

MatchDay
Requirements Analysis Document
Versione 2.5



MatchDay

Data: 14/10/2024

| | |
|---|------------------|
| Progetto: MatchDay | Versione: 2.2 |
| Documento: Requirements Analysis Document | Data: 14/10/2024 |

Coordinatore del progetto:

| Nome | Matricola |
|------|-----------|
| | |
| | |

Partecipanti:

| Nome | Matricola |
|-------------------|------------|
| Nicola Moscufo | 0512114886 |
| Francesco Moscufo | 0512115027 |
| Vincenzo Vitale | 0512113542 |
| | |
| | |
| | |

| | |
|--------------------|--|
| Scritto da: | Vincenzo Vitale, Nicola Moscufo, Francesco Moscufo |
|--------------------|--|

Revision History

| Data | Versione | Descrizione | Autore |
|------------|----------|--|--|
| 09/11/2024 | 1.0 | Aggiunta di Class Diagram, Sequence Diagram, Activity Diagram e Mock-ups | Vincenzo Vitale, Nicola Moscufo, Francesco Moscufo |

| | | | |
|------------|-----|---|--|
| 20/11/2024 | 2.0 | Modifica Class Diagram, Sequence Diagram, Activity Diagram | Vincenzo Vitale, Nicola Moscufo, Francesco Moscufo |
| 23/12/2024 | 2.1 | Revisione e modifica di Sequence e Activity Diagram | Vincenzo Vitale, Nicola Moscufo, Francesco Moscufo |
| 29/12/2024 | 2.2 | Modifica dei Sequence e Activity Diagram e aggiornamento dei Mock-ups | Francesco Moscufo, Vincenzo Vitale, Nicola Moscufo |
| 05/01/2025 | 2.3 | Modifica Class Diagram | Francesco Moscufo, Vincenzo Vitale, Nicola Moscufo |
| 08/01/2025 | 2.4 | Modifica Class Diagram e aggiunta Mock-ups | Francesco Moscufo, Vincenzo Vitale, Nicola Moscufo |
| 17/01/2025 | 2.5 | Modifiche: Class Diagram, Casi D'uso e Sequence Diagram | Francesco Moscufo, Vincenzo Vitale, Nicola Moscufo |
| 17/01/2025 | 2.6 | Modifica al class diagram con nuove relazioni appropriate. | Francesco Moscufo, Vincenzo Vitale, Nicola Moscufo |
| | | Ingegneria del Software | Pagina 2 di 31 |

Indice

Sommario

| | |
|--|---|
| 1. INTRODUZIONE..... | 4 |
| 1.1. Scopo del sistema..... | 4 |
| 1.2. Ambito del sistema | 4 |
| 1.3. Obiettivi e criteri di successo del sistema | 4 |
| 1.4. Definizioni, acronimi, abbreviazioni | 5 |
| 1.5. Panoramica..... | 5 |
| 2. SISTEMA PROPOSTO | 5 |
| 2.1. Panoramica..... | 5 |
| 2.2. Requisiti funzionali..... | 6 |
| 2.3. Requisiti non funzionali..... | 7 |
| 2.3.1. Usabilità | 7 |

| | |
|-------------------------------|----|
| 2.3.2. Affidabilità | 7 |
| 2.3.3. Prestazioni | 7 |
| 2.3.4. Sostenibilità | 7 |
| 2.3.5. Implementazione | 8 |
| 2.3.6. Interfaccia | 8 |
| 2.4. Modelli di sistema | 8 |
| 2.4.1. Class Diagram | 8 |
| 2.4.2. Scenari | 8 |
| 2.4.3. Casi d'uso | 10 |
| 2.4.4. Sequence Diagram | 17 |
| 2.4.5. Activity Diagram | 21 |
| 2.4.6. Mock-ups | 22 |

1. INTRODUZIONE

1.1. Scopo del sistema

Lo scopo di MatchDay è sviluppare un sistema software che automatizzi la gestione delle prenotazioni di campi sportivi, permettendo agli utenti di verificare la disponibilità in tempo reale, effettuare prenotazioni e ricevere conferme immediate, migliorando l'efficienza e l'esperienza utente. Inoltre, il sistema fornirà agli amministratori una piattaforma centralizzata per gestire e monitorare le prenotazioni, riducendo errori e disorganizzazione.

1.2. Ambito del sistema

MatchDay è progettato per entrare nel settore della gestione delle prenotazioni sportive online, rendendo più semplice per gli utenti prenotare campi sportivi.

Il sistema deve supportare le seguenti funzionalità:

- Prenotazione di campi sportivi
- Aggiunta e gestione di prenotazioni multiple
- Visualizzazione della disponibilità dei campi in tempo reale
- Notifiche di conferma, modifica o cancellazione delle prenotazioni
- Visualizzazione dello storico delle prenotazioni

1.3. Obiettivi e criteri di successo del sistema

Elenco obiettivi:

- Consentire a tutti gli utenti registrati di visualizzare i campi sportivi disponibili
- Visualizzare in tempo reale la disponibilità dei campi
- Permettere la prenotazione dei campi in pochi passaggi
- Visualizzare e gestire lo storico delle prenotazioni effettuate

Criteri di successo:

- Disponibilità: il sistema deve essere accessibile 24/7
- Sicurezza e privacy: autenticazione sicura e crittografia dei dati
- Scalabilità: supportare l'accesso simultaneo di più utenti senza perdita di prestazioni
- Robustezza: gestire correttamente tutte le interazioni dell'utente e prevenire errori o prenotazioni doppie
- Tempo di risposta: il sistema deve rispondere entro pochi secondi per le operazioni principali

1.4. Definizioni, acronimi, abbreviazioni

RAD: Requirements Analysis Document

SDD: System Design Object

ODD: Object Design Document

UC: Use Case

RF: Requisito Funzionale

NRF: Requisito Non Funzionale

Gestore: L'utente responsabile della gestione operativa delle prenotazioni. Il gestore ha il compito di confermare, modificare e cancellare le prenotazioni effettuate dagli utenti. Inoltre, ha l'autorità di aggiungere, aggiornare e rimuovere campi sportivi nel sistema. Utente: Un utilizzatore del software registrato correttamente

1.5. Panoramica

Il seguente documento RAD è diviso in sezioni che insieme specificano i dettagli di questo progetto.

2. SISTEMA PROPOSTO

2.1. Panoramica

Il software è utilizzato da utenti correttamente registrati.

Utente Registrato:

- Visualizzare i campi presenti e gli orari associati ai campi
- Visualizzare le fasce orarie disponibili per il campo selezionato
- Può inviare una richiesta di prenotazione
- Può inviare una richiesta di modifica della prenotazione

- Può cancellare la sua prenotazione
- Visualizzare lo storico delle prenotazioni Gestore:
- Aggiunta di campi sportivi
- Modifica di campi sportivi
- Eliminazione dei campi sportivi
- Confermare/rifiutare prenotazioni
- Confermare/rifiutare le modifiche alle prenotazioni

2.2. Requisiti funzionali

Priorità Alta: funzionalità strettamente necessaria al corretto funzionamento del software
 Priorità Media: funzionalità non strettamente necessaria al corretto funzionamento del software

| Identificativo | Priorità | Descrizione |
|----------------|----------|---|
| FR_1 | Alta | All'avvio dell'app, gli utenti devono completare la registrazione per accedere alla home page |
| FR_2 | Alta | Gli utenti devono registrarsi e accedere utilizzando le proprie credenziali. |
| FR_3 | Alta | Gli utenti possono visualizzare i campi disponibili e le fasce orarie libere. |
| FR_4 | Alta | Gli utenti possono prenotare un campo per un giorno e un orario specifico. |
| FR_5 | Alta | Il gestore può aggiungere, modificare e rimuovere campi e gestire le prenotazioni |
| FR_6 | Alta | Gestori e utenti possono visualizzare lo storico di tutte le prenotazioni effettuate |

2.3. Requisiti non funzionali

| Identificativo | Descrizione |
|----------------|--|
| NFR_1 | Il sistema deve essere veloce e reattivo, con tempi di risposta ridotti. |
| NFR_2 | Le informazioni e i dati personali degli utenti devono essere protetti da crittografia tramite Firebase. |
| NFR_3 | Il sistema deve essere scalabile, in grado di supportare un numero crescente di utenti e prenotazioni |
| NFR_4 | Deve essere garantito il supporto per accessi concorrenti attraverso Firebase Realtime Database. |

2.3.1. Usabilità

Il sistema deve offrire un'esperienza utente fluida e intuitiva, con un'interfaccia semplice che consenta agli utenti di effettuare le prenotazioni in modo rapido.

2.3.2. Affidabilità

Il sistema deve garantire un'elevata affidabilità, con un'architettura solida in grado di gestire un grande numero di accessi concorrenti senza compromettere le prestazioni.

2.3.3. Prestazioni

Le prestazioni del sistema sono fondamentali e devono soddisfare i requisiti di velocità e reattività, con tempi di risposta rapidi. Il sistema deve garantire un'ottimizzazione delle risorse per mantenere una performance elevata, anche con un numero crescente di utenti e prenotazioni.

2.3.4. Sostenibilità

Il sistema deve essere facilmente mantenibile e aggiornabile, con l'uso di tecnologie scalabili come Firebase. Eventuali aggiornamenti e miglioramenti delle funzionalità devono essere implementati senza compromettere la sicurezza e la stabilità dell'applicazione.

2.3.5. Implementazione

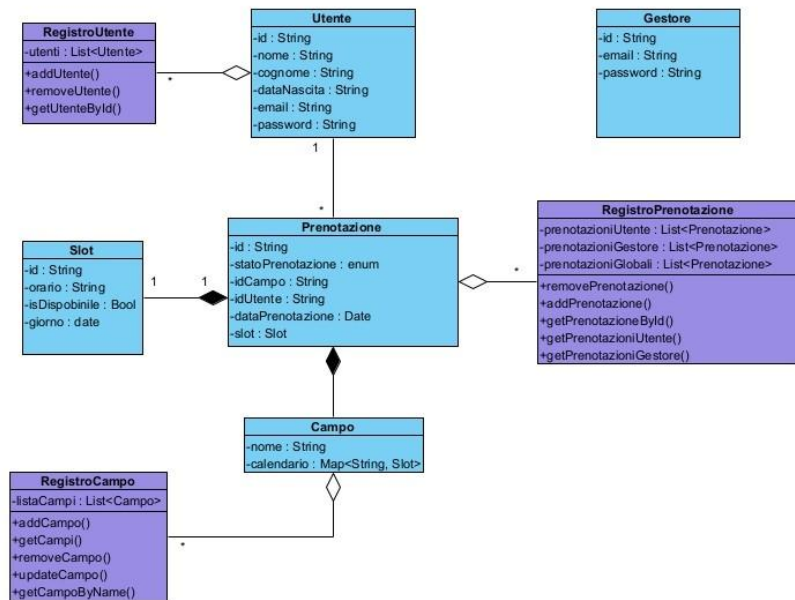
L'implementazione del sistema deve seguire un'architettura basata su Firebase, per garantire la sicurezza dei dati con crittografia. Le funzionalità del sistema devono essere progettate per supportare accessi concorrenti attraverso il Firebase Realtime Database, garantendo scalabilità e prestazioni elevate.

2.3.6. Interfaccia

L'interfaccia utente deve essere progettata per migliorare l'esperienza dell'utente, mantenendo un design chiaro e reattivo.

2.4. Modelli di sistema

2.4.1. Class Diagram



2.4.2. Scenari

SC_1: Richiesta di prenotazione da parte di un cliente

Marco accede al software, seleziona il campo che preferisce, sceglie dal calendario un giorno che preferisce e in seguito seleziona uno slot di orario di suo gradimento. Una volta selezionato lo slot, la richiesta verrà elaborata dal sistema e resterà in attesa di una conferma o un rifiuto da parte del gestore.

SC_2: Conferma richiesta di una prenotazione

Mario, il gestore del centro sportivo, riceve una richiesta di prenotazione attraverso l'applicazione dedicata al centro sportivo. Accedendo alla sezione prenotazioni, Mario visualizza una lista di richieste di prenotazione con stato "In attesa".

Tra queste, individua la prenotazione di Marco, la quale è elencata con i relativi dettagli (nome utente, data, orario e campo richiesto). Mario seleziona la richiesta specifica e clicca sull'opzione "Conferma prenotazione".

A seguito di questa azione: 1. Il sistema aggiorna automaticamente lo stato della richiesta di prenotazione a "Confermata".

2. Il sistema di gestione delle disponibilità registra lo slot orario richiesto come "Non disponibile" nel calendario delle prenotazioni.

3. L'operazione viene registrata nel database del sistema e la prenotazione confermata è visibile esclusivamente nella piattaforma di gestione.

SC_3: Rifiuto di una richiesta di prenotazione

Mario, il gestore del centro sportivo, riceve una richiesta di prenotazione per il campo di padel tramite l'applicazione del centro. La richiesta, effettuata da Luigi, riguarda il giorno 15 gennaio dalle 15:00 alle 16:00. Accedendo alla sezione prenotazioni, Mario individua la richiesta di Marco tra le pendenti e procede con il rifiuto cliccando sul pulsante "Rifiuta richiesta".

Il sistema aggiorna lo stato della prenotazione di Marco a "annullata" e lo slot orario corrispondente torna ad essere "Disponibile".

SC_4: Modifica di una prenotazione

Con il sistema proposto, Laura può ora accedere alla lista delle prenotazioni confermate dal gestore e selezionare l'opzione per modificarla. La modifica viene effettuata scegliendo un nuovo orario o il giorno direttamente dal calendario.

L'app consente di modificare la prenotazione scegliendo un nuovo giorno e orario, a condizione che la modifica venga effettuata entro 24 ore dall'orario inizialmente prenotato.

SC_5: Conferma modifica prenotazione

Mario accede al sistema di gestione e consulta la lista delle richieste di modifica pendenti. Individua la richiesta di Alessia, relativa al campo di beach volley per il 23 gennaio, dalle 9:00 alle 10:00.

Seleziona la richiesta e clicca sul pulsante "Conferma" per approvare la modifica. Il sistema aggiorna la prenotazione originale, creando una nuova prenotazione con giorno o orario modificati in base alle preferenze indicate da Alessia.

SC_6: Annullamento di una prenotazione da parte del cliente

Luca accede al sistema di prenotazione tramite il proprio dispositivo e visualizza la prenotazione precedentemente effettuata. Seleziona l'opzione "Annulla prenotazione", avviando la procedura di cancellazione.

Il sistema verifica il rispetto del termine ultimo di 24 ore antecedenti l'orario di inizio previsto per consentire l'annullamento. Una volta confermato, la prenotazione viene eliminata dal sistema, lo stato del campo torna a "Disponibile", e l'operazione viene registrata in tempo reale.

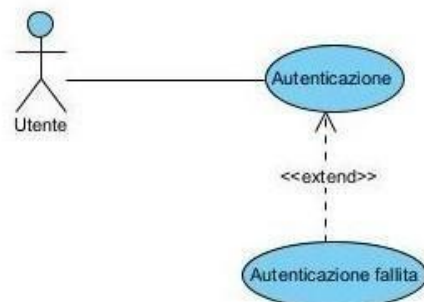
2.4.3. Casi d'uso

UC1: Autenticazione

- **Attore:** Utente, Gestore
- **Entry Condition:** L'utente apre l'app e vede la schermata di login • **Flusso di eventi:**
 1. L'utente/gestore inserisce la propria email e password
 2. L'utente/gestore clicca sul pulsante "Accedi"
 3. Il sistema invia i dati a Firebase per l'autenticazione
 4. Firebase verifica le credenziali
 5. Se le credenziali sono valide, il sistema permette l'accesso e visualizza la home page dell'app
- **Exit Condition:** L'utente/gestore è autenticato e viene reindirizzato alla pagina principale con le funzionalità disponibili.
- **Exception:** Se le credenziali sono errate, il sistema mostra un messaggio di errore e l'utente resta sulla schermata di login

UC1.1: Autenticazione fallita

- **Attore:** Utente/Gestore
- **Entry Condition:** L'utente/gestore ha provato ad autenticarsi ma le credenziali non sono corrette
- **Flusso di eventi:**
 1. Il sistema mostra un messaggio di errore
 2. Il sistema attende un nuovo inserimento dei dati
- **Exit condition:** L'utente/gestore si trova nella pagina di login



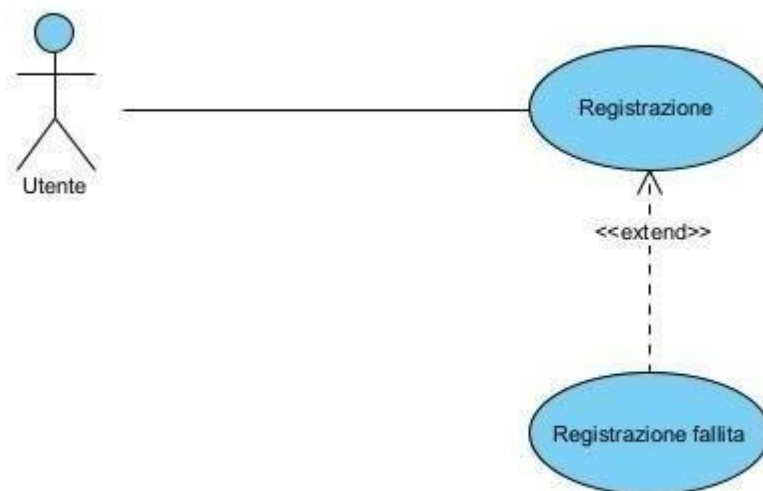
UC2: Registrazione Utente • Attore: Utente

- **Entry Condition:** L'utente si trova nella pagina di Login • **Flusso di eventi:**

1. L'utente seleziona "Registrazione"
 2. Il sistema reindirizza alla pagina di registrazione
 3. L'utente inserisce nome, cognome, numero di telefono, e-mail, password.
 4. Il sistema controlla il formato dei dati inseriti
 5. L'utente invia i dati al sistema
 6. Il sistema reindirizza l'utente alla pagina home
- **Exit Condition:** L'utente è autenticato e si trova nella pagina home
 - **Exception:** Se il sistema rileva un formato errato, mostra un messaggio di errore per ogni formato errato

UC2.1: Registrazione fallita • Attore: Utente

- **Entry Condition:** L'utente ha provato a registrarsi ma il formato inserito non è corretto • **Flusso di eventi:**
 1. Il sistema mostra un messaggio di errore per ogni campo con formato errato
 2. Il sistema attende un nuovo inserimento dei dati
- **Exit Condition:** L'utente si trova nella pagina di registrazione



UC3: Effettuare una prenotazione • Attore: Utente

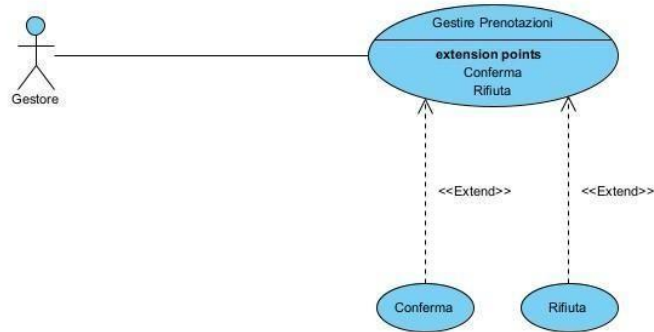
- **Entry Condition:** L'utente ha selezionato il campo e vuole fare una prenotazione per il giorno desiderato
- **Flusso di eventi:**
 1. L'utente seleziona il campo che vuole

2. Il sistema mostra un calendario con tutti gli orari disponibili
 3. L'utente sceglie l'orario e il giorno che vuole
 4. L'utente conferma la prenotazione
- **Exit Condition:**
 1. La richiesta di prenotazione è stata inviata con successo.
 - **Exception:**
 1. In caso di errore di rete o problemi con Firebase, il sistema mostra un messaggio di errore e informa l'utente che la richiesta non è stata completata.
 2. In caso di una prenotazione doppia da parte di due utenti che prenotano allo stesso tempo, verrà creata soltanto la prenotazione che riesce ad arrivare al server in modo più rapido.



UC4: Gestire Prenotazioni

- **Attori:** Gestore
- **Entry Condition:** il gestore ha selezionato una prenotazione in attesa.
- **Flusso di eventi:**
 1. Il gestore clicca sull'icona delle prenotazioni
 2. Il gestore visualizza tutte le prenotazioni con stato "in attesa"
 3. Il gestore sceglie di **confermare** o **rifiutare** le prenotazioni
 4. Il sistema aggiorna lo stato delle prenotazioni
- **Exit Condition:** 1. La prenotazione è stata gestita correttamente (confermata o rifiutata)
- **Exception:**
 1. Errori di rete, impossibilità di aggiornare la prenotazione.



UC5: Modificare una prenotazione (lato utente) • Attore: Utente

- **Entry Condition:** L'utente ha una prenotazione esistente che può essere modificata.
- **Flusso di eventi:**
 1. L'utente seleziona la propria prenotazione esistente dalla lista di prenotazioni attive.
 2. L'utente sceglie l'opzione "Modifica prenotazione".
 3. L'utente modifica uno o più dettagli della prenotazione (ad esempio data, ora, ecc.).
 4. L'utente invia la richiesta di modifica cliccando su "Invia modifica".
 5. Il sistema conferma che la richiesta è stata inoltrata correttamente al gestore per l'approvazione.
- **Exit Condition:**

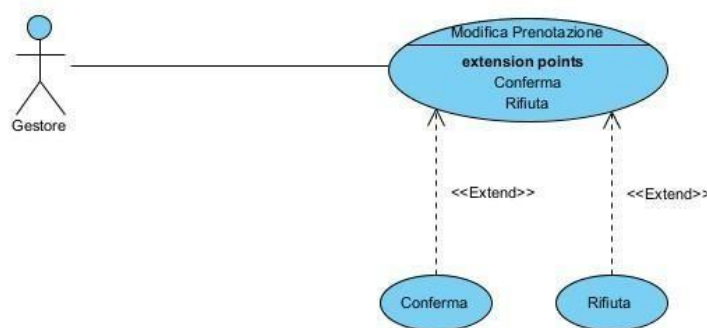
La richiesta di modifica è stata inviata con successo al sistema.
- **Exception:**
 1. Errori di rete, impossibilità di aggiornare la prenotazione.



UC6: Gestire richieste di modifica prenotazioni (lato gestore) • Attore: Gestore

- **Entry Condition:** Il gestore ha ricevuto una richiesta di modifica di una prenotazione.
- **Flusso di eventi:**

1. Il gestore visualizza le richieste di modifica delle prenotazioni.
 2. Il gestore valuta la richiesta di modifica.
 3. Il gestore accetta o rifiuta la modifica. • **Exit Condition:**
 1. Il sistema aggiorna il nuovo stato della prenotazione.
- **Excpetion:**
 1. Errori di rete, impossibilità di aggiornare la prenotazione.



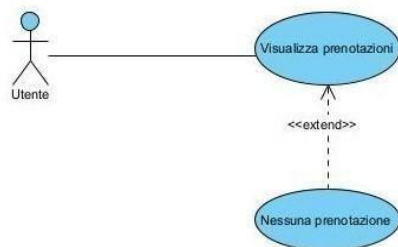
UC5: Cancella prenotazione • Attori: Utente

- **Entry Condition:** L'utente ha una prenotazione esistente alla quale non può più andare • **Flusso di eventi:**
 1. L'utente seleziona la propria prenotazione dalla lista di prenotazioni attive
 2. L'utente sceglie l'opzione "Elimina prenotazione"
 3. Il sistema chiede conferma dell'eliminazione
 4. L'utente conferma l'eliminazione
 5. Il sistema elimina immediatamente la prenotazione
- **Exit Condition:**
 - La prenotazione viene eliminata con successo e lo slot ritornerà disponibile



UC6: Visualizzazione delle prenotazioni • Attori: Utente

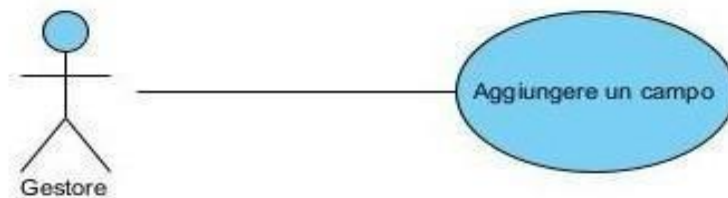
- **Entry Condition:** L'utente è autenticato e si trova nella sezione "Le mie prenotazioni" • **Flusso di eventi:**
 1. L'utente accede alla sezione "Le mie prenotazioni" dal menu dell'app
 2. Il sistema richiede a Firebase l'elenco delle prenotazioni effettuate dall'utente
 3. Il sistema mostra all'utente la lista delle sue prenotazioni.
- **Exit Condition:** L'utente visualizza correttamente la lista delle prenotazioni
- **Exception:** Se non ci sono prenotazioni, il sistema mostra il messaggio "Non hai effettuato nessuna prenotazione"



UC7: Aggiunta di un campo da parte del gestore

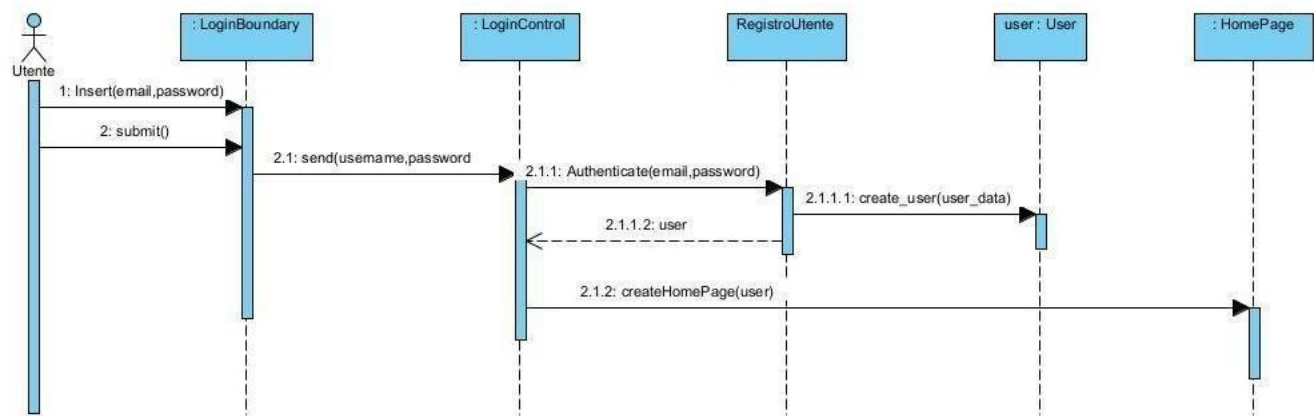
- **Attori:** Gestore
- **Entry Condition:** Il gestore si trova nella schermata di gestione dei campi
- **Flusso di eventi:**

1. Il gestore seleziona l'opzione "Aggiungi campo" della schermata di gestione
 2. Il sistema mostra una schermata di inserimento con i campi necessari
 3. Il gestore inserisce tutte le informazioni richieste e verifica che siano corrette
 4. Il gestore conferma l'aggiunta cliccando su "Aggiungi"
 5. Il sistema notifica al gestore che l'aggiunta è stata completata con successo.
- **Exit Condition:** Il nuovo campo è stato aggiunto correttamente al sistema ed è disponibile per le prenotazioni degli utenti
 - **Exception:** Se il gestore lascia dei campi obbligatori vuoti o inserisce dati in un formato non corretto, il sistema mostra un messaggio di errore e richiede la correzione dei dati prima di poter proseguire

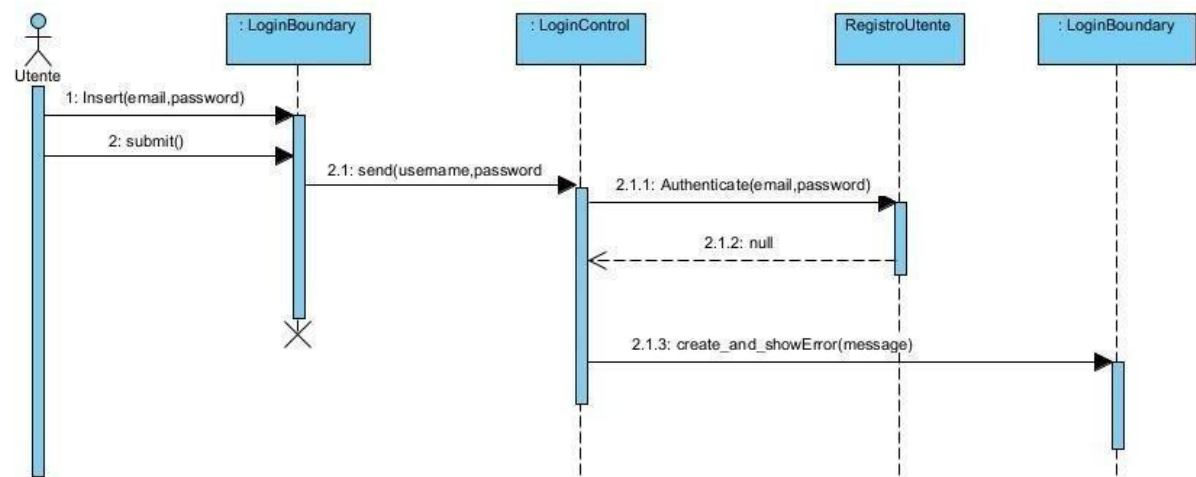


2.4.4. Sequence Diagram

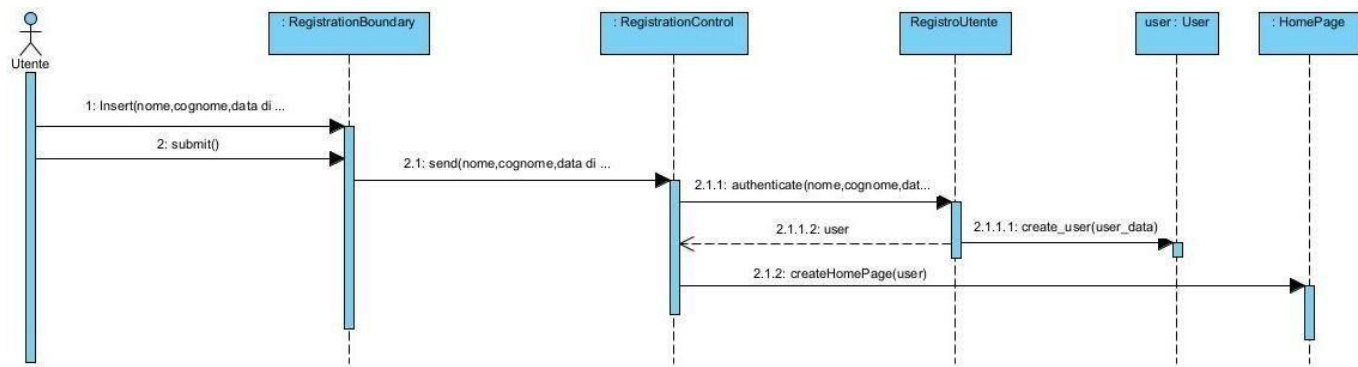
SD_1 Autenticazione



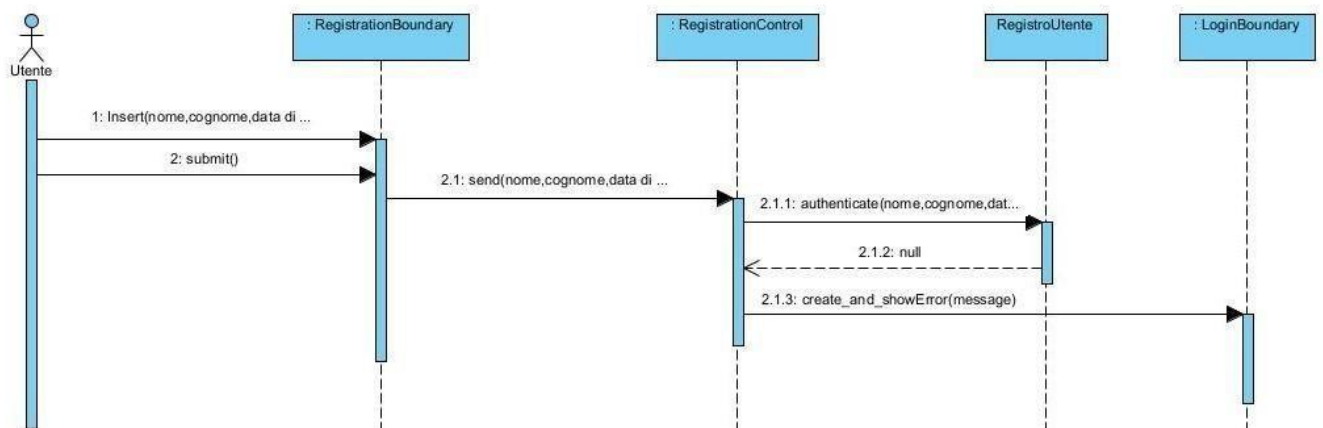
SD_1_1 Autenticazione Fallita



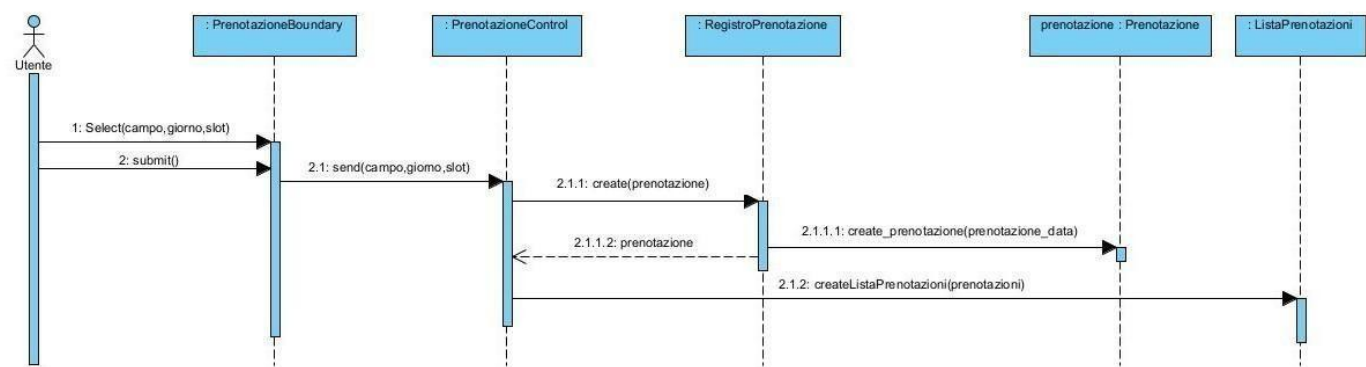
SD_2 Registrazione



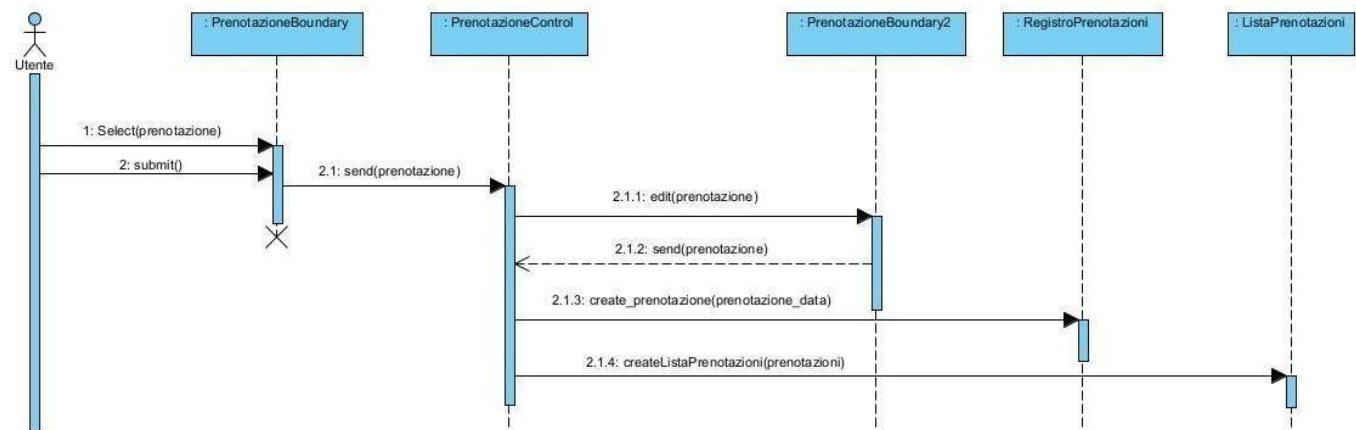
SD_2_1 Registrazione Fallita



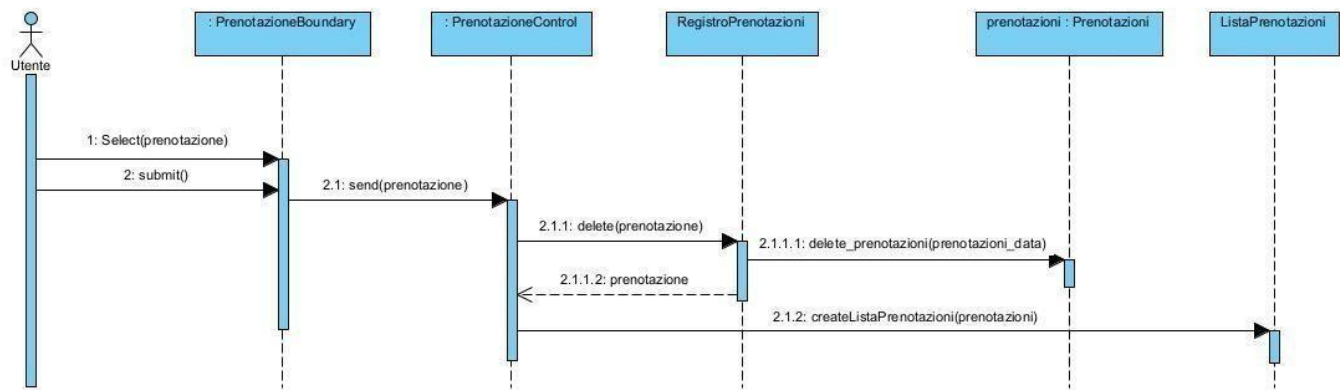
SD_3 Prenotazione Campo



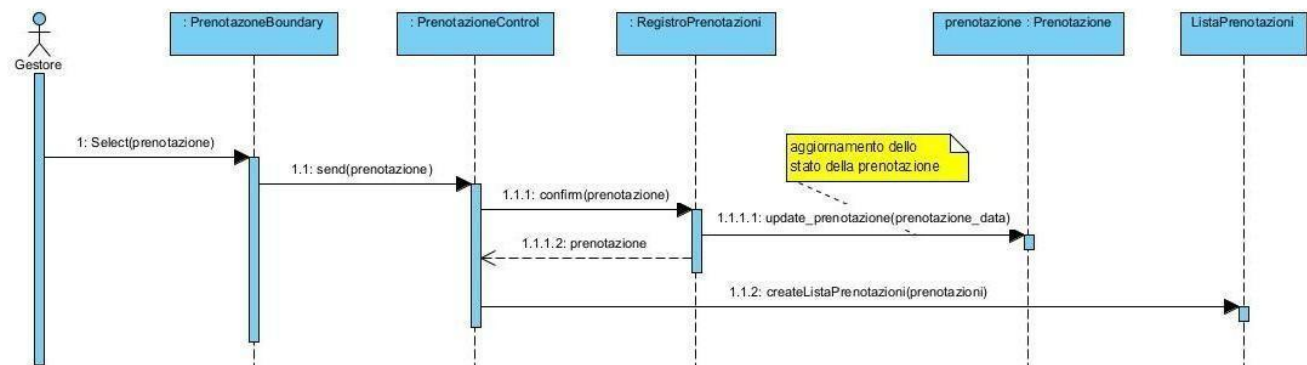
SD_3_1 Modifica Prenotazione



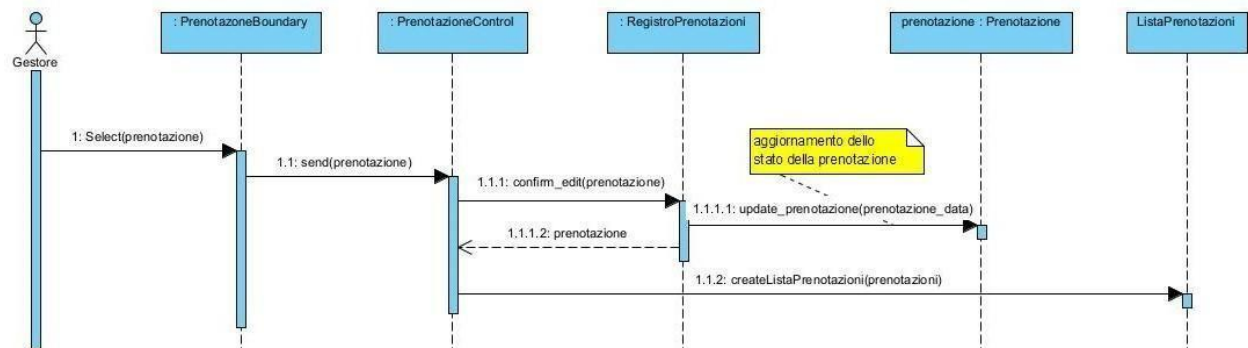
SD_3_2 Cancellazione Prenotazione



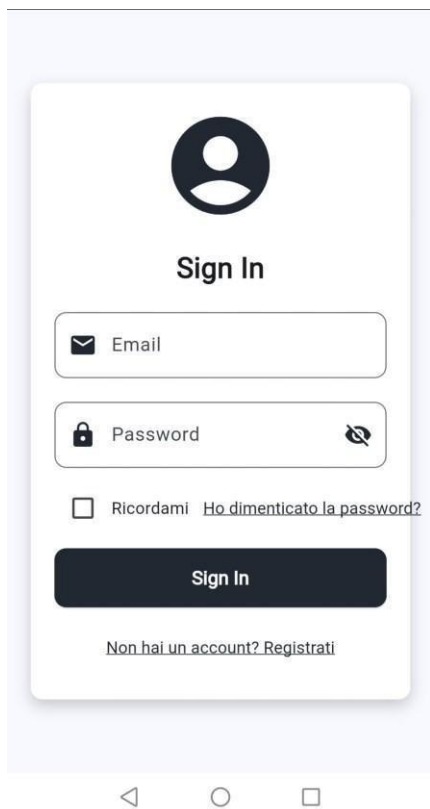
SD_4 Conferma Prenotazione Gestore



SD_5 Conferma Modifiche Gestore



2.4.5. Activity Diagram



A mobile app screen for signing in. At the top is a dark blue header with a white back arrow and the text "Registrazione". Below the header is a white card with a dark blue user icon at the top. The card has the title "Sign In" in bold. It contains two input fields: "Email" with an envelope icon and "Password" with a lock icon and a toggle eye icon. Below the password field is a checkbox labeled "Ricordami" and a link "Ho dimenticato la password?". At the bottom of the card is a dark blue button labeled "Sign In". Below the card is a link "Non hai un account? Registrati". The screen has a light blue background and a white Android navigation bar at the bottom.

Sign In

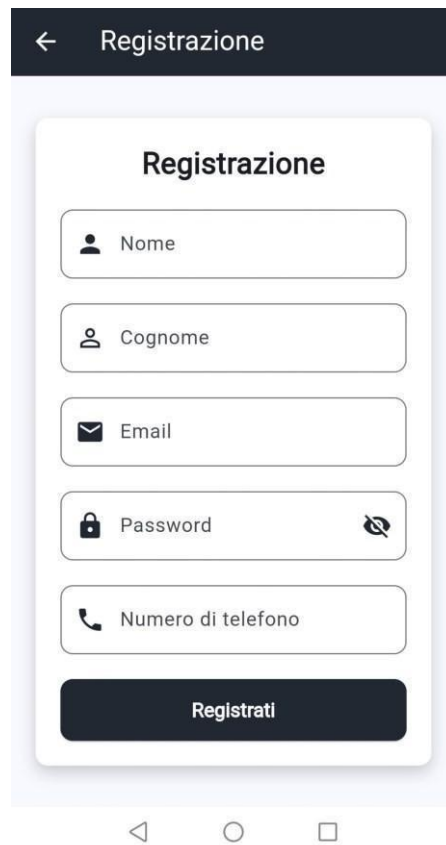
Email

Password

☐ Ricordami [Ho dimenticato la password?](#)

Sign In

[Non hai un account? Registrati](#)



A mobile app screen for registration. At the top is a dark blue header with a white back arrow and the text "Registrazione". Below the header is a white card with the title "Registrazione" in bold. It contains four input fields: "Nome" with a person icon, "Cognome" with a person icon, "Email" with an envelope icon, and "Password" with a lock icon and a toggle eye icon. Below the password field is a field for "Numero di telefono" with a phone icon. At the bottom of the card is a dark blue button labeled "Registrati". The screen has a light blue background and a white Android navigation bar at the bottom.

Registrazione

Nome

Cognome

Email

Password

Numero di telefono

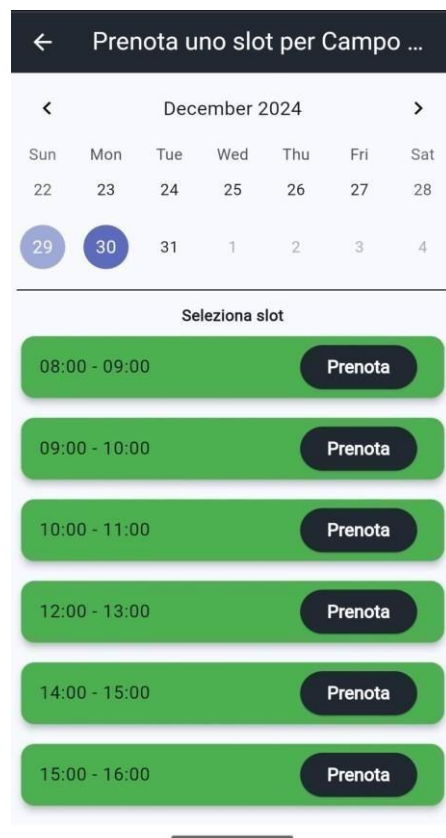
Registrati

MU_3 Home Utente

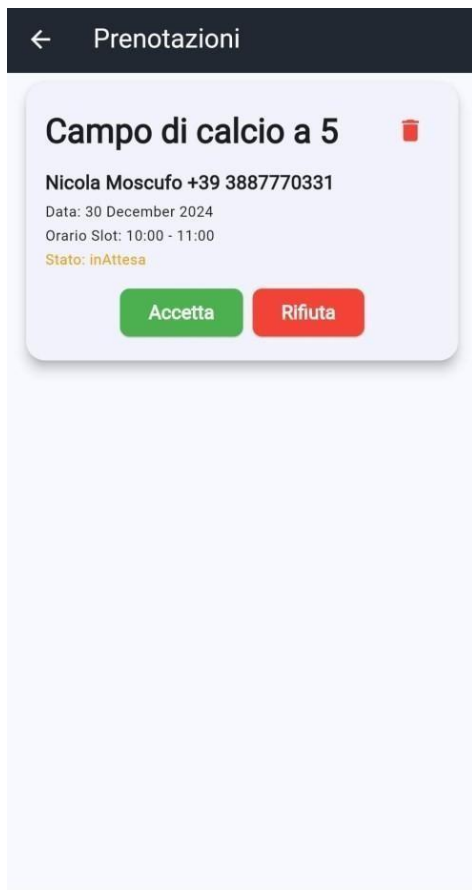
MU_4 Calendario Slot Utente



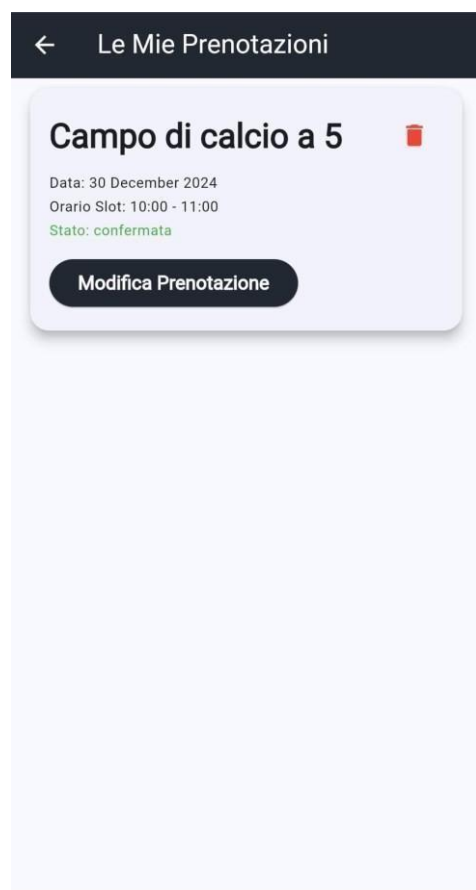
MU_5 Richiesta Prenotazione Effettuata



MU_6 Prenotazione Confermata



MU_7 Scelta Nuovo Slot per Modifica



MU_8 Richiesta Modifica Effettuata

←

Modifica Prenotazione

<

December 2024

>

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
| 22 | 23 | 24 | 25 | 26 | 27 | 28 |
| 29 | 30 | 31 | 1 | 2 | 3 | 4 |

Seleziona nuovo slot

08:00 - 09:00

Modifica

09:00 - 10:00

Modifica

10:00 - 11:00

Modifica

12:00 - 13:00

Non disponibile

14:00 - 15:00

Modifica

15:00 - 16:00

Modifica

←

Le Mie Prenotazioni

Campo di calcio a 5

Data: 30 December 2024

Orario Slot: 12:00 - 13:00

Stato: richiestaModifica

Campo di calcio a 5

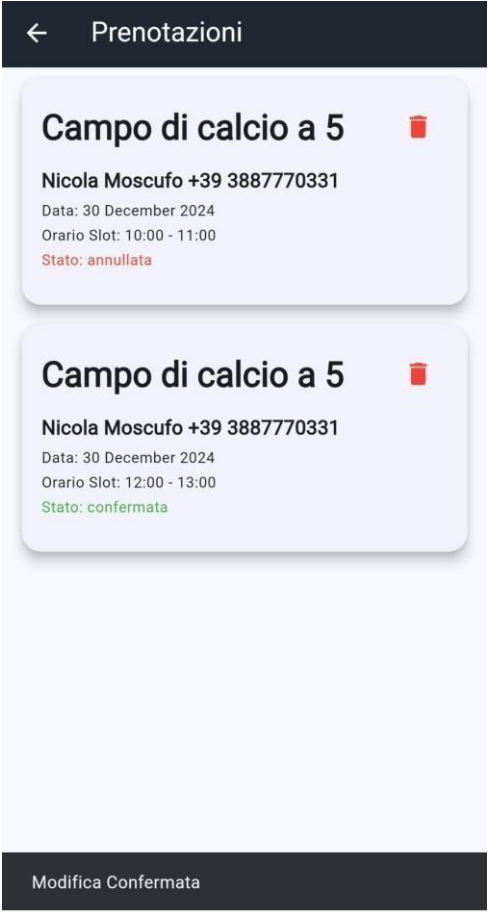
Data: 30 December 2024

Orario Slot: 10:00 - 11:00

Stato: annullata

MU_9 Richiesta Modifica Accettata

MU_10 Cancellazione Prenotazione



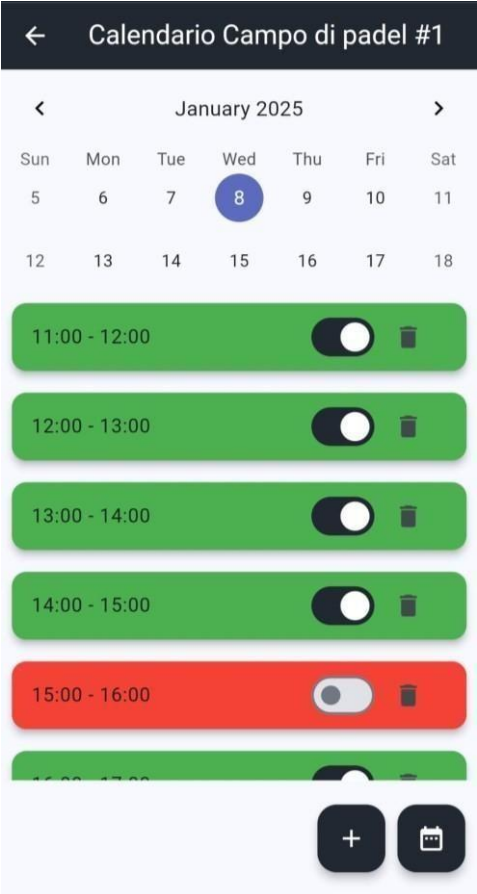
MU_11 Home Gestore



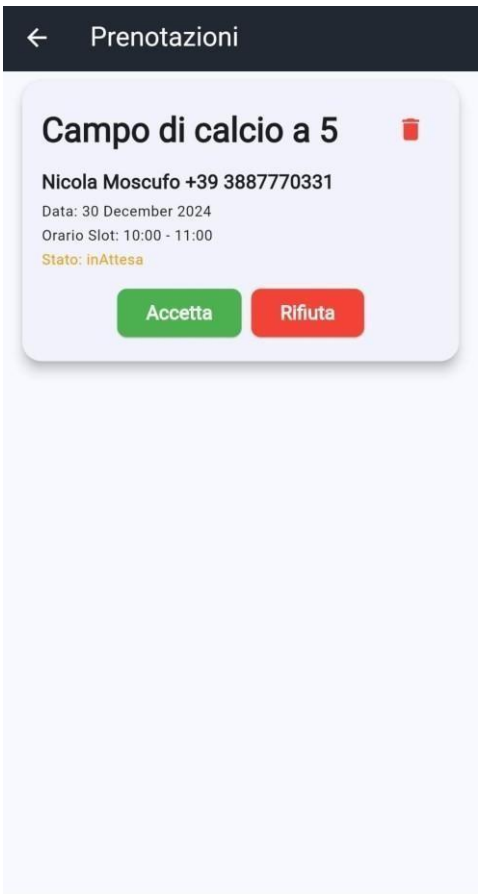
MU_12 Aggiunta Campo



MU_12 Aggiunta Slot Orari



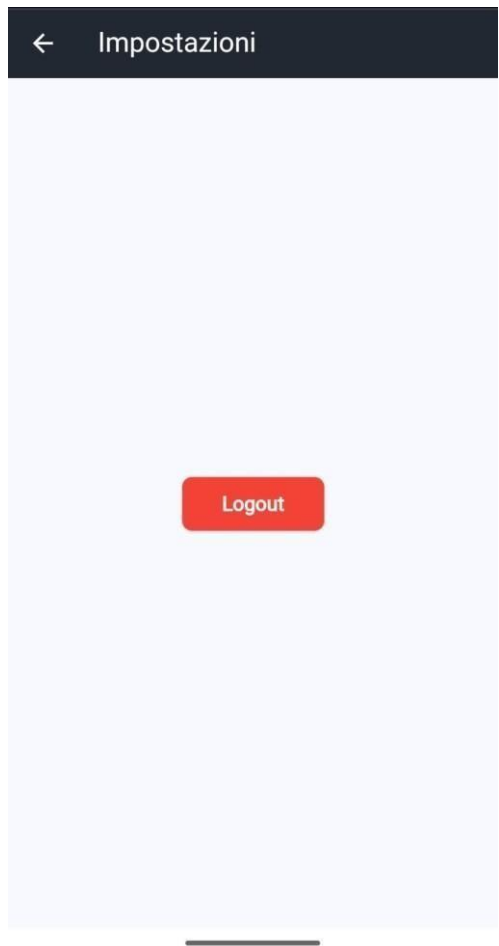
MU_13 Richiesta Prenotazione



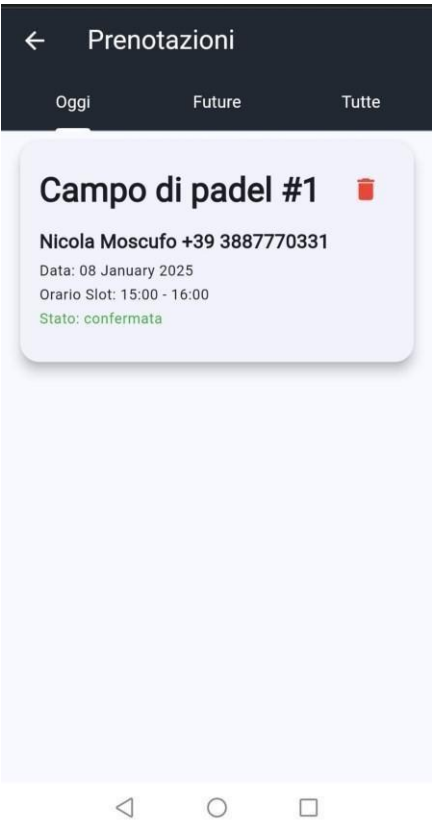
MU_14 Richiesta Modifica



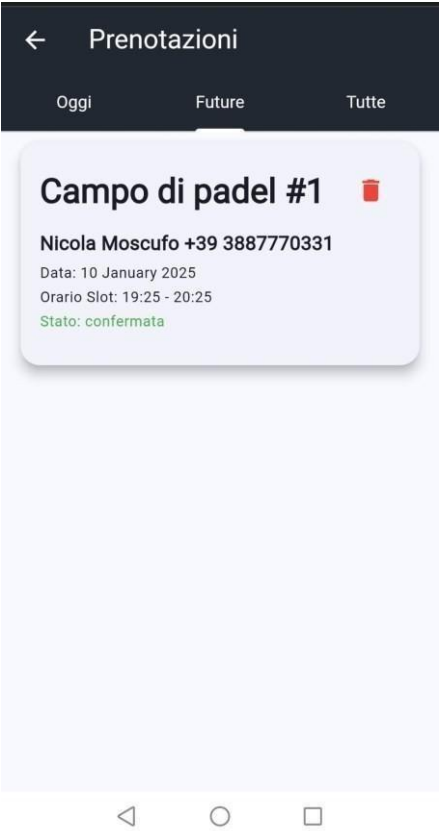
MU_15 Logout



MU_16 Lista prenotazioni giorno corrente Gestore



MU_17 Lista prenotazioni future Gestore



MU_18 Lista di tutte le prenotazioni Gestore

