, di ogni categoria, iscritto per iscritto, quale è l'allenatore; questo considerando che, probabilmente, la segretaria non conoscerà, uno per uno, tutti gli allenatori o, in generale, lo staff e i lori ruoli all'interno della società.

Si fa notare, infine, che gli utenti di tipo *allenatore* possono accedere alla parte relativa agli *iscritti*, e, di conseguenza, possono anche controllare lo stato dei certificati degli atleti tramite le notifiche a sinistra nello schermo. Questa scelta, seppure all'inizio può sembrare un errore, in realtà è stata presa di comune accordo con il responsabile tecnico degli allenatori per stimolare tutti i tecnici della società ad aver cura dei propri atleti e a controllare se i loro certificati sono validi o meno.

In generale, l'interfaccia si presenta molto semplice e intuitiva proprio con lo scopo di facilitare l'usabilità da parte di ogni utilizzatore, non solo quelli del presente, ma anche quelli che in futuro faranno parte della società. Infatti, vista la facilità di utilizzo, non ci sarà bisogno di istruire i nuovi utenti all'utilizzo del software, almeno nelle sue parti principali, perchè l'utente sarà indirizzato e guidato dal programma stesso in tutto ciò che vorrà fare.

4.3 Funzionalità implementate e dettagli su come queste supportano le esigenze della società di rugby

Tutte le funzionalità implementate sono derivate dai requisiti richiesti dalla società stessa. Tutte e tre le macroaree del programma sono dotate delle funzionalità *CRUD*, ovvero *create* (in questo caso sarebbe più corretto parlare di inserimento piuttosto che di creazione), *read* (è possibile visualizzare i dati di ogni elemento del gestionale), *update* (i dati possono essere aggiornati), ed, infine, *remove* (è possibile rimuove gli elementi dal sistema).

Questa sopra elencata è la base del programma; poi sono state sviluppate diverse altre funzionalità che permettono, per prima cosa, di soddisfare completamente le richieste della società, ma ci sono anche parti implementate per permettere un utilizzo semplice e fluido. Ad esempio, nella home page, a sinistra, troviamo una serie di pulsanti che non solo permettono

la navigazione, ma consentono, in ogni macro area, di ordinare gli elementi secondo diversi parametri; inoltre, è presente una barra di ricerca. Ciò è stato fatto in virtù dei volumi di oggetti con cui la società sportiva dovrà lavorare; per avere un esempio, è sufficiente controllare le tabelle utilizzate, in un capitolo precedente, per l'analisi logica della base di dati. Aggiungendo queste possibilità di utilizzo diversificate, si permette all'utente finale di risparmiare tempo e di avere, nel complesso, un'utilizzo *user friendly*.

Oltre a ciò, un'altra funzionalità importante, ai fini delle esigenze della società, è quella relativa alle notifiche delle scadenze o delle assenze dei certificati medici; questi avvisi, se cliccati, aprono una finestra contenente delle informazioni riguardo l'atleta che presenta problemi con il certificato medico ed eventuali contatti, utili alla segretaria o all'allenatore, per informarlo. La finestra presenta solo due pulsanti, ovvero la classica croce rossa in alto a destra e un pulsante con scritto *OK*; solo al click di quest'ultimo la notifica verrà tolta dalla lista. Ciò è stato fatto per permettere, a chi di dovere, di gestire in maniera più agevole questa problematica, consentendo di eliminare i promemoria solo una volta che, effettivamente, si dà risposta di presa visione al sistema; chiaramente, se il certificato medico non viene aggiornato ad una nuova scadenza o non viene caricato, l'avviso continuerà a comparire ad ogni accesso al gestionale, anche se precedentemente tolto dalla lista.

Ultima, ma non meno importante, funzionalità che il gestionale progettato offre è quella di permettere, in questo caso, solo all'allenatore, o eventualemente all'utente designato come amministratore, di inserire la formazione dei giocatori di una certa partita. Questa funzione è stata implementata per permettere ai tecnici di avere un resoconto dei giocatori che sono stati schierati nelle gare precedenti, oltre che iniziare ad istruire i *team manager* alla compilazione delle liste gara pre-partita, comunicando con gli allenatori, funzionalità, anche questa, richiesta durante le interviste ad inizio progettazione. La possibilità di inserire una formazione per una partita, tuttavia, è stata implementata in modo tale che non sia obbligatoria, ma, sopratutto, non venga richiesta contestualmente all'inserimento della partita stessa. Questo è stato fatto perchè, per la categoria di bambini tra i 5 e i 7 anni, ad esempio, non è necessario tenere traccia di chi ha giocato, in quanto, dal punto di vista tecnico, non è fondamentale; dunque, si è ritenuto opportuno aggiungere un pulsante che appare nella finestra, relativa ad una certa partita, solo dopo il suo effettivo inserimento nel gestionale, così da non appesantire inutilmente l'interfaccia. Ogni tecnico, comunque, a prescindere dalla sua categoria di appartenenza, ha sempre la possibilità di inserire una formazione per un evento.

4.4 Robustezza e affidabilità del sistema

Quella appena fatta è stata una panoramica generale delle componenti, architetturali e funzionali, fondamentali del sistema gestionale. Vedremo,ora, come viene garantita l'efficienza del software. Per prima cosa, si fa notare che l'adozione di una base di dati dedicata, anche se in un server locale, permette di garantire un utilizzo perpetuo e sicuro anche in casi di mancanza di connessione Internet e in caso di mancanza di spazio di memoria sul calcolatore in uso. Questo comporta, anche, che le operazioni cosidette *CRUD* sono gestite dalla base di dati stessa, attraverso delle *query* personalizzate, che quindi non possono avere problemi.

Si potrebbe incappare in degli errori se gli argomenti passati alla *query* non sono esatti, oppure non sono nel numero giusto. Inoltre potrebbero verificarsi dei problemi se, prendendo i dati, appunto, dal database, questi non vengono gestiti correttamente per creare quelle strutture dati che poi vengono utilizzati dal gestionale per svolgere le sue operazioni. Fondamentalmente i problemi principali che si potrebbero incontrare, durante l'esecuzione del programma, sono questi. Nelle prossime sottosezioni vengono descritti i metodi con cui si sono risolti, prevenuti o, eventualmente, limitati questi problemi.

4.4.1 Gestione delle Eccezioni e degli Errori

Per gestire il problema principale, ovvero quello legato alla corretta connessione con la base di dati, si è deciso di utilizzare un semplice *try-catch* (in Python *try-except*), dove, sostanzialmente, si verifica se, già in fase di login, la connessione con il database è correttamente stabilita. In caso si verificasse l'eccezione denominata *DatabaseError*, viene mostrato nello schermo una finestra di errore per metterne a conoscenza l'utente. Così facendo si evitano tutti i casi derivanti da questo errore, come ad esempio, l'esecuzione di una qualsiasi operazione sulla base di dati senza avere una connessione con essa; ciò viene reso impossibile perchè, effettuando il controllo all'inizio, si esclude a priori la possibilità nell'esecuzione del software, considerando anche che, essendo il programma pensato per un uso su server locali, la possibilità che la connessione al server non sia presente o che si interrompa durante l'esecuzione è molto remota o, comunque, dovuta ad eventuali errori sporadici difficilmente gestibili e, anche, individuabili.

L'altra tipologia di errori, quella legata al passaggio di parametri alle *query*, viene gestita impedendo all'utente di inserire nei form degli input che non sono conformi a ciò che il campo di immissione stesso richede. Infatti, se, per esempio, si vuole inserire un numero di telefono, escludendo il prefisso, questo sarà composto di soli numeri, dunque il sistema forza l'utente ad inserire i dati in quel formato, oppure, ancora, se si inserisce una mail, questa avrà come formato *stringa@stringa.stringa*; se l'utente fornisce un input non valido, il sistema non permette l'inserimento del dato nella base di dati. In questo modo, quindi, sono state create delle regole di validazione dell'input specifiche per ogni tipo di dato che le richiedeva.

Ultimo errore gestito, che potrebbe passare in secondo piano, è quello legato all'inserimento della formazione. Dal punto di vista procedurale, l'inserimento di una formazione è complesso perchè bisogna appurarsi che i giocatori inseriti facciano effettivamente parte della squadra che ha giocato l'evento e, inoltre, bisogna controllare che ogni giocatore compaia una volta sola. Questa problematica è stata prevenuta, come nel caso precedente, non utilizzando un *try-catch*, anche se si sarebbe potuto creare un'eccezione specifica, ma semplicemente, strutturando le varie *view* e, di conseguenza, le relative funzioni in modo tale che non permettessero in alcun modo di incorrere in questo errore.

Dal punto di vista degli errori, dunque, possiamo dire che, nonostante il codice non presenti *try-catch*, fatta eccezione per la situazione sopra elencata, esso è strutturato in modo tale da guidare l'utilizzatore e non permettergli di compiere errori. Questa cosa è stata fatta sopratutto per aumentare ulteriormente l'usabilità del programma, creando meno schermate di errore, ma rendendo l'immissione di dati e l'esecuzione di certe operazioni più vincolate.

4.4.2 Metodi di test e verifica

Per verificare il corretto funzionamento delle operazioni sui dati si sono utilizzati degli *unit test*. Questi servono a compiere delle operazioni simulate; al loro termine, comunicano, attraverso una stringa, se le esecuzioni sono fallite o meno ed, eventualemente, il tipo di problema che si è verificato.

Nel caso di questo gestionale i test sono stati raggruppati in una *directory*, denominata *Test*; questi file contengono, dunque, dei controlli sulle operazioni, principalmente di tipo *CRUD*, svolte appoggiandosi sulla base di dati. Eseguendo i test tramite terminale si è visto come questi non riportano errori, come mostrato in Figura 4.21; dunque, si può concludere che le operazioni sono ben strutturate e che, di conseguenza, il *model* del software ricalca con precisone le tabelle del database.

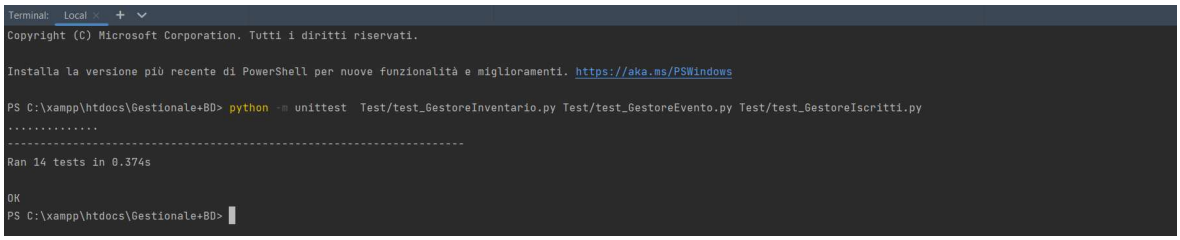

Figura 4.21: Foto del terminale dopo esecuzione dei 3 file di test

CAPITOLO 5

Implementazione e Manuale Utente

In questo capitolo verranno descritte le modalità con cui si è implementato il sistema gestionale. Si porrà l'attenzione sulle scelte tecniche fatte e sulle loro motivazioni, inoltre si descriveranno gli strumenti e le tecnologie usate per agevolare la fase di implementazione. Nell'ultima sezione, poi, verrà fornito un semplice manuale utente che avrà lo scopo di guidare l'utilizzatore sulle funzionalità del gestionale ed, evenutalmente, risolvere, o prevenire, alcuni problemi in cui si potrebbe incorrere

5.1 Implementazione

In questa prima sezione del capitolo si descriverà il processo e, in particolare, le scelte fatte in fase di implementazione del sistema gestionale. Si precisa che, con la parola *implementazione*, si intende l'insieme di tutte le fasi, procedure e azioni che hanno portato alla stesura del codice del programma. Dunque, bisognerà descrivere le parti che compongono il programma, dando ovviamente maggior attenzione a scelte particolari o, per qualche motivo, rilevanti.

Le scelte riguardanti l'organizzazione dei file sono già state descritte nei precedenti capitoli, pur essendo scelte che, in parte, sono state fatte in fase di implementazione, e, dunque, non verranno ritrattate. Ciò su cui ci soffermerà sono le librerie usate per gestire i dati provenienti dal database, e gli strumenti usati per produrre agilmente il codice. Questi argomenti verranno trattati nelle seguenti sottosezioni.

5.1.1 Librerie utilizzate per la gestione dei dati

Per l'unione fra codice Python e database SQL si è pensato di utilizzare una tecnologia *ORM (Object Relational Mapping)*. Questa tecnica permette di utlizzare i dati contenuti nelle tabelle della base di dati e le *query* relative mappandole in degli oggetti; nello specifico, si creano delle classi, con delle opportune parole chiave o strutture, che collegano il database al software, poi si creano altre classi per le *query*. Così facendo, ogni volta che si ha bisogno di una o dell'altra si richiamano questi oggetti senza dover utilizzare direttamente le *query* del database stesso.

Nel caso di questo gestionale, l'unica libreria utilizzata per la gestione dei dati è *SQLAlchemy*; questa, molto potente e flessibile, permette di utilizzare la tecnologia *ORM* per mappare le tabelle del database relazionale in delle classi, come, ad esempio, nella Figura 5.1, dove, ogni riga della tabella stessa corrisponde ad un oggetto istanza della classe relativa. Oltre a ciò, *SQLAlchemy* permette di scrivere, in Python, delle *query* SQL in modo diretto oppure, come è stato fatto nel caso della Figura 5.2, permette di utilizzare delle *query API*, che

```python
from sqlalchemy import Column, Integer, ForeignKey
from sqlalchemy.orm import relationship
from .Evento import Evento

class Allenamento(Evento):
    __tablename__ = 'allenamento'

    codice = Column(Integer, ForeignKey('evento.codice'), primary_key=True)
    preparazioneAtletica = Column(Integer)
    attacco = Column(Integer)
    difesa = Column(Integer)
    morale = Column(Integer)
    presenze = Column(Integer)
    def __init__(self, data, note, under, preparazioneAtletica, attacco, difesa, morale, presenze,codice):
        super().__init__(data, note, under,codice)
        self.preparazioneAtletica = preparazioneAtletica
        self.attacco = attacco
        self.difesa = difesa
        self.morale = morale
        self.presenze = presenze
```

Figura 5.1: Mappatura ORM della tabella SQL `Allenamento`

```python
def modificaAllenamento(self, allenamento):
    evento_da_modificare = session.query(Evento).filter_by(codice=allenamento['codice']).first()
    if evento_da_modificare != None:
        evento_da_modificare.data = allenamento['data']
        evento_da_modificare.note = allenamento['note']
        evento_da_modificare.under = allenamento['under']
    allenamento_da_modificare = session.query(Allenamento).filter_by(codice=allenamento['codice']).first()
    if allenamento_da_modificare != None:
        allenamento_da_modificare.preparazioneAtletica = allenamento['preparazioneAtletica']
        allenamento_da_modificare.presenze = allenamento['presenze']
        allenamento_da_modificare.attacco = allenamento['attacco']
        allenamento_da_modificare.difesa = allenamento['difesa']
        allenamento_da_modificare.morale = allenamento['morale']
    session.commit()
```

Figura 5.2: Query API di SQLAlchemy che compaiono nel metodo `modificaAllenamento`

consentono una scrittura, di queste richieste semplice, ma comunque efficiente e completa. Non si sono utilizzate delle *query* dirette perchè avrebbero inutilmente appesantito il codice, risultando di difficile comprensione e non facilmente manutenibili.

5.1.2 Strumenti e tecnologie

Uno degli strumenti utilizzati per la creazione del software è *QtDesigner*. Questo programma è un editor grafico per la creazione di *GUI*, provenienti da file con estensione `.ui`. Sostanzialmente, esso mette a disposizione tutte le strutture della libreria `Qt`, permettendone l'utilizzo attraverso un editor che mostra l'interfaccia in fase di creazione. Questo ha permesso di rendersi conto in tempo reale se il design di ogni parte dell'applicazione fosse adeguato o meno per la società di rugby. Ovviamente l'utilizzo di questo software ha anche permesso di risparmiare molto tempo nello scrivere diverse righe di codice che, altrimenti, sarebbero dovute scrivere manualmente.

Un altro software importante utilizzato, sopratutto per la verifica della corretta esecuzione del gestionale, è *XAMPP*. Questo programma permette di creare un server locale e di basare su di esso la creazione di un database relazionale di tipo SQL. Così facendo si è potuto controllare che il gestionale funzionasse correttamente e, allo stesso modo, che il database fosse correttamente strutturato e che rispondesse correttamente alle operazioni che il programma richiedeva di eseguire.

Un ulteriore software usato è *Enterprise Architect*. Questo è un programma che viene utilizzato in ambito lavorativo e permette di creare grafici e diagrammi UML di ogni tipo. Nel caso corrente, ha permesso di creare tutti i diagrammi utilizzati in fase di progettazione del programma. *Enterprise Architect* è un software molto potente che può essere utilizzato

per strutturare ed eseguire la progettazione di un software nel suo complesso senza risultare riduttivo o scarno. In questo caso è stato utilizzato, anche se si sarebbero potuti usare programmi molto più semplici, per avere a disposizione un ambiente maggiormente professionale e che permettesse di strutturare il gestionale nella sua interezza senza dover usufruire di programmi terzi.

Non ci sono altri strumenti rilevanti che sono stati utilizzati nella creazione di questo programma; tuttavia, è giusto citare l'IDE utilizzato per la stesura del programma, ovvero *PyCharm*. Questo è, probabilmente, l'IDE più utilizzato per la scrittura di programmi Python, essendo dotato dei più semplici, ma, allo stesso tempo, efficienti, *tool* per la programmazione in questo linguaggio. Per questo progetto, come di norma accade, è stato molto utilizzato il debugger *built-in* per evitare che ci fossero errori nascosti in fase di esecuzione o eventuali *bug*, in altri casi difficilmente rilevabili.

5.2 Manuale Utente

L'intento di questa sezione è quello di fornire un manuale d'uso per gli utilizzatori del sistema in modo tale da guidarli o, eventualmente, risolvere eventuali problemi da loro incontrati durante l'esecuzione.

Si pone l'attenzione sull'utilizzo dei termini, volutamente in maiuscolo *ALLENATORE e SEGRETARIA*. Quando vicino ad una frase sarà riportata una di queste due espressioni significa che il funzionamento è limitato a quella specifica figura. Si ricorda, inoltre, che questo discorso non vale per l'utente *amministratore*, in quanto egli ha accesso ad ogni funzionalità del gestionale.

5.2.1 Registrazione e Accesso

All'avvio dell'applicazione viene richiesto l'accesso con delle credenziali da parte dell'utente. Questo può registrarsi, sempre nella stessa schermata del login, inserendo un nome utente, una password a piacere e un codice che potrà essere: *amministratore, segretaria, allenatore* a seconda della persona che lo utilizza. Una volta registrato l'utente può effettuare l'accesso all'applicazione inserendo i suoi dati nel lato sinistro della schermata di accesso.

Se l'utente commette un errore in una delle due fasi di esecuzione, il sistema lo avvertirà con un messaggio di errore sullo schermo. Un errore comparirà, come precedentemente detto, anche se la connessione con il database del programma non è stata stabilita correttamente. Si rimanda alla Figura 4.15 per visualizzare la schermata di accesso.

5.2.2 Navigazione nell'Applicazione

La navigazione è semplice ed intuitiva. Ci sono, sul lato sinistro della schermata, dei pulsanti che portano alle seguenti aree: Iscritti, Inventario(SEGRETARIA), Eventi(ALLENATORE). Una volta all'interno di ogni area sarà possibile visualizzare gli oggetti, gli eventi o gli iscritti, a seconda del caso, relativi ad ogni categoria, muovendosi fra le schede in alto nello schermo. Ogni scheda rappresenta una specifica categoria della società, dunque sia squadre di atleti, sia staff che tecnici. Sarà possibile, per ogni categoria, ordinare gli elementi mediante due parametri differenti e, oltretutto, se si cerca un elemento in particolare, sarà possibile farlo attraverso l'apposita barra di ricerca. Gli elementi dell'inventario saranno ricercabili tramite nome o quantità degli oggetti, gli iscritti si potranno cercare tramite nome, cognome o codice fiscale, mentre gli eventi saranno ricercabili tramite data o una parola qualsiasi nelle note. Anche qui, si rimanda alla Figura 4.16 per vedere la schermata principale; nel caso della

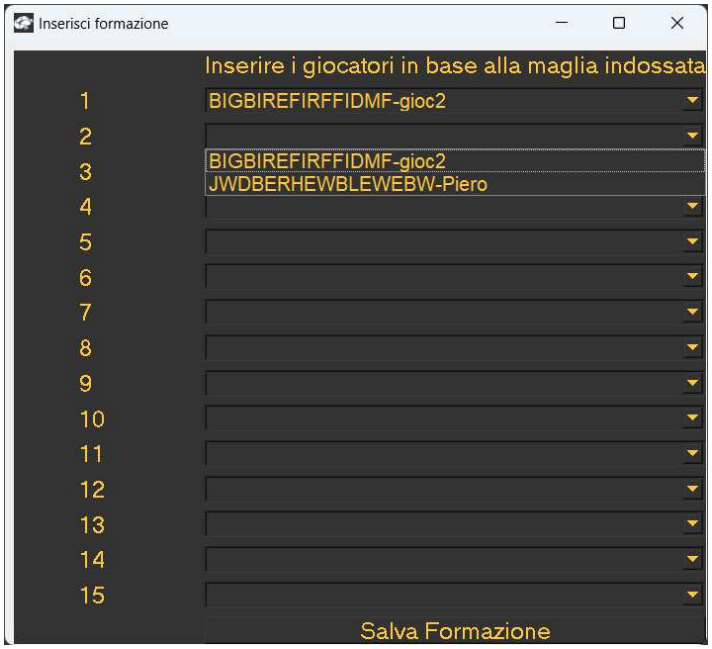

Figura 5.3: Finestra di inserimento di una formazione

figura, l'utente che ha effettuato l'accesso è di tipo amministratore, lo si nota dal fatto che può navigare in tutte e tre le aree del gestionale.

5.2.3 Funzionalità principali e gestione dati

Oltre alle funzionalità già descritte in precedenza, le cui modalità di fruizione sono già state esplicate, si possono modificare o eliminare elementi tramite il click del tasto destro su uno di essi; ovviamente, cliccandoli normalmente si potranno visualizzare le loro informazioni. Si possono, poi, inserire oggetti tramite gli appositi pulsanti posizionati al lato sinistro dello schermo; al click di uno di essi apparirà una form, come quella mostrata nella Figura 4.20, specifica per ogni caso, in cui andranno immessi dei dati per poi inserire l'elemento nel gestionale e poterlo visualizzare. Come descritto in uno dei capitoli precedenti, il sistema guiderà l'utente al corretto inserimento dei dati, avvisandolo con dei messaggi sullo schermo se compierà degli errori nell'inserimento o nella modifica.

Sarà, poi, possibile visualizzare le notifiche riguardanti i certificati scaduti degli atleti e sarà possibile inserire delle formazioni negli eventi di tipo partita (ALLENATORE); per fare questa cosa sarà sufficiente cliccare la partita di riferimento e poi l'apposito pulsante contenuto all'interno della schermata visualizzata. Non sarà necesserio inserire una formazione completa, ma essa dovrà essere composta da atleti differenti; anche qui il sistema gestirà l'inserimento mostrando all'utilizzatore solo giocatori che effettivamente possono essere inseriti. Una finestra di inserimento di una formazione si presenta come mostrato in Figura 5.3. Si ricorda che vanno inseriti tutti atleti diversi; in caso contrario il sistema guiderà alla correzione, notificando che sono stati inseriti degli atleti più volte.

5.2.4 Risoluzione problemi comuni

Tra i problemi che si potrebbero verificare, ovviamente, c'è l'inserimento del codice in fase di login. Il sistema è *case sensitive*, cioè i codici vanno inseriti necessariamente tutti in minuscolo, altrimenti il sistema li considererà come errati.

Un'altra problematica da descrivere è il mancato aggiornamento delle notifiche.Infatti, leLe notifiche, riguardanti le scadenze dei certificati, si aggiorneranno solo quando il sistema viene riavviato; si presume che vengano inseriti solo atleti aventi i certificati validi o che, comunque, si sia a conoscenza del fatto che il certificato medico sia assente o non valido.

Bisogna prestare attenzione, poi, alla barra di ricerca; questa compie ricerche esaustive solo se l'input di ricerca è fatto in maniera corretta; ciò significa che non si possono cercare, ad esempio, un allenamento scrivendo solo il mese o l'anno, ma è necessario scrivere tutta la data per intero. Stessa cosa vale per tutte le altre ricerca che possono essere effettuate. Importante ricordare che le date sono nel formato *yyyy-mm-dd*.

Si pone l'attenzione anche al fatto che, durante la ricerca, si potrebbero avere risultati diversi, anche di diverse aree. Ciò è stato fatto di proposito proprio per permettere una ricerca incrociata su tutte le tre aree. Ad esempio, se ricercassimo la parola *Mario* e ci fosse un iscritto con quel nome e un evento con scritta una nota come *Mario si è fatto male*, la ricerca produrrà come risultato entrambi gli elementi.

Ultima problematica, che potrebbe verificarsi frequentemente, è quella relativa al salvataggio degli inserimenti e delle modifiche degli elementi. Ogni volta che si compie una di queste due azioni, bisogna verificare di aver premuto il tasto *salva*. Se, per errore, si preme il tasto canonico per la chiusura della finestra, ovvero la croce rossa in alto a destra, le operazioni non verranno salvate e non sarà possibile in alcun modo recuperare i dati inseriti.

CAPITOLO 6

Discussione in merito al lavoro svolto

In questo capitolo viene svolta un'indagine riguardo la qualità del gestionale sviluppato. Verrà, nella prima parte, svolta un'analisi che metterà in risalto pregi e difetti del software in modo da rendersi completamente conto di ciò che si è creato; inoltre, nella seconda parte, verranno fatti dei confronti con tre sistemi gestionali, già esistenti nel mercato, così da avere un metro di giudizio pratico e immediato per il software

6.1 Analisi SWOT

In questa sezione si commeteranno e discuteranno, in maniera critica, il lavoro svolto e i risultati raggiunti; per farlo si utilizzerà una tecnica chiamata *Analisi SWOT*. Questa tipologia di analisi è uno strumento di pianificazione strategica utilizzato per valutare i punti di forza, le debolezze, le opportunità e le minacce di un progetto, di un'impresa o di qualsiasi altra situazione in cui un'organizzazione o un individuo debba prendere una decisione per il raggiungimento di un obiettivo.

L'analisi SWOT ha origine da una ricerca condotta dall'economista statunitense Albert Humphrey negli anni '60. Humphrey era un consulente aziendale che lavorava presso lo Stanford Research Institute, e condusse la ricerca per comprendere i motivi per cui la pianificazione aziendale falliva. Successivamente, l'analisi SWOT è stata resa pubblica per la prima volta nel 1972, e da allora è diventata uno strumento ampiamente utilizzato in una varietà di contesti.

A prescindere dal motivo per cui si sta utilizzando questa strategia, le parti che vengono analizzate sono: i punti di forza (*strengths*), ovvero le caratteristiche interne di un'organizzazione o di un progetto che consentono di avere un vantaggio competitivo; le debolezze (*weaknesses*), ovvero le caratteristiche interne di un'organizzazione o di un progetto che possono ostacolarne il successo; le opportunità (*opportunities*), ovvero fattori esterni che possono favorire il successo di un'organizzazione o di un progetto; le minacce (*threats*), ovvero fattori esterni che possono causare problemi nel corretto sviluppo del progetto.

6.1.1 Punti di forza

Un sicuro punto di forza di questo sistema gestionale è l'essere stato creato su misura per la società sportiva che lo richiede. Ciò permette, ad esso, di prevedere tutte le funzionalità necessarie allo sviluppo e all'organizzazione della società di rugby. Anche se nel web si possono trovare diversi altri tipi di gestionali, nessuno di questi sarà mai perfettamente

adattabile, come quello creato, sulle esigenze della società, essendo essa una realtà piccola e, quindi, rispetto a grandi club sportivi, non necessita di funzioni troppo complesse.

Un altro punto di forza è la versatilità del gestionale; infatti, questo può essere usato per diversi scopi e da diverse tipologie di persone, come, ad esempio, allenatori o persone di segreteria; ciò aumenta la longevità del programma, che in questo modo, potrà essere utilizzato per più tempo, ma permetterà, anche, di compiere azioni diverse, tutte però sullo stesso software. Questo particolare potrebbe risultare inutile; in realtà, però, è una peculiarità molto importante in quanto permette agli utilizzatori sia di non perder tempo sia di non doversi abituare ad un gestionale diverso per ogni area da controllare o da gestire. Ad esempio, se un allenatore deve inserire la formazione di una partita e, contemporaneamente, vuole controllare se dispone di tutto il materiale tecnico necessario per disputare l'evento, può farlo contemporanemente senza il minimo sforzo.

Ultimo, ma forse più grande, punto di forza del gestionale è la sua semplicità nell'utilizzo. Per capire l'importanza di questo ultimo punto basta considerare che, in media, in ambito lavorativo, un sistema gestionale viene utilizzato dalle 2 alle 3 ore. Avendo un'interfaccia intuitiva e molto interattiva, il gestionale permette un utilizzo agevole anche per lunghi periodi, consentendo all'utente di non affaticarsi mentalmente troppo durante il lavoro. Inoltre, parlando di questo argomento, bisogna anche ricordare come il gestionale guidi l'utente durante l'utilizzo delle funzionalità del database, escludendo ogni possibile errore, o ambiguità, che potrebbero generarsi da un utilizzo improprio di queste funzioni. Questi attributi, uniti fra loro, permettono a questo programma di essere uno strumento efficiente e completo, ma di semplice utilizzo.

6.1.2 Debolezze

Un problema, presente nel gestionale, è le modalità di accesso. In particolare, il modo con cui viene effettuata la registrazione non rispetta gli standard canonici utilizzati in altre applicazioni; non viene prestata molta attenzione alla sicurezza e all'unicità di ogni singolo account, ciò si può notare dal fatto che non viene richiesta, all'atto della registrazione, una conferma della password e non viene usato nessun altro metodo di verfica. Inoltre, bisogna anche considerare il fatto che non è presente un metodo per recuperare le credenziali, ed eventualemente cambiarle, nel caso in cui l'utente se le fosse dimenticate. Queste considerazioni, seppur, in parte, erano state calcolate al momento della progettazione e dell'implementazione, rendono il gestionale facilmente accessibile anche da persone non autorizzate. Questa debolezza è, però, mitigata dal fatto che, comunque, questo gestionale è pensato per un uso interno alla società e, in particolare, da un unico computer, visto che utilizza un server locale; dunque, si presume che le persone che vi abbiano accesso siano controllate e veicolate dalla società stessa.

Seguendo il discorso appena fatto, si nota che un'altra debolezza del gestionale ideato è la scarsa personalizzazione per ogni utente. Infatti, oltre alla normale disitinzione dei ruoli degli utilizzatori, non c'è nessun altro elemento che permette di diversificare, e quindi rendere unica, l'esperienza di ogni utente. Questo aspetto, anche se tende ad aumentare l'efficacia del software, facendo in modo che ognuno abbia lo stesso software a disposizione e permettendone, quindi, un uso maggiormente collettivo, potrebbe, nel lungo periodo, rendere il programma monotono facendo si, di conseguenza, che esso perda di interesse agli occhi degli utenti.

Passando ad aspetti più tecnici, un'altra debolezza che il gestionale ha è quella relativa ai documenti di cui esso tiene traccia. Il gestionale si limita solo a controllare o meno se un certo documento esiste; nel caso del certificato medico permette anche di inserire la scadenza; tuttavia non dà la possibilità di caricare nel database il documento stesso; questo potrebbe risultare un ostacolo per la società di rugby nel tenere sotto controllo i documenti

di ciascun atleta, vincolando la società a tenere una copia cartacea di ogni documento, di cui vuole avere un riscontro, che ha segnato come presente nel sistema. Va, poi, tenuto in considerazione che il gestionale non considera, per scelta, volendo evitare di appesantire il software con funzionalità non espressamente richieste dalle persone intervistate per i requisiti, la situazione legata agli atleti minorenni. Non viene data la possibilità di inserire e conservare le generalità dei genitori o tutori che iscrivono gli atleti minorenni, costringendo, anche qui, la società a dover farsi carico di conservare queste informazioni materialemente.

Altra debolezza è legata all'utilizzo della libreria utilizzata dell'interfaccia, ovvero *Qt*. Seppur questa è assolutamente sufficiente a rendere contemporanemente intuitivo e visivamente gradevole il gestionale, risulta una libreria ormai superata. Sarà, sicuramente, un vincolo per futuri sviluppi, visto che, se si volesse modificare l'interfaccia, probabilmente sarebbe necessario ripartire da zero utilizzando librerie più evulute e moderne.

6.1.3 Opportunità

Essendo la società di Falconara molto attiva sul territorio per favorire la crescita dei numeri nelle varie categorie, sicuramente, la mole di atleti, che in futuro si iscriverà al club, aumenterà, favorendo, di conseguenza, l'utilizzo del gestionale per organizzare tutti i vari profili. Inoltre, aumentando il numero di giocatori, aumenteranno anche i certificati da tenere sotto controllo, rendendo, così, fondamentale uno strumento di notificazione delle scadenze di essi, come quello implementato nel software.

Bisogna, anche, considerare che, probabilmente, in futuro, aumentando i numeri, ci sarà una maggior richiesta nella merce messa in vendita nel settore del merchandising; dunque, avere la possibilità di tenere traccia degli oggetti ed, eventualmente, avere un modo per rintracciare i fornitori rapidamente risulterà, sicuramente, molto utile.

C'è poi da fare un discorso più ampio; essendo il rugby, almeno in Italia, uno sport poco praticato, ma che sta prendendo campo negli ultimi anni, non ha a dispozione molti strumenti software dedicati per la gestione e l'analisi delle partite o degli allenamenti. Sicuramente dei top club in Italia avranno a disposizione programmi avanzati per svolgere queste operazioni; tuttavia i costi di questi software saranno difficilmente sostenibili da realtà come quelle di Falconara; dunque, avere a disposizione un gestionale dedicato, creato appositamente per il rugby, sarà sicuramente una risorsa importante da sfruttare per il club falconarese, e, in futuro, potrebbe rappresentare anche una valida soluzione, con le opportune modifiche, anche per le altre società rugbistiche marchigiane e non.

6.1.4 Minacce

Il pericolo maggiore a cui è sottoposto un gestionale in una socetà, o in generale in un'azienda, di piccole dimensioni è che, spesso, vista la mancanza di un'organizzazione completa ed efficiente, si tende a compiere certe operazioni in maniera superficiale, non individuando con certezza chi deve fare cosa, e il più velocemente possibile, visto che, non di rado, si tende sempre a fare le cose all'ultimo momento possibile, comportamento ormai definito come *sindrome dello studente*. L'utilizzo di un gestionale può essere visto, sicuramente, come un modo per cambiare questa metodologia di lavoro; tuttavia può anche avere l'effetto opposto; se non si entra subito in confidenza con il software e non se ne comprende a pieno l'utilità, il suo utilizzo può passare in secondo piano, relegandolo ad un'operazione di tipo formale che, però, non è considerata di grande utilità. Questo spiega anche il motivo per cui si è data tanta attenzione alla semplicità e all'efficacia durante la progettazione e l'implementazione del software.

Va, poi, tenuto in considerazione che, magari, non tutte le parti del gestionale potrebbero essere utilizzate nella stessa misura; ciò perchè non è detto che tutti coloro che dovranno

Figura 6.1: Home page del gestionale Rugby Manager 2.0

usare il software lo comprenderanno a pieno in tutte le sue parti e potrebbero optare per altri tipi di gestione. Questo potrebbe accadere sempre perchè la socetà sportiva in questione non è così grande da avere necessariamente una metodologia di lavoro unica e ben definita, ma è probabile che, a volte, vengano fatte delle scelte basandosi su ciò che è più utile nell'immediato piuttosto che su ciò che potrebbe portare giovamento anche in futuro. L'adozione di un software simile, in realtà, dovrebbe aiutare a cambiare questa situazione; tuttavia non è detto che ci riesca, almeno nell'immediato; ciò farebbe sì che il software non verrebbe utilizzato a fondo fin da subito.

6.2 Sistemi correlati

L'analisi del software appena fatta, è molto importante per giudicare, con occhio critico, ciò che si è progettato. Un metodo molto concreto che può essere utilizzato per avere un giudizio sul gestionale è quello di compararlo con altri programmi simili già esistenti. Dunque ora si procederà descrivendo alcuni gestionali sportivi analizzandone analogie e differenze con quello progettato; va fatta, però, una premessa, ovvero che difficilmente sul web si trovano gestionali creati appositamente per il rugby, ed è anche uno dei motivi per cui è stato necessario crearne uno da zero; dunque si procederà esaminando software gestionali sportivi che saranno, più o meno, adattabili al rugby.

6.2.1 Rugby Manager 2.0

Questo è un gestionale prodotto dal responsabile sviluppo club della società *rugby Perugia Junior*. Questo programma è già stato citato, in uno dei primi capitoli di questa trattazione, in quanto è uno dei prinicipali *competitor* del software sviluppato, essendo uno dei pochi gestionali disponibili sul web esclusivamente dedicati al rugby. Il programma della socetà umbra è utilizzabile da chiunque, previo un abbonamento annuale, ed è per questo che lo si confronterà con quello appena implementato. Nelle Figure 6.1 e 6.2 si possono notare alcune schermate prese dal sito ufficiale di *Rugby Manager*.

Analogie

Entrambi i gestionali si dividono in macro aree e prevedono funzionalità diversificate a seconda di chi utilizza il programma. Anche in *Rugby Manager* è possibile gestire sia i documenti sia i dati anagrafici degli iscritti, organizzati per categorie di appartenenza. Si possono gestire ed analizzare gli eventi, sia partite che allenamenti, dando, per ognuno, dei punteggi specifici in base alle prestazioni delle squadre che vi hanno preso parte. Per ogni partita, anche qui, è possibile inserire la formazione scegliendo opportunamente tra gli atleti. Inoltre, è presente, spostandosi in una delle aree dedicate alle squadre, una sorta di promemoria dei certificati medici scaduti. Anche *Rugby Manager* rende disponibili, o limita, certe funzionalità in base all'utente che effettua l'accesso.

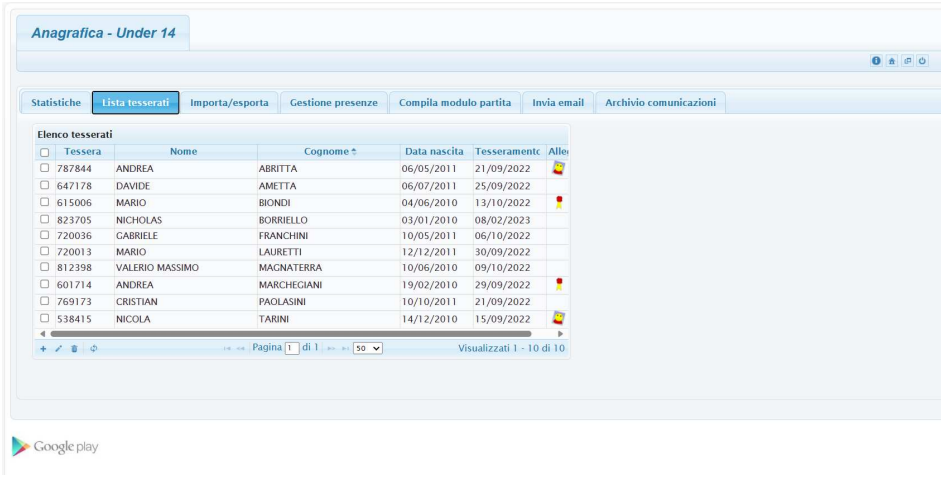

Figura 6.2: Schermata di Rugby Manager 2.0 dedicata agli iscritti della categoria under 14

Differenze

Seppure presente anche nel gestionale descritto in questa tesi, l'area di *Rugby Manager 2.0* dedicata alle statistiche degli eventi sportivi è molto più strutturata; in particolare, in essa è possibile dare un punteggio, da 1 a 5, a diverse categorie molto specifiche e, utilizzando i dati inseriti dall'utente, il gestionale crea in automatico degli diagrammi a torta o degli istogrammi che descrivono le prestazioni in vari intervalli di tempo.

In *Rugby Manager* è possibile anche tenere traccia delle presenze di ogni atleta; anche qui, poi, il programma, in automatico, può costruire delle tabelle che descrivono l'andamento delle presenze generali agli eventi del club.

Nella sezione *gestione visite mediche* è presente una sorta di calendario dove, per ogni giorno, vengono indicati eventuali scadenze di certifcati medici; ovviamente, le date vengono prese dai dati che devono essere opportunamente inseriti dall'utente. In merito agli inserimenti, questo gestionale permette di importare o esportare da file *Excel* tabelle contenenti gli iscritti, aumentando, così, la portabilità del sistema. Tuttavia, non è presente un sistema di notifiche automatico che ricorda le scadenze dei certificati.

Non sono presenti funzionalità riguardanti l'attrezzatura della società o il merchandising, nè è presente la possibilità di effetturare ricerche testuali.

Infine, va notato che *Rugby Manager 2.0* è una *web app* e, dunque, non è disponibile senza connessione ad Internet.

6.2.2 Gesosport

Questo è un altro sistema gestionale abbastanza utilizzato dalle società di rugby, e non solo; questo perchè il programma è fatto in modo tale da soddisfare le esigenze di più tipi di sport. Il gestionale in questione è di sicura efficienza viste anche le società che lo utilizzano, come, ad esempio, il *Firenze rugby 1931* o *il Gispi rugby Prato*.

In particolare, il sistema presenta diverse funzionalità divise nelle seguenti aree: area anagrafica, area logistica, area economica, area tecnica e area statisitca. Inoltre, presenta un'applicazione mobile, non potente come lo strumento software, ma che, comunque, permette di usfruire di un set base di funzioni.

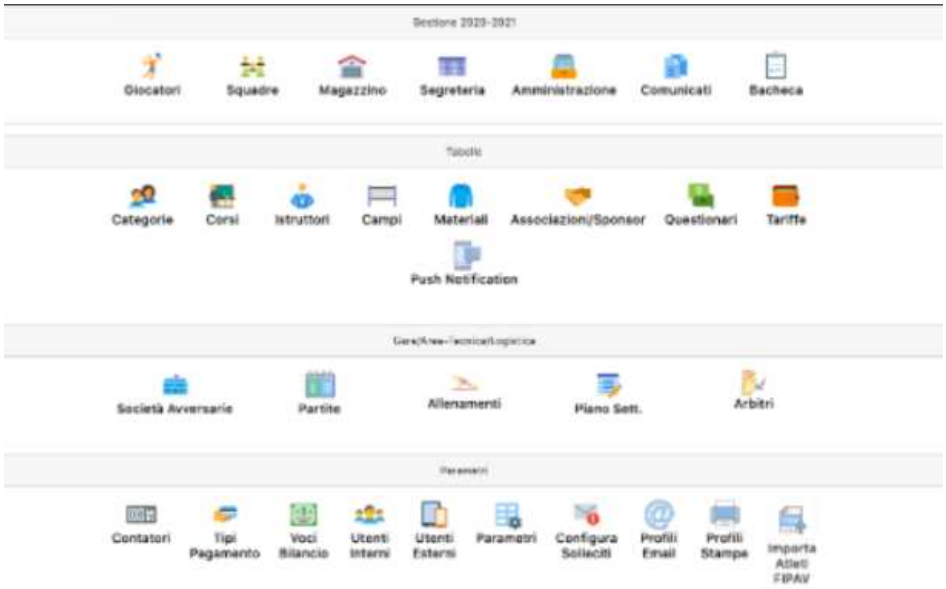

Figura 6.3: Home page di Gesosport

Analogie

Il software è, anch'esso, diviso in aree e, in particolare, implementa sia l'area anagrafica sia quella tecnica che quella logistica; queste macro aree ricalcano alla perfezione le sezioni del gestionale descritto in questa tesi; si noti che la parte logistica corrisponde a quella dell'inventario. Esso presenta un sistema di avvisi per le scadenze dei certificati medici e permette di controllare le situazioni legate ad ogni singolo atleta. L'area logistica tiene traccia delle attrezzature a disposizione di ogni squadra e permette di registrare gli eventi, quali allenamenti e partite. Nell'area tecnica, tra le altre, cose è possibile inserire le formazioni delle partite.

Da notare, infine, come si vede dalla Figura 6.3, la semplicità dela schermata principale, che divide le varie funzionalità in più sezioni, risultando, così, molto intuitiva.

Differenze

Gesosport è, sicuramente, molto completo dal punto di vista angrafico, dato che, comunque, permette il contatto diretto con gli atleti che non hanno saldato quote o hanno il certificato scaduto; anche la sezione dedicata agli eventi è molto funzionale, visto che permette di creare delle vere e proprie *liste gara*, ovvero dei fogli, utili a persone come i *team manager*, in cui vengono racchiuse tutte le informazioni principali sulla partita, sulla propria squadra e sulla squadra avversaria.

Non sono state citate tutte le altre aree che non trovano riscontro con il gestionale della società di Falconara, ovvero quelle relative alla parte economica, in cui si controllano le entrate e le uscite per poi poter redigere dei bilanci in tempo reale, e quella statistica, in cui si possono ottenere report in base al numero di iscritti, confrontandolo con quello degli anni precedenti, e statistiche sui rendimenti dei singoli giocatori.

L'app dedicata, inoltre, presenta varie sezioni, diverse dalla parte software, che possono essere utilizzate anche dagli iscritti stessi; ad esempio viene data loro la possibilità di segnare le presenze agli allenamenti, oppure di compiere i pagamenti delle quote annuali direttamente in app, o, ancora, di leggere avvisi e news della società.

Tuttavia, seppure *Gesosport* includa tutte queste funzionalità, non sembra esserci una parte dedicata alle statistiche degli allenamenti e delle partite. Non viene nemmeno menzionata la

possibilità di diversificare le funzioni in base agli utilizzatori, nè viene dedicata una parte al merchandising della società.

6.2.3 Sportivi in cloud

Questo è uno dei sistemi gestionali più famosi in Italia; basti pensare che è partner della nazionale italiana di calcio e, in ambito rugby, è utilizzato dal *Petrarca rugby*, una delle squadre italiane più importanti.

Come gli altri gestionali visti, anche questo si divide in due aree di competenza, una anagrafica e l'altra finanziaria. La particolarità è che il sistema gestionale è tutto sviluppato online, sotto forma di web app, e, dunque, ha un'altissima portabilità, avendo come condizione, però, una connessione ad Internet.

Analogie

Anche con questo gestionale è possibile tenere sotto controllo tutti gli iscritti, gli eventuali, certificati, con tanto di promemoria sulle scadenze, e i loro profili societari. Nell'area gestionale, definita così nel sito ufficiale di *Sportivi in cloud*, oltre a quelle degli iscritti, si fa riferimento a tutta una serie di funzionalità dedicate alle attrezzature che appartengono alla società e ad altre, invece, pensate per l'area del merchandising. Dunque, facendo un paragone con il gestionale per il rugby Falconara, si può dire che vengono completamente implementate, seppur con alcune differenze tra le funzionalità, le aree dedicate agli iscritti e all'inventario.

Tra le varie cose che il gestionale permette di fare, c'è anche la possibilità di dividere gli iscritti in base a delle categorie di appartenenza, definendone implicitamente un ruolo, e anche di tenere traccia di chi ha pagato la quota annuale, chi è socio o chi ha particolari certificati, come il *BLSD*.

Differenze

Va detto che il gestionale *Sportivi in cloud* ha sicuramente una caratura più professionale e, per così dire, burocratica. L'applicazione in questione presta molta attenzione a guidare ed aiutare l'utente nella corretta gestione degli iscritti e del materiale nell'inventario; infatti, tra le funzionalità, a differenza del gestionale descritto nella trattazione precedente, se ne possono trovare alcune molto specifiche, come *polizze e assicurazioni*, per quanto riguarda l'attrezzatura, o *diplomi e abilitazioni* degli iscritti. Oltre a ciò, poi, sempre nell'area gestionale, sono presenti tante funzionalità, raggruppate per categoria, come, ad esempio, quelle dedicate alle comunicazioni e alle news, o, ancora, quelle dedicate alle informazioni generali della società stessa.

Una grossa differenza tra i due gestionali sta, poi, nella scelta di come sviluppare l'applicazione. Infatti, *Sportivi in cloud* sviluppa un'area finanziaria nel minimo dettaglio, sempre progettata con un'accezione molto burocratica e professionale; tuttavia, a differenza del sistema gestionale per il rugby, non fornisce nessun supporto per l'area tecnica, non consentendo di organizzare gli eventi, nè di inserire formazioni nè, tanto meno, di avere un sistema di analisi degli stessi.

CAPITOLO 7

Conclusioni

Questa trattazione riguardava la progettazione di un sistema gestionale che potesse aiutare nell'organizzazione delle operazioni quotidiane della società sportiva rugby Falconara Dinamis ASD.

Si è partiti dando una descrizione generale dei gestionali elencandone tipologie e caratteristiche principali, motivando perchè sarebbe opportuno utilizzarli in qualsiasi ambito, sia professionale che sportivo. Si è posta l'attenzione anche sulle sfide e sulle problematiche che, sviluppando un sistema informatico di questo tipo, si devono fronteggiare e, naturalmente, provare a risolvere.

Dopo questa prima parte descrittiva, si è proceduto parlando della società di rugby, della sua struttura e delle relative problematiche organizzative, e si è visto che sono principalmente dovute ad un'inadatta gestione delle risorse. Fatto ciò, si sono definiti i requisiti e le specifiche, che avrebbero, poi, indirizzato tutta la progettazione del gestionale; durante l'analisi dei requisiti si sono inziate a fare le prime scelte di progetto: si sono individuate 3 macro aree funzionali, *iscritti*, *inventario* e *eventi*, si sono formalizzati i requisiti stessi mettendoli in una forma adatta allo sviluppo progettuale, dividendoli in funzionali e non funzionali, ed, infine, si sono creati i casi d'uso, gli scenari che poi si sarebbero verificati e, di conseguenza, gestiti nel programma.

Ideato l'insieme base delle funzionalità del programma, si è definito lo schema delle entità e relazioni principali del sistema. Si è, dunque, definito un primo modello dei dati partendo dal linguaggio naturale, per poi creare un primo diagramma *entità-relazione*, definito come *schema concettuale*. Da questo schema si sono fatte delle analisi logiche, volte alla ristrutturazione del diagramma stesso; si è verificata la presenza di ridondanze e di gerarchie fra entità, valutando, per ogni caso, cosa fosse opportuno fare. Poi, si sono effettuati accorpamenti e partizionamenti di entità, a seconda che queste fossero troppo semplici o, viceversa, complesse, ed, infine, si sono eliminati eventuali attributi multivalore, non essendo essi rappresentabili in una base di dati. A questo punto, dopo aver fatto una traduzione dei dati analizzati nel modello relazionale, si è definito lo schema E/R finale del gestionale.

Completate le attività precedenti, che erano principalmente introduttive, si è passato a descrivere la progettazione della componente applicativa. Si è iniziato parlando dell'architettura del sistema; dunque, prima si sono analizzate le principali funzionalità del software, definendone per ognuna diversi tipi di diagrammi. Inoltre, si sono analizzate le classi, che avrebbero composto il gestionale, usando il diagramma delle classi di progettazione, e, successivamente, si sono utlizzati dei diagrammi delle macchine a stati per descrivere gli stati possibili degli

oggetti delle classi *iscritto* e, generalmente, *oggetto*. Analizzato tutto il contesto funzionale, si è parlato delle scelte architetturali, ed in particolare della scelta di implementare il pattern *MVC*. Definito il *back end* del programma si è parlato dell'interfaccia utente, descrivendola e riassumendo le componenti principali, coma la scelta di creare diverse sezioni per ogni categoria in ogni macro area. Inoltre, si è voluto porre l'attenzione su tutte quelle scelte di design che sono state prese per aumentare l'usabilità del gestionale, come, ad esempio, la sezione delle notifiche dei certificati. Seguendo il discorso delle scelte volte all'usabilità si sono descritte le principali funzionalità implementate per supportare le esigenze della società di rugby, dunque, ad esempio, si è parlato della facilità di navigazione tra le varie parti del sistema, della struttura delle notifiche dei certificati e dell'accessibilità limitata alle funzioni da parte di utenti aventi ruoli diversi. In ultima istanza si è parlato dell'affidabilità del sistema; si è posta l'attenzione sulla sicurezza nell'utilizzo del database e sulla poca possibilità di commettere errori da parte dell'utente; si è, poi, dedicata una sottosezione anche ai test generati per controllare il corretto funzionamento delle funzioni del database.

Terminata la progettazione si è discussa l'implementazione. Si è parlato delle librerie usate per gestire i dati del database, in particolare *SQLAlchemy*; poi, si sono descritti tutti gli strumenti e le tecnologie adoperate per un corretto ed efficace sviluppo del software, come *Xampp*, per l'utilizzo di server locali, o *QtDesigner*, per lo sviluppo dell'interfaccia. A questo punto si è creato un manuale utente per guidare ulteriormente l'utilizzatore nelle funzionalità del programma. L'ultimo capitolo della trattazione è stato dedicato ad una approfondita analisi qualitativa del software; a tale fine, è stata utilizzata la tecnica di analisi *SWOT* per ricercare punti di forza, di debolezza, oltre che opprtunità e minacce. Si è visto che, probabilmente, sarebbe stato necessario porre maggior attenzione nelle funzionalità di accesso e registrazione, ma si è, anche, elogiata la versatilità e semplicità di utilizzo, nonostante la completezza, del software. Alla fine si è svolta un indagine, elencando analogie e differenze, fra i gestionali sportivi più famosi in Italia, soprattuto in ambito rugbistico, e quello implementato.

Il sistema gestionale risulta, dunque, terminato, visto che tutti i requisiti e le specifiche, richieste dalla società di Falconara sono stati completamente implementati in un software che è, ormai, pronto per l'utilizzo. Perciò, si potrebbe definire il gestionale come completo dal punto di vista realizzativo, anche se esso lascia a disposizione diverse opportunità per il futuro sviluppo del programma. Potrebbe essere molto utile per la società di rugby avere a disposizione un'app mobile, soprattutto per gli allenatori, così da poter usufruire delle potenzialità di analisi degli allenamenti e delle partite molto più facilmente rispetto che adesso; ovviamente, un'app mobile, sarebbe utile anche in altri contesti relativi al gestionale, come, ad esempio, quello relativo all'area del merchandising. Poi, una volta creata l'applicazione, sarà doveroso implementare funzionalità online, permettendo, così, di poter usufruire del gestionale, condividendo, con gli altri membri della società, le proprie operazioni e le modifiche effettuate. Soltanto con queste due modifiche il gestionale farebbe un grande salto qualitativo, diventando davvero competitivo con gli altri programmi del suo genere.

Infine, una cosa che potrebbe essere maggiormente sviluppata è l'area relativa all'analisi degli eventi sportivi. Questa, così come è ora, è già uno dei punti di forza del gestionale che lo rende diverso, e maggiormente adattabile, rispetto agli altri programmi presenti sul mercato. Sicuramente, però, potrebbe essere sviluppata in maniera più dettagliata, implementando una serie di funzionalità, possibilmente con l'aiuto di esperti del settore, che permetterebbero di far diventare quest'area del gestionale quasi un software di analisi tecnica separato.

In generale, il gestionale, è stato progettato avendo una forma e una struttura favorevole al cambiamento; non sarà un problema aggiornare il programma apportando le modifiche sopra descritte.

Bibliografia

BLAHA, M., LORENSEN, W., RUMBAUGH, J., EDDY, F. e PREMERLANI, W. (2004), *Object-Oriented Modeling And Design With Uml 2Nd Edition*.

ELMASRI, R. e NAVATHE, S. B. (1994), *Fundamentals of Database Systems*.

FOWLER, M. (1997), *UML Distilled: A Brief Guide to the Standard Object Modeling Language*.

LARMAN, C. (1997), *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*.

LAUDON, K. C. e LAUDON, J. P. (2021), *Management Information Systems: Managing the Digital Firm*.

LEON, A. e LEON, M. (2008), *Enterprise Resource Planning*.

MARTIN, R. C. (2008), *Clean Code: A Handbook of Agile Software Craftsmanship*.

SEIDL, M., SCHOLZ, M., HUEMER, C. e KAPPEL, G. (2015), *UML @ Classroom: An Introduction to Object-Oriented Modeling*.

SOMMERVILLE, L. (2017), *Introduzione all'ingegneria del software*, 10 ed.

SPETH, C. (2023), *L'analisi SWOT: Uno strumento fondamentale per lo sviluppo della strategia aziendale*.

TEOREY, T. J., LIGHTSTONE, S. S. e NADEAU, T. (2005), *Database Modeling and Design: Logical Design*.

WIEGERS, K. e BEATTY, J. (2013), *Software Requirements*.

Siti web consultati

- Datalog: "software gestionale: che caratteristiche deve avere?" – `https://www.datalog.it/software-gestionale-che-caratteristiche-deve-avere/`
- TechTarget-ERP – `https://www.techtarget.com/searcherp/`

- Visual Paradigm-UML Guide – `https://www.visual-paradigm.com/guide/`
- Lucidchart-Software Engineering – `https://www.lucidchart.com/pages/solutions/engineering`
- MindTools-SWOT Analysis – `https://www.mindtools.com/amtbj63/swot-analysis`
- Gartner – `https://www.gartner.com/en/information-technology`
- Bizzdesign-Enterprise Architecture – `https://bizzdesign.com/blog/category/enterprise-architecture/`
- IBM-Software Engineering Principles – `https://www.ibm.com/topics/software-enginee`
- SQLCourse – `https://www.sqlcourse.com/`
- MySql Documentation – `https://dev.mysql.com/doc/`
- SqlAlchemy Documentation – `https://www.sqlalchemy.org/library.html`

Ringraziamenti

Prima di tutto devo ringraziare la mia famiglia, mamma Viridiana e babbo Paolo su tutti, per avermi sostenuto durante il percorso di studi, fin qui intrapreso. Inoltre, mi sento di dover ringraziare le mie nonne, Luisa e Cesarina, in quanto mi sono sempre state vicine in questi tre anni. Voglio anche dire grazie ai miei nonni, Giuseppe e Gianni, che, anche senza essere fisicamente con me, sono sicuro che mi hanno accompagnato durante ogni momento e sarebbero sicuramente stati orgogliosi di vedermi arrivato a questo punto.

Ringrazio tantissimo Elena, la mia ragazza, per avermi supportato e, soprattuto, sopportato con amore in ogni momento, sia felice sia triste, che ho avuto durante la laurea triennale. In questo momento lei sta terminando il corso di studi in fisioterapia, perciò colgo l'occasione per ricordarle che deve tenere duro e continuare a credere in se stessa, proprio come lei ha fatto con me. Grazie, poi, anche a Samuele e Daniel, i miei compagni di corso, con cui ho condiviso gran parte degli esami che ho sostenuto; il mio augurio va anche a loro, nella speranza che possano realizzarsi al meglio negli studi.

Grazie mille, anche, al professor Ursino per essersi reso sempre molto disponibile in ogni momento degli ultimi mesi, ed avermi aiutato, sempre con gentilezza e puntualità, nella scrittura della tesi.

Infine, grazie alla società Rugby Falconara Dinamis e al suo presidente Stefano per essersi messi a disposizione e aver fornito le informazioni per la costruzione del gestionale descritto nella tesi.