

# UNIVERSITÀ DEGLI STUDI DI ROMA TOR VERGATA



**Documentazione progetto ISPW**

**Federico Guglielmini**

## Sommario

1	Specifiche dei requisiti software .....	3
1.1.1	Introduzione.....	3
1.1.2	Panoramica del sistema .....	3
1.1.3	Requisiti Hardware e Software .....	3
1.1.4	Sistemi correlati.....	3
1.2	User Stories.....	3
1.2.1	US-1.....	3
1.2.2	US-2 .....	4
1.2.3	US-3 .....	4
1.3	Functional Requirements.....	4
1.3.1	FR-1.....	4
1.3.2	FR-2 .....	4
1.3.3	FR-3 .....	4
1.4	Use Case.....	4
1.4.1	Use Case Diagram.....	4
1.4.2	Use Case Internal Steps .....	5
2	Storyboards .....	6
3	Design .....	10
3.1	VOPC.....	10
3.2	Desing-Level.....	11
3.3	Design Patterns .....	11
3.4	Activity Diagram .....	12
3.5	Sequence Diagram.....	13
3.6	State Diagram.....	14
4.	Link utili .....	14

# 1 Specifiche dei requisiti software

## 1.1.1 Introduzione

Questo documento ha lo scopo di fornire una panoramica dettagliata di **CalcettoHub**, un'applicazione utile per la prenotazione di campi da calcetto e organizzazione di partite tra amici. Si tratteranno le funzionalità principali dell'applicazione andando ad evidenziare come è stata progettata e quali vantaggi offre.

## 1.1.2 Panoramica del sistema

L'applicazione è pensata per facilitare i giocatori e i gestori di campi da calcetto nella prenotazione delle partite, tramite funzionalità quali:

- **Prenotazione di un campo:** l'applicazione permette ai giocatori di cercare un campo tramite l'inserimento di un indirizzo o della città in cui si vuole cercare, visualizzando i risultati su una mappa, in modo da essere facilitati anche nel raggiungimento del campo per cui si effettua una prenotazione.
- **Gestione delle prenotazioni:** Gli utenti, sia giocatori che gestori, possono visualizzare le prenotazioni a loro associate, con la possibilità di effettuare una disdetta qualora sorgessero inconvenienti.

## 1.1.3 Requisiti Hardware e Software

**Requisiti software:** Java SDK, IDE: IntelliJ IDEA e database MySQL

**Requisiti hardware:** Processore moderno in grado di eseguire applicazioni java, computer con almeno 4 GB di RAM e connessione ad internet.

## 1.1.4 Sistemi correlati

**InCampo:** piattaforma web per la gestione di centri sportivi e associazioni permettendo agli utenti di prenotare campi (calcetto, padel, tennis ...) e ai gestori di gestire prenotazioni, tesserati e scadenze in modo automatico. Offre una interfaccia abbastanza intuitiva senza bisogno di registrazione. CalcettoHub offre un'interfaccia altrettanto user friendly con l'aggiunta che ti mostra i campi direttamente sulla mappa, così che si può scegliere quello più adatto alle esigenze dei giocatori, sia in termini di prezzo che di distanza, e inoltre si può raggiungere il campo scelto in maniera facile e veloce tramite navigatore.

# 1.2 User Stories

## 1.2.1 US-1

Come giocatore, voglio cercare i campi disponibili inserendo una località o un indirizzo e visualizzare i risultati su mappa, in modo da scegliere il campo con distanza, prezzo e fasce orari disponibili più adatto e prenotarlo.

### **1.2.2 US-2**

Come gestore del campo, voglio visualizzare e modificare lo stato delle prenotazioni dei campi che gestisco (confermare/disdire), in modo da organizzare l'operatività e informare i giocatori in caso di indisponibilità del campo.

### **1.2.3 US-3**

Come giocatore, voglio creare una partita scegliendo data, ora e un campo, tra quelli disponibili, e decidere se renderla pubblica o solo su invito, in modo da gestire le adesioni e avere una lista aggiornata dei partecipanti.

## **1.3 Functional Requirements**

### **1.3.1 FR-1**

Il sistema deve permettere all'utente di cercare i campi inserendo località o indirizzo, visualizzando i risultati su mappa.

### **1.3.2 FR-2**

Dopo che l'utente ha selezionato un campo e inserito una data, il sistema deve mostrare le fasce orarie disponibili per la prenotazione di quel campo in quella data. Se non ci sono fasce orarie disponibili, il sistema deve comunicarlo chiaramente all'utente.

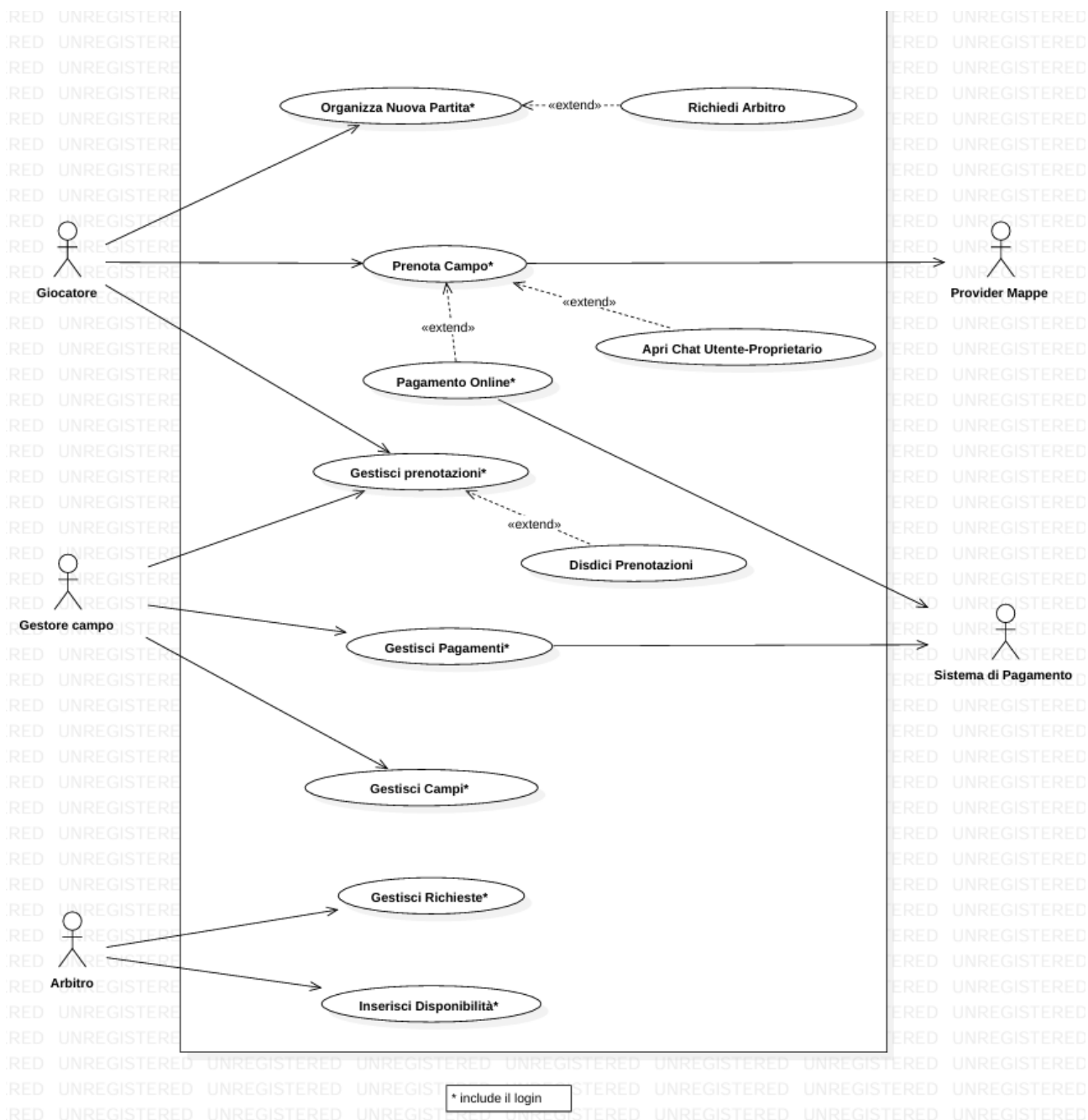
### **1.3.3 FR-3**

Ogni volta che viene effettuata una nuova prenotazione per un campo, il sistema deve inviare una notifica al gestore del campo contenente: campo prenotato, data di prenotazione, fascia oraria scelta ed e-mail dell'utente prenotante.

## **1.4. Use Case**

### **1.4.1 Use Case Diagram**

Di seguito vengono riportate le funzionalità principali di CalchettoHub.



## 1.4.2 Use Case Internal Steps

Nome use Case: Prenota Campo

1. L'utente effettua il login.
2. L'utente inserisce la località e/o l'indirizzo per cercare un campo disponibile.
3. Il sistema mostra i campi disponibili sulla mappa e su una lista dove sono specificati il nome del campo, indirizzo, località, tipo di superficie e prezzo.
4. L'utente seleziona un campo specifico e inserisce la data per cui vuole effettuare la prenotazione.
5. Il sistema mostra all'utente le fasce orarie disponibili per quella data.
6. L'utente seleziona la fascia oraria per cui vuole effettuare la prenotazione.

7. Il sistema mostra all'utente i metodi di pagamento disponibili (pagamento online oppure in contanti).
8. L'utente seleziona pagamento online e viene indirizzato al servizio pagamento online.
9. Il sistema registra la nuova prenotazione e invia una notifica al gestore del campo dell'avvenuta prenotazione.
10. Un giorno prima della partita, il sistema invia una notifica all'utente per ricordargli della sua prenotazione.

Estensioni:

2a. **Nessun campo disponibile:** il sistema informa l'utente che non ci sono campi disponibili per la località e/o indirizzo scelto.

4a. **Campo chiuso per la data scelta:** il sistema informa l'utente che il campo è chiuso nella data scelta; quindi, invita l'utente a inserire una nuova data di prenotazione.

4b. **Nessuna fascia oraria disponibile:** il sistema informa l'utente che tutte le fasce orarie presenti nella data scelta sono già state prenotate.

7a. **Pagamento in contanti:** l'utente seleziona pagamento in contanti. Il pagamento verrà effettuato al momento della partita. Vai allo step 9.

9a. **Servizio chat Utente-Proprietario:** il sistema offre all'utente la possibilità di aprire una chat con il gestore del campo per poter chiedere delle informazioni.

10a. **Recensione post-partita:** In seguito allo svolgimento della partita, il sistema invita l'utente a lasciare una recensione sul campo in cui si è tenuta la partita.

## 2 Storyboards

In questa sezione vengono mostrate alcune funzionalità, sotto forma di **storyboards**, dell'applicazione CalchettoHub.



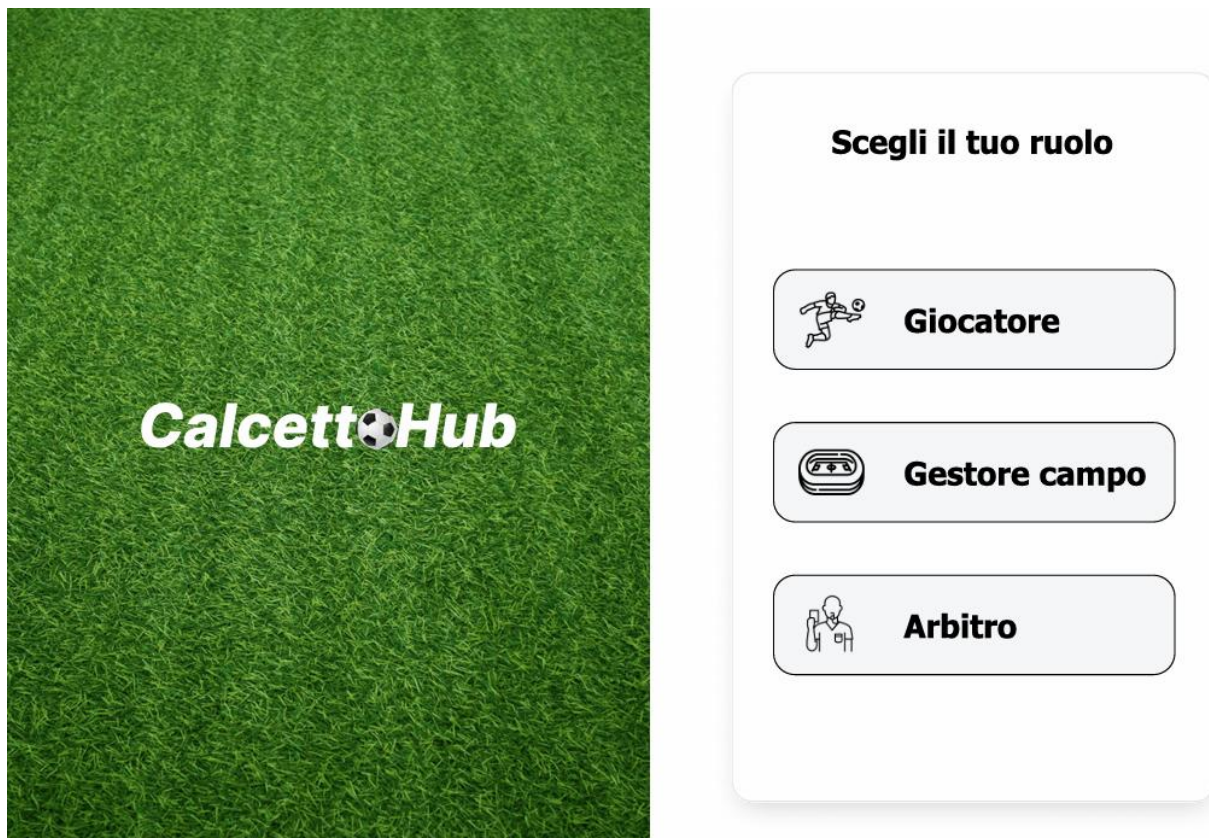


Figura 1: schermata della selezione ruolo

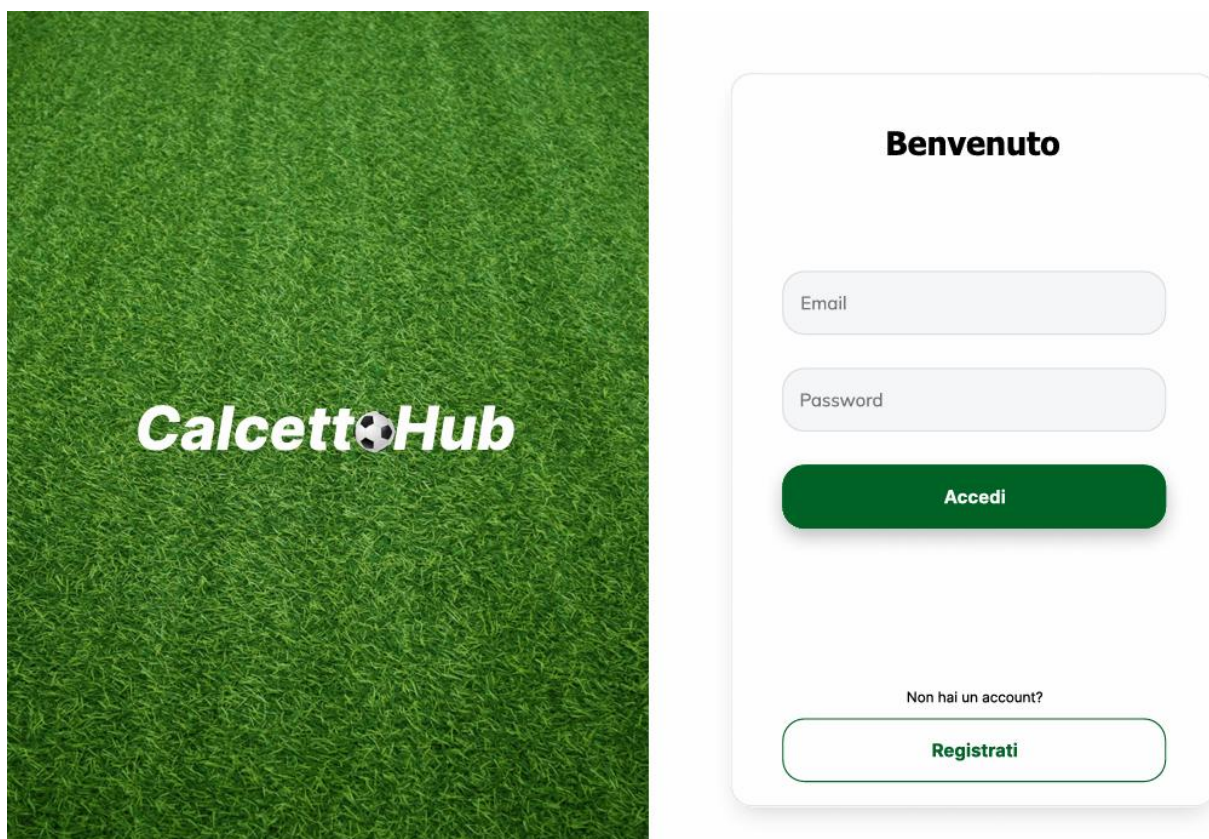
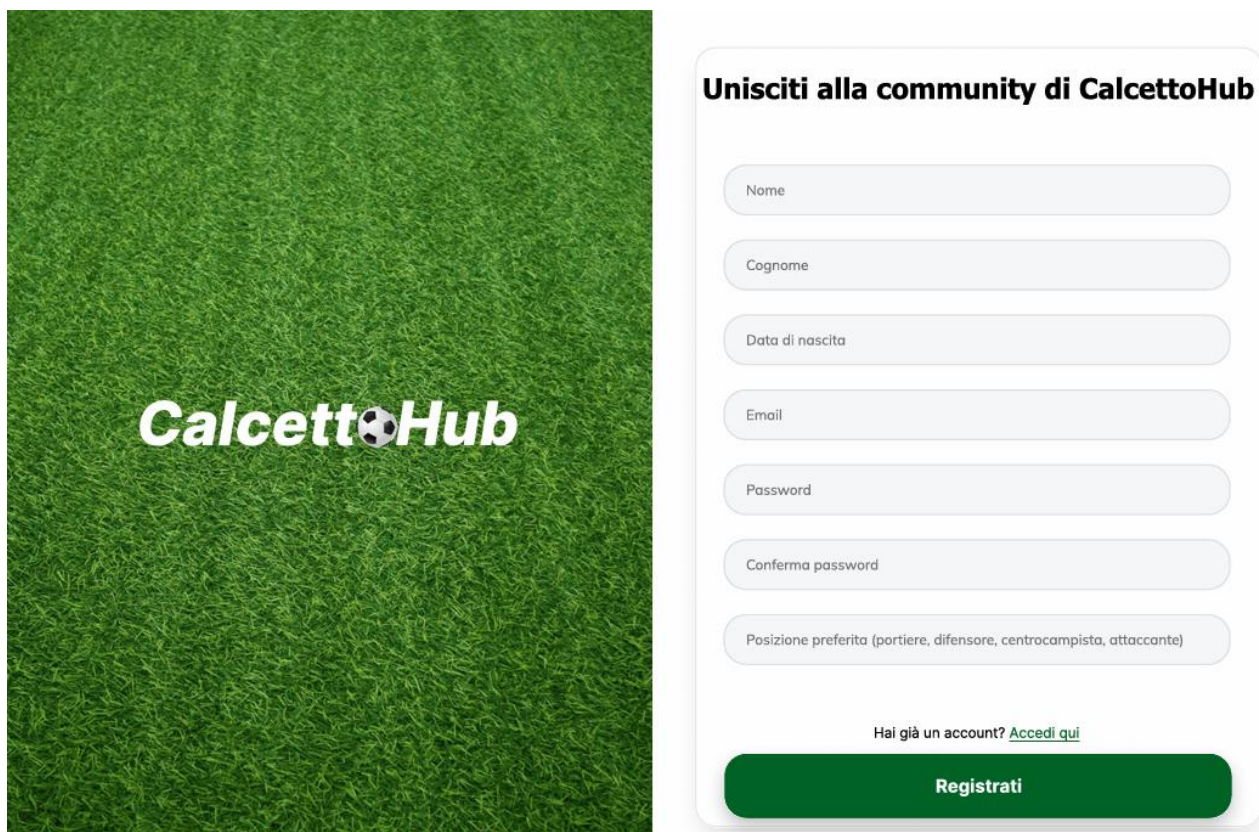


Figura 2: schermata del login





The registration screen features a green grass background on the left with the 'CalcettoHub' logo. On the right, a white card contains the title 'Unisciti alla community di CalcettoHub' and a series of input fields: 'Nome', 'Cognome', 'Data di nascita', 'Email', 'Password', 'Conferma password', and 'Posizione preferita (portiere, difensore, centrocampista, attaccante)'. Below these fields is a link 'Hai già un account? [Accedi qui](#)' and a large green 'Registrati' button.

**Unisciti alla community di CalcettoHub**

Nome

Cognome

Data di nascita

Email

Password

Conferma password

Posizione preferita (portiere, difensore, centrocampista, attaccante)

Hai già un account? [Accedi qui](#)

**Registrati**

Figura 3: schermata di registrazione per il giocatore

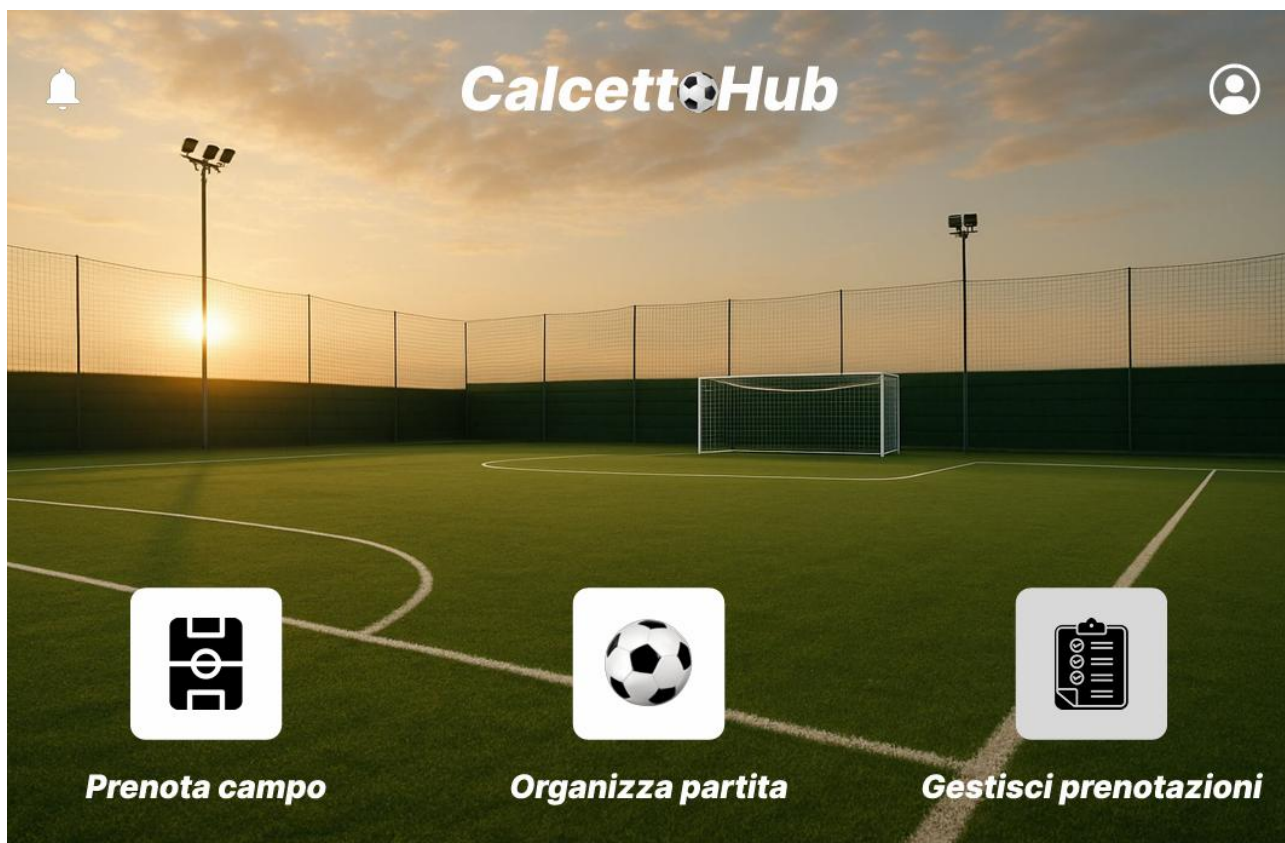


Figura 4: schermata home del giocatore





Figura 5: schermata home del gestore del campo

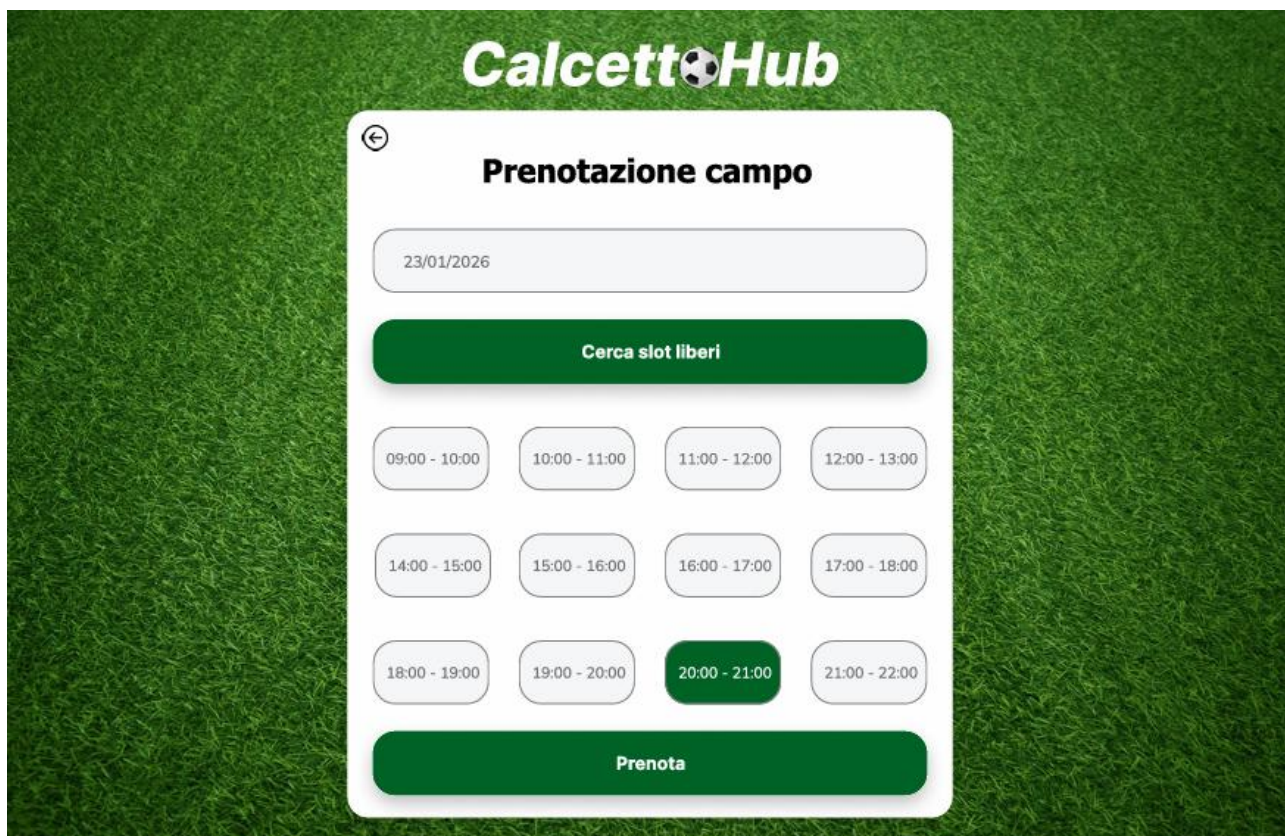
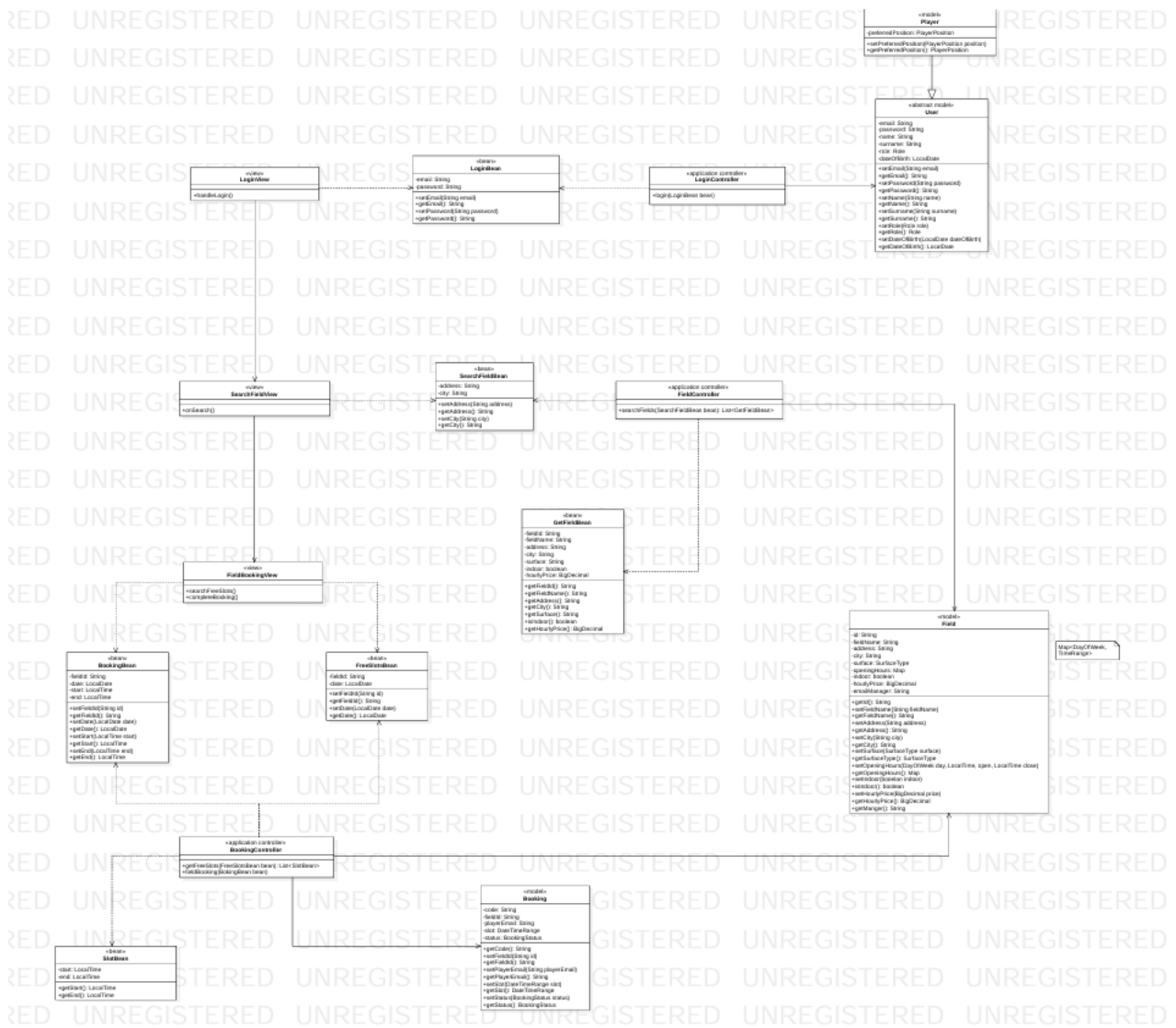


Figura 6: schermata della scelta della data e lo slot da prenotare

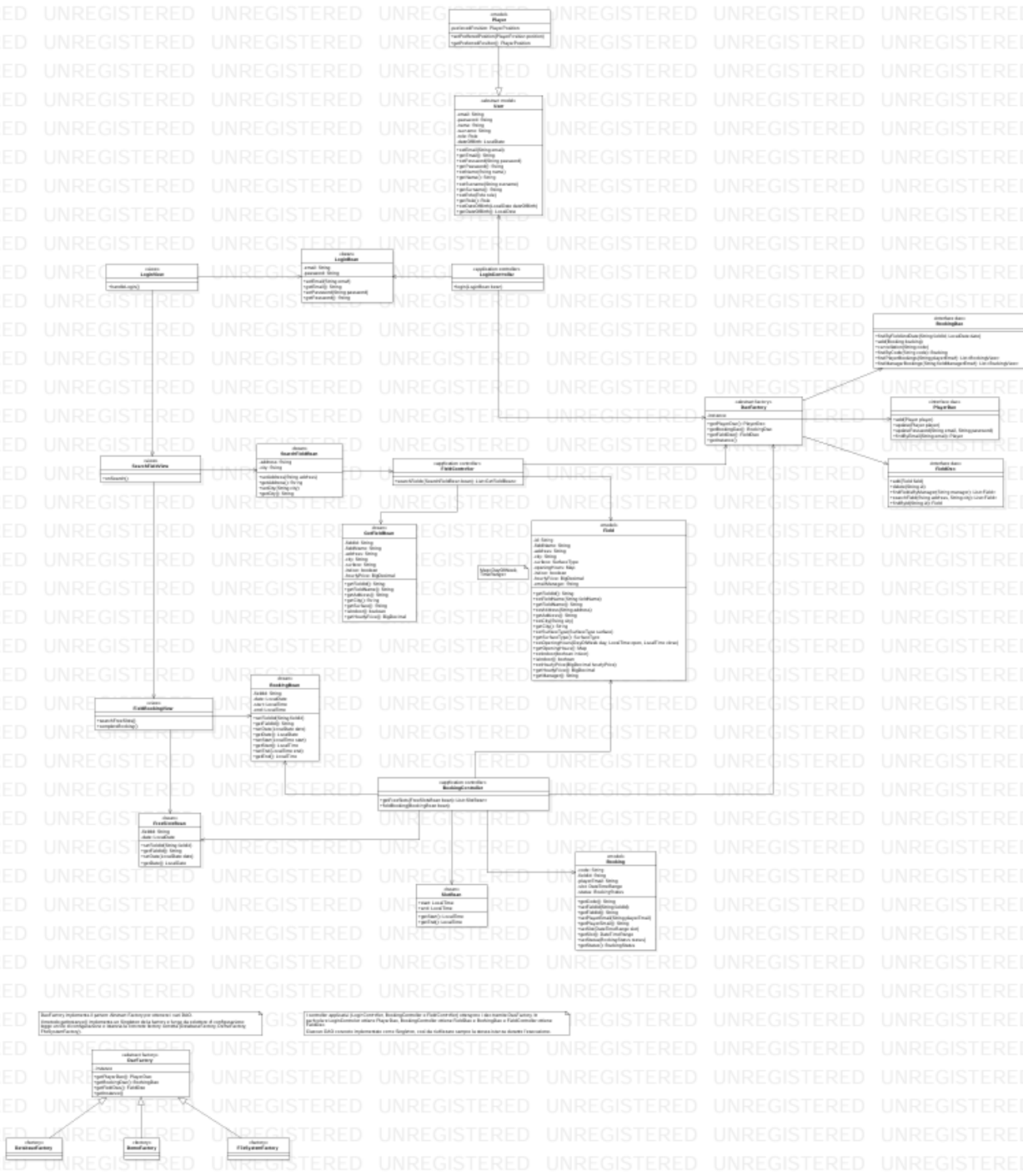
### 3 Design

Questi diagrammi sono visualizzabili in maniera più approfondita nella cartella /**Deliverables** del progetto.

### 3.1 VOPC



### 3.2 Desing-Level



### 3.3 Design Patterns

Sono stati utilizzati questi due pattern in particolare:

Ho utilizzato il pattern **Abstract Factory** per creare famiglie di Dao in base alla configurazione scelta per l'applicazione (database, demo o file system), configurazione letta dal file *config.properties*. Inoltre, ho adottato il singleton per garantire che esista una sola istanza della factory e dei singoli dao, evitando creazioni multiple e assicurando un

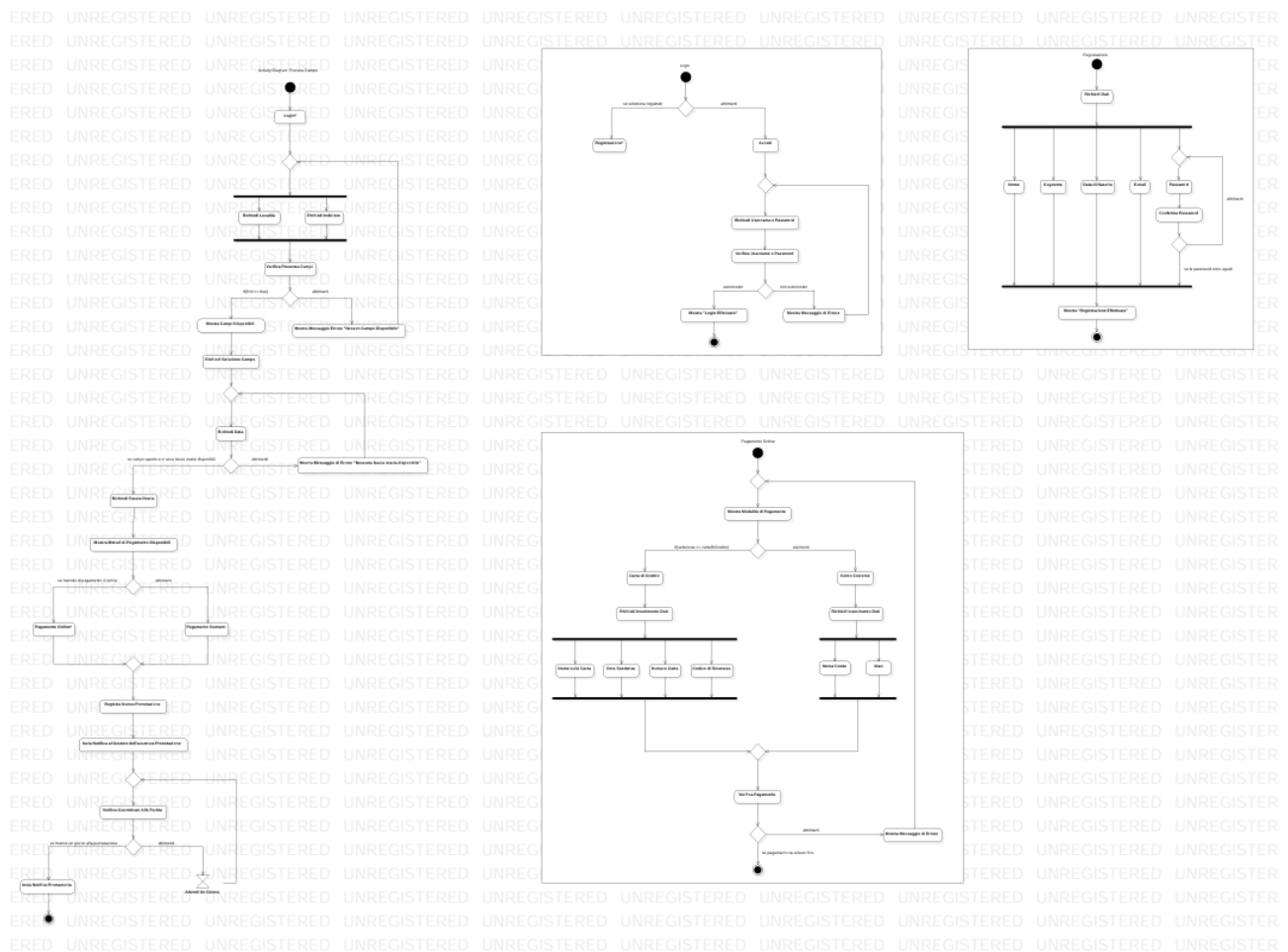
punto di accesso globale e consistente agli oggetti di persistenza. Ho applicato il pattern singleton anche ad altri componenti, non solo per i dao, in particolare per gestire la connessione al database, gestire la sessione e gestire il repository in-memory demo. In particolare, è stato applicato in:

- SessionManager: per gestire in modo centralizzato la sessione dell'utente (login, scadenza e refresh);
- DatabaseConnction: per garantire un unico punto di accesso alla connessione JDBC e alla configurazione del DB;
- DemoRepository per condividere un'unica "memoria" in modalità demo tra tutti i DAO demo.

In questo modo evito istanze multiple, duplicazione di configurazione e incoerenze di stato.

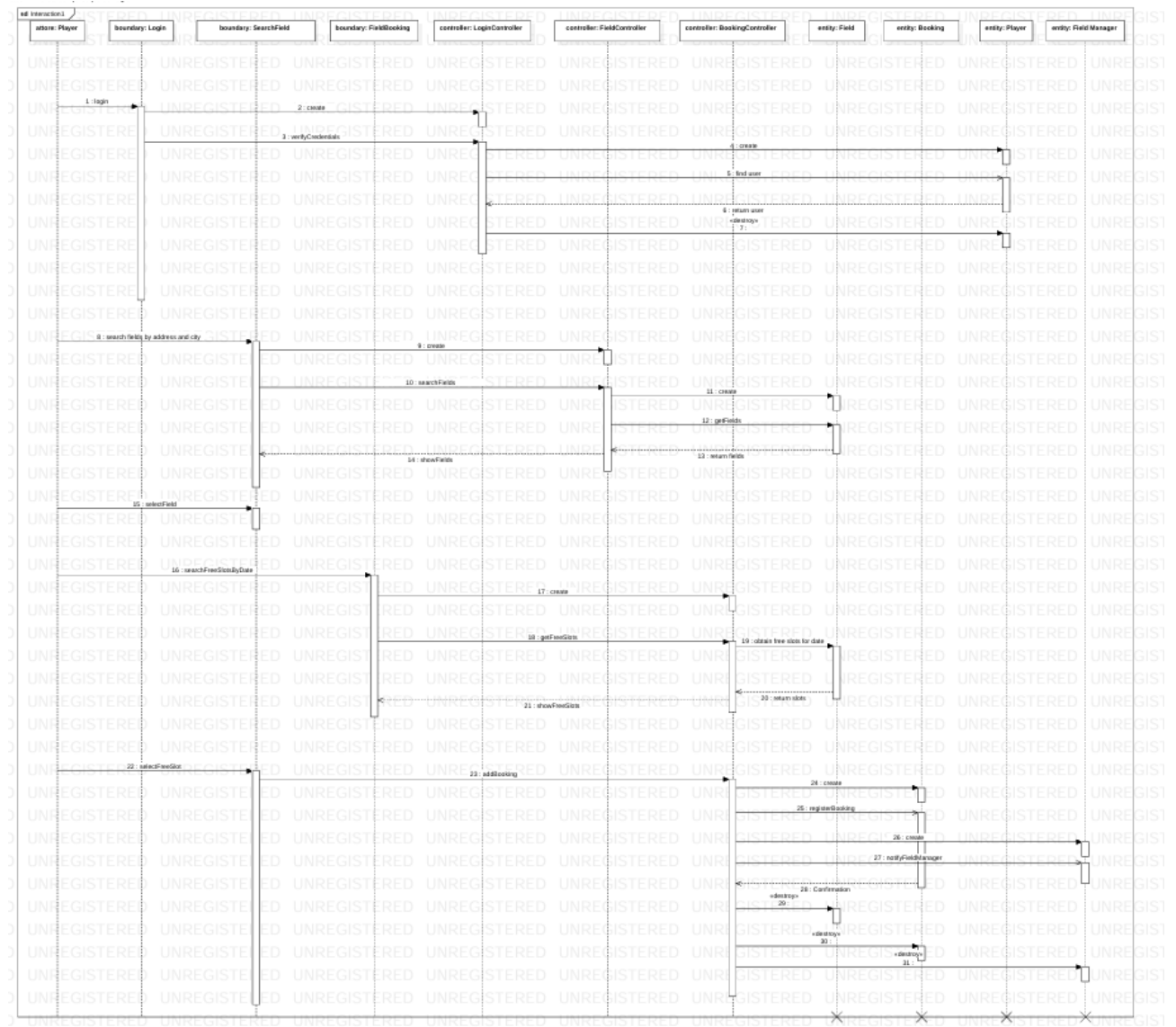
**Nota:** i successivi diagrammi sono visualizzabili in maniera più dettagliata nella cartella del progetto **/Deliverables**.

### 3.4 Activity Diagram





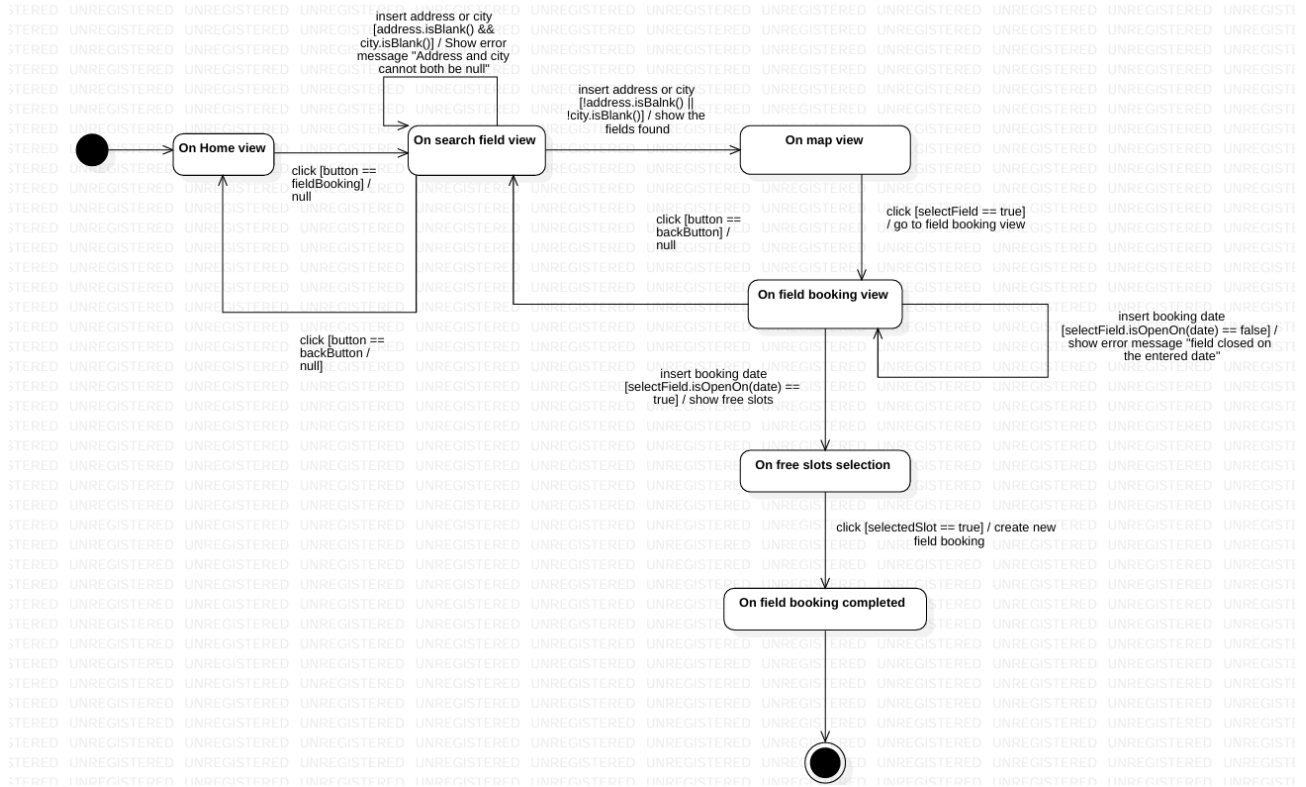
### 3.5 Sequence Diagram





## 3.6 State Diagram

State Diagram di Prenota Campo



## 4. Link utili

**GitHub:** <https://github.com/Federico10012004/Ingegneria-del-software.git>

**SonarCloud:** <https://sonarcloud.io/organizations/federico10012004/projects>

**Video applicazione:** <https://youtu.be/5HzV5OjYdyY>