

**MatchDay**  
**Requirements Analysis Document**  
**Versione 2.1**



**MatchDay**

Progetto: MatchDay	Versione: 2.1
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

<b>Scritto da:</b>	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

# 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 .....	4
1.5.	Panoramica .....	5
2.	SISTEMA PROPOSTO .....	5
2.1.	Panoramica .....	5
2.2.	Requisiti funzionali .....	5
2.3.	Requisiti non funzionali .....	6
2.3.1.	Usabilità .....	6
2.3.2.	Affidabilità .....	6
2.3.3.	Prestazioni .....	6
2.3.4.	Sostenibilità .....	7
2.3.5.	Implementazione .....	7
2.3.6.	Interfaccia .....	7
2.4.	Modelli di sistema .....	7
2.4.1.	Class Diagram .....	7
2.4.2.	Scenari .....	7
2.4.3.	Casi d'uso .....	9
2.4.4.	Sequence Diagram .....	16
2.4.5.	Activity Diagram .....	20
2.4.6.	Mock-ups .....	21

# 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 inviare una richiesta di cancellazione della prenotazione
- Visualizzare lo storico delle prenotazioni

Gestore:

- Aggiunta di campi sportivi
- Modifica di campi sportivi
- Eliminazione dei campi sportivi
- Confermare prenotazioni
- Modificare prenotazioni
- Eliminare 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	Gli amministratori possono aggiungere, modificare e rimuovere campi e gestire le prenotazioni
FR_6	Media	Il sistema invia notifiche per confermare le prenotazioni e informare su eventuali modifiche.
FR_7	Media	Admin 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

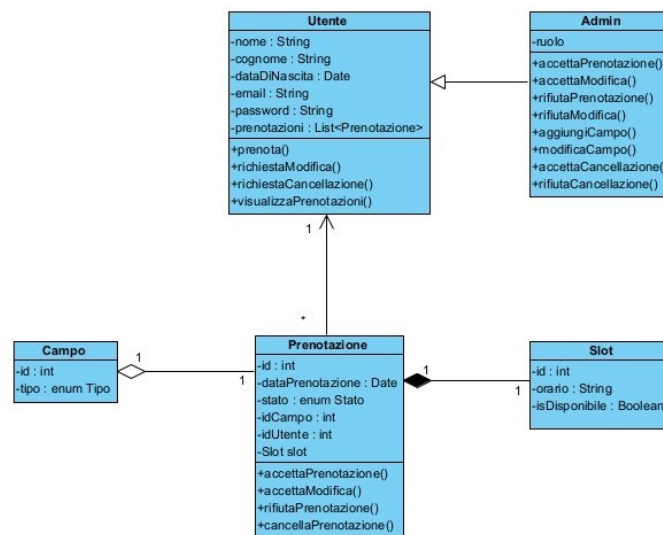
L'implementazione del sistema deve seguire un'architettura basata su Firebase, per garantire la sicurezza dei dati con crittografia. Le funzionalità del sistema devono essere progettate per supportare accessi concorrenti attraverso il Firebase Realtime Database, garantendo scalabilità e prestazioni elevate.

### 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: Prenotazione da parte di un cliente

Con l'app, Marco accede al sistema dal suo smartphone, seleziona il campo da calcio e visualizza gli orari disponibili in tempo reale. Prenota un campo specifico, ricevendo subito una notifica con l'orario e la durata della sua prenotazione. Non ci sono errori, e l'esperienza di prenotazione è fluida e affidabile.

### **SC\_2: Modifica di una prenotazione**

Laura accede rapidamente alla sua prenotazione e sceglie l'opzione per modificarla. Può visualizzare gli orari alternativi disponibili per quel giorno e selezionare uno slot diverso. Riceve una notifica di conferma con la nuova prenotazione, tutto in pochi minuti, senza dover fare una seconda chiamata.

### **SC\_3: Gestione delle prenotazioni da parte di un amministratore**

Con la dashboard amministrativa, Mario può visualizzare tutte le prenotazioni in un unico calendario digitale. Può aggiungere, modificare o cancellare prenotazioni in pochi clic, e vedere la disponibilità giornaliera e settimanale in tempo reale. La gestione del centro sportivo diventa molto più semplice e organizzata.

### **SC\_4: Annullamento di una prenotazione da parte del cliente**

Con l'app, Luca può trovare la sua prenotazione e annullarla con pochi tocchi. L'annullamento è registrato immediatamente, e Luca riceve una notifica che conferma l'operazione, senza il bisogno di fare una telefonata.

### **SC\_5: Accesso in tempo reale alla disponibilità**

Attraverso l'app, Sara può verificare immediatamente la disponibilità del campo da tennis e prenotare uno slot libero. L'intera operazione è completata in pochi minuti, e Sara può organizzare l'attività con i suoi amici senza alcun dubbio sulla disponibilità del campo.

### **SC\_6: Gestione centralizzata di più campi**

Con il sistema centralizzato, il gestore può vedere tutte le prenotazioni per ogni tipo di campo in un'unica schermata. Può monitorare la disponibilità in tempo reale e assegnare i campi in modo ottimale, riducendo gli errori e risparmiando tempo.

### **SC\_7: Prenotazione rapida durante una lezione**

Giulia è una studentessa universitaria che vuole prenotare un campo da calcio per il pomeriggio. Tuttavia, si trova in aula e non può fare una telefonata senza disturbare la lezione. Normalmente, dovrebbe aspettare la fine della lezione e sperare che ci sia ancora un campo disponibile. Senza dover parlare, Giulia apre l'app in modalità silenziosa durante la lezione e seleziona il campo di calcio. In pochi secondi, sceglie l'orario, prenota il campo e riceve una conferma immediata. Può così rimanere concentrata sulla lezione e assicurarsi di avere il campo disponibile per il pomeriggio.

### **SC\_8: Prenotazioni senza stress per utenti con difficoltà di comunicazione**

Luca soffre di ansia sociale e fare telefonate lo mette a disagio. Vuole prenotare un campo da tennis per giocare con un amico, ma il pensiero di dover parlare con qualcuno al telefono lo preoccupa, al punto che spesso rinuncia a fare attività che gli piacciono.

Con l'app, Luca può prenotare senza interagire con nessuno. Visualizza gli orari disponibili, seleziona un campo e riceve una notifica di conferma. In questo modo, Luca può organizzare le sue attività in modo indipendente, senza dover affrontare situazioni che gli causano stress.



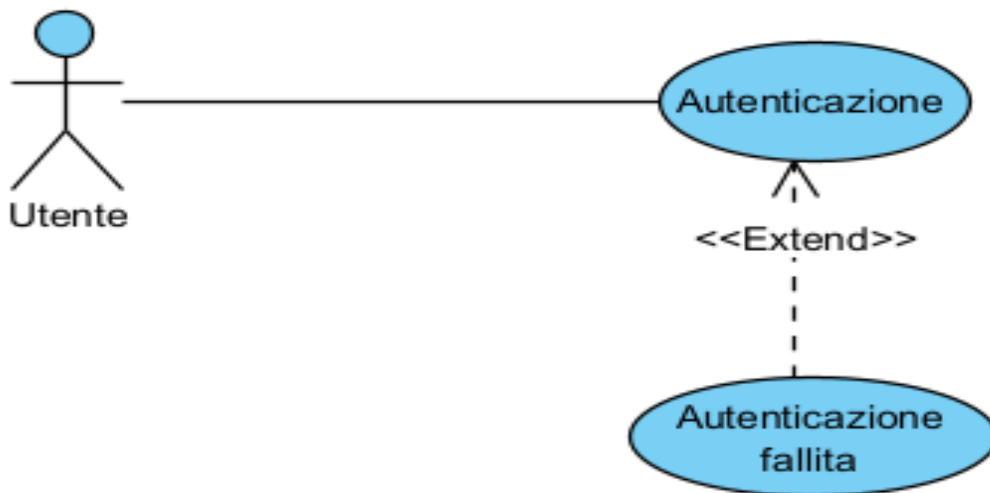
### 2.4.3. Casi d'uso

#### 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

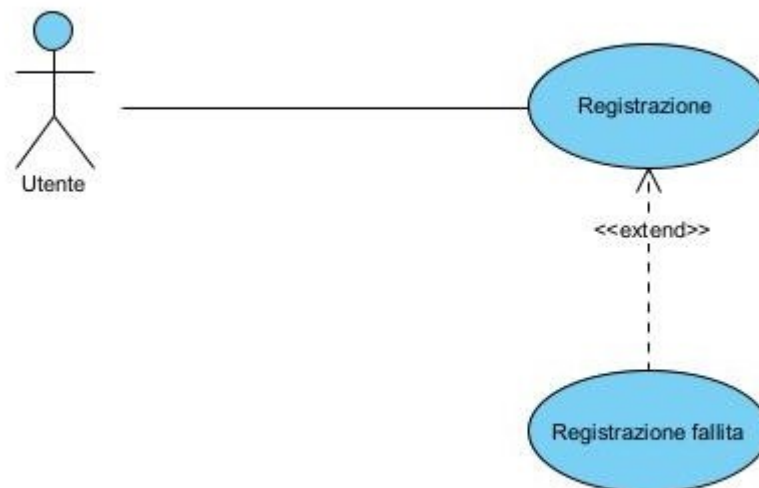


## UC2: Registrazione Utente

- 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

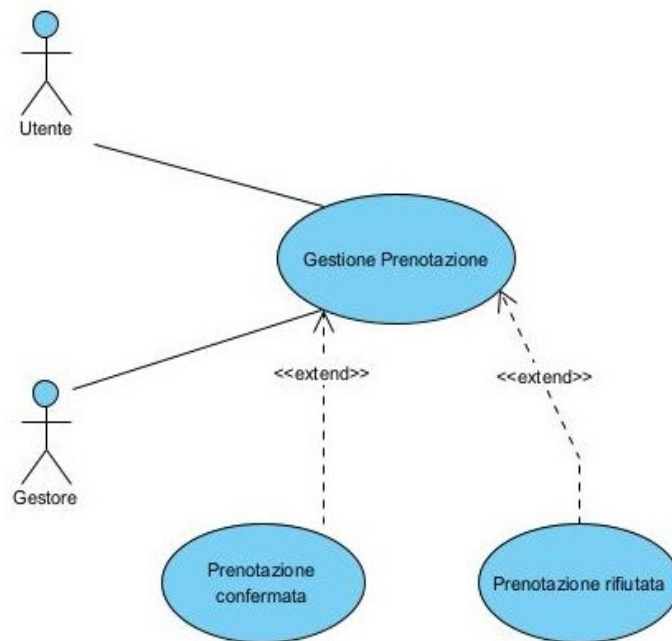
### UC2.1: Registrazione fallita

- 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



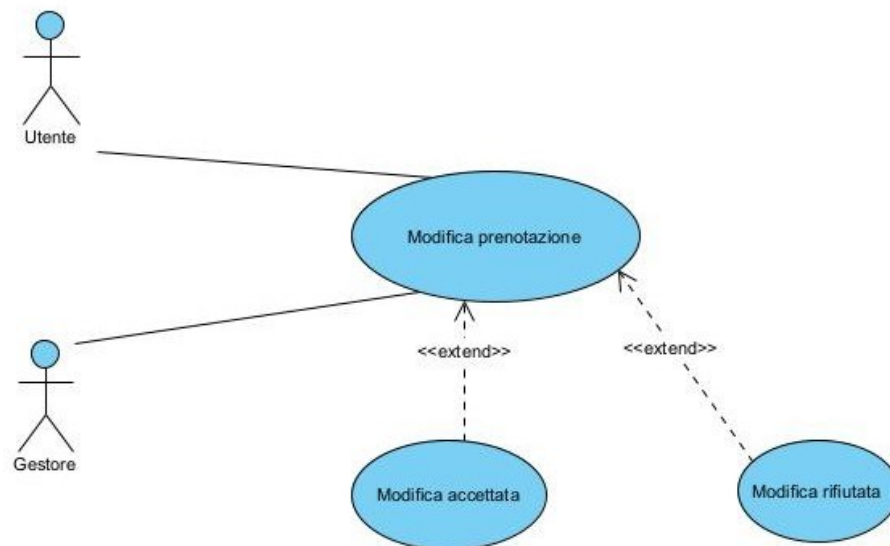
### UC3: Effettuare una prenotazione

- Attore: Utente, Gestore
- 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
  5. Il sistema invia la richiesta al gestore
  6. Il gestore deciderà se confermare o meno la prenotazione
- Exit Condition:
  - Se la prenotazione è confermata dal gestore, l'utente riceve una notifica di conferma.
  - Se la prenotazione viene rifiutata, l'utente riceve una notifica e può fare una nuova prenotazione
- Exception: 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.



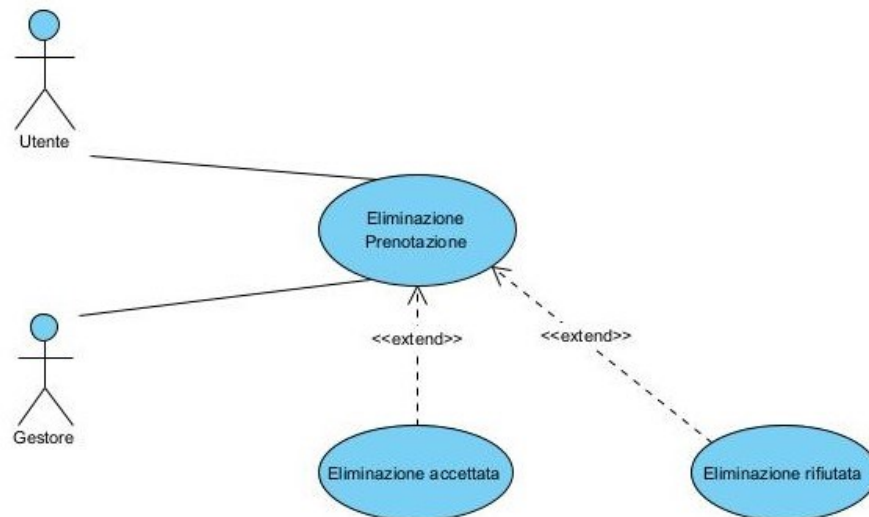
#### UC4: Modificare una prenotazione

- Attori: Utente, Gestore
- Entry Condition: L'utente ha una prenotazione già 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 e invia la richiesta di modifica cliccando su "Invia modifica".
  4. Il sistema notifica che la richiesta di modifica è stata inviata al gestore per l'approvazione.
  5. Il gestore riceve la richiesta di modifica della prenotazione.
  6. Il gestore valuta la motivazione legata alla modifica e sceglie di accettarla o rifiutarla
- Exit Condition:
  - Se il gestore accetta, la prenotazione verrà aggiornata con le modifiche richieste
  - Se il gestore rifiuta, la prenotazione non verrà modificata



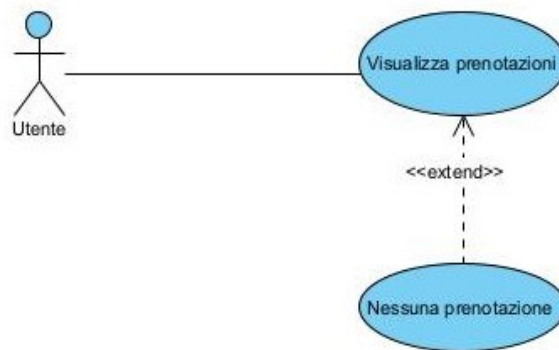
### UC5: Eliminazione di una prenotazione

- Attori: Utente, Gestore
- 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. L'utente aggiunge la motivazione dell'eliminazione
  4. Il sistema chiede conferma dell'eliminazione
  5. L'utente conferma l'eliminazione
  6. Il sistema invia la richiesta di eliminazione al gestore per l'approvazione
  7. Il gestore riceve la richiesta di eliminazione e valuta la motivazione dell'utente.
  8. Il gestore decide se accettare o meno l'eliminazione
- Exit Condition:
  - Se il gestore accetta la richiesta, la prenotazione verrà correttamente eliminata
  - Se il gestore rifiuta la richiesta, la prenotazione non verrà eliminata ma rimarrà attiva



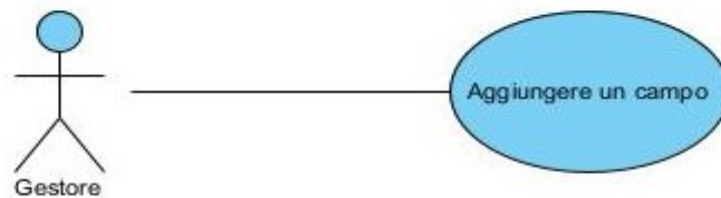
## UC6: Visualizzazione delle prenotazioni

- 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"



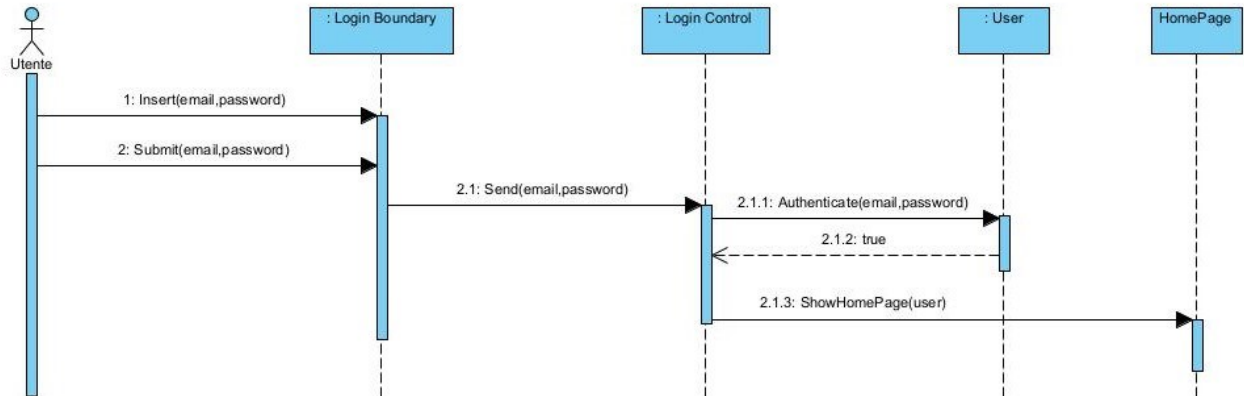
### UC7: Aggiunta di un campo da parte del gestore

- 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

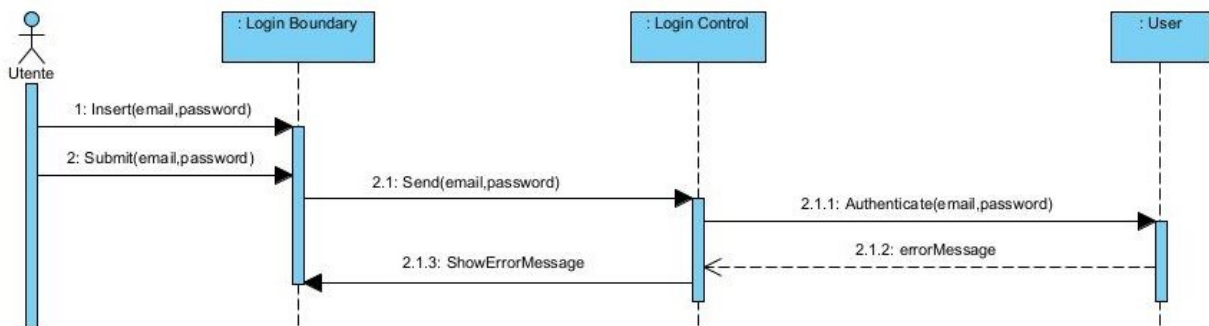


## 2.4.4. Sequence Diagram

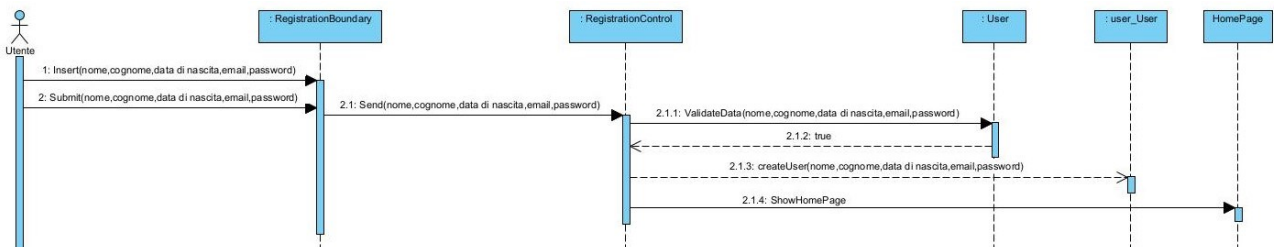
### SD\_1 Autenticazione



### SD\_1\_1 Autenticazione Fallita

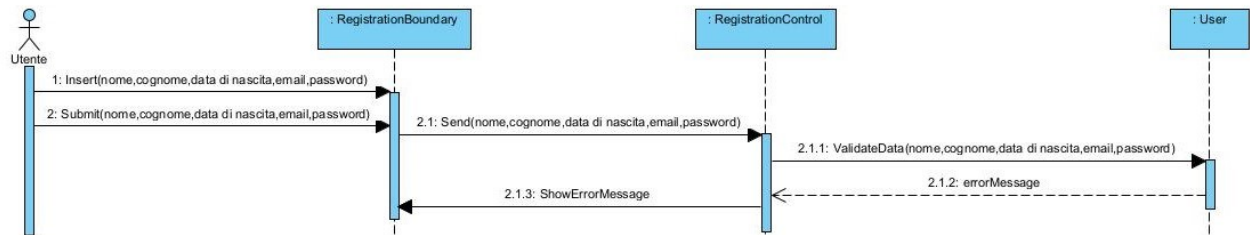


### SD\_2 Registrazione

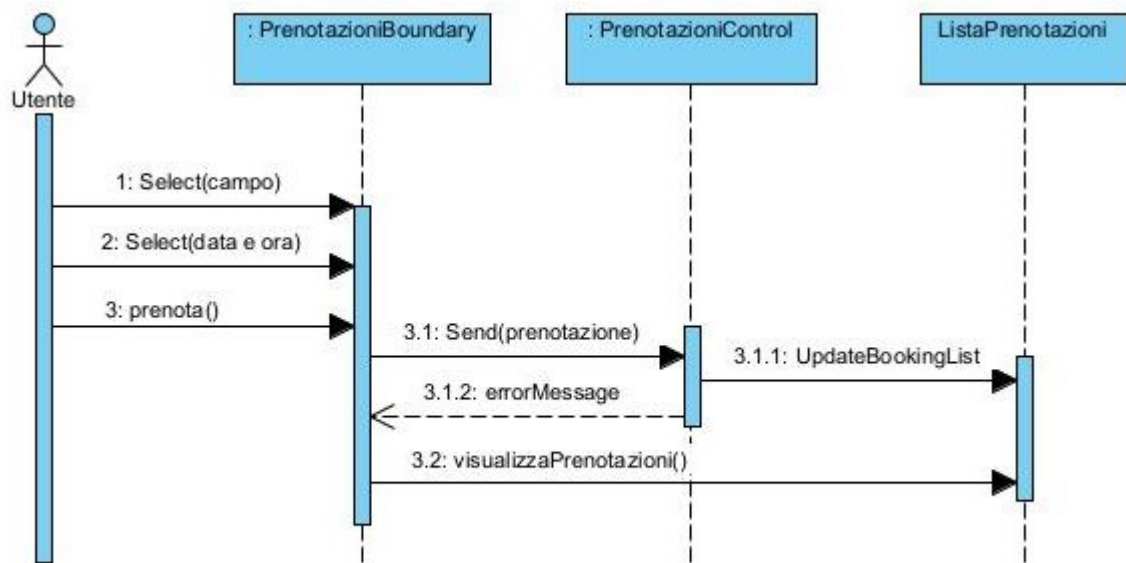




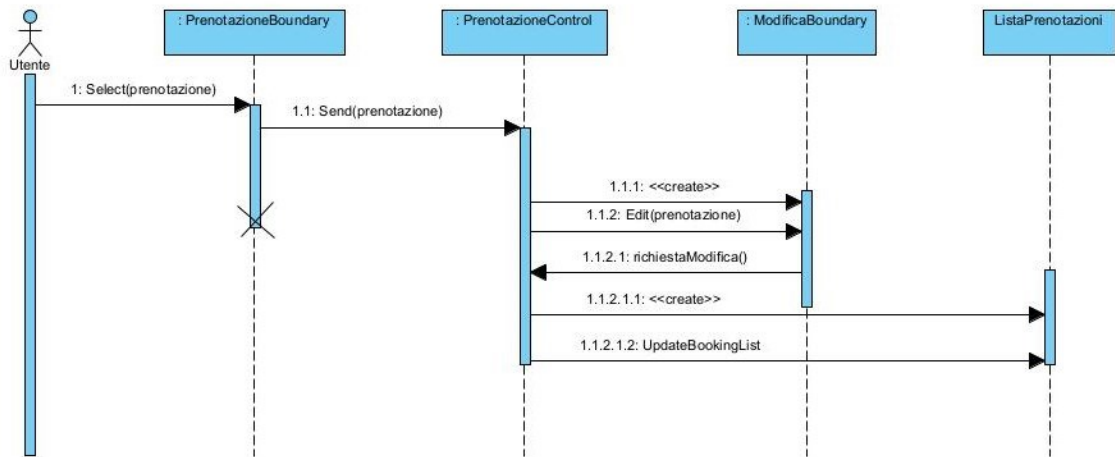
## SD\_2\_1 Registrazione Fallita



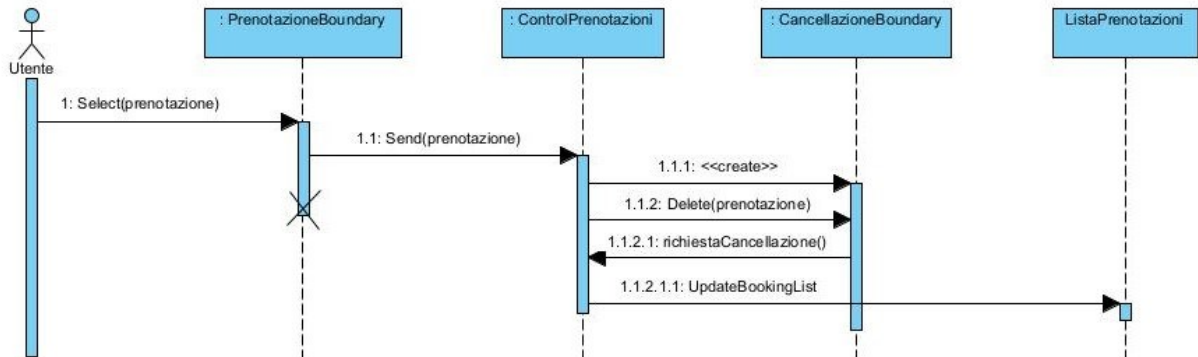
## SD\_3 Prenotazione Campo



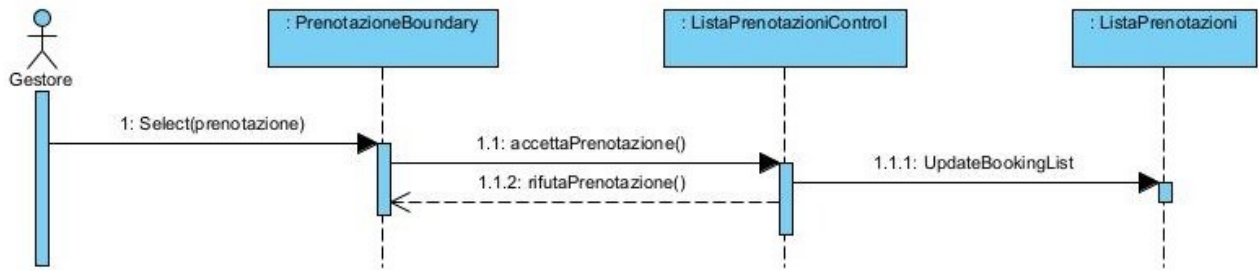
### SD\_3\_1 Modifica Prenotazione



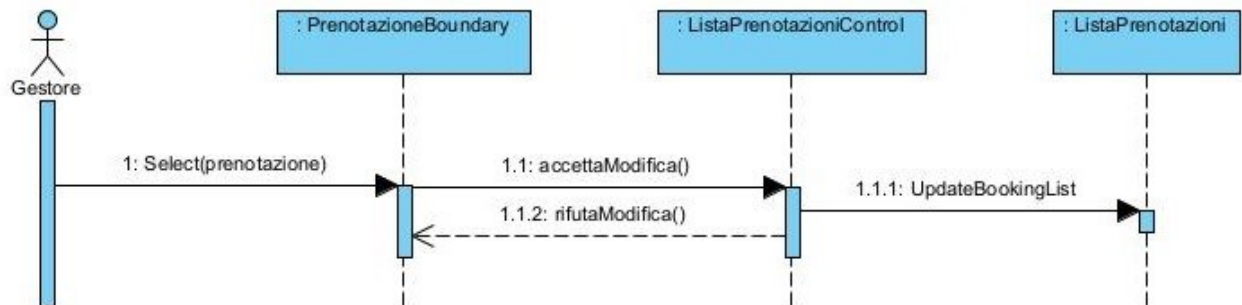
### SD\_3\_2 Cancellazione Prenotazione



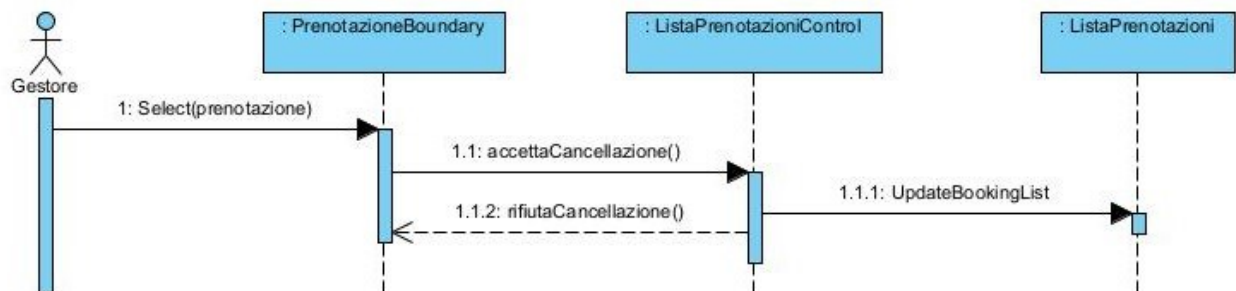
## SD\_4 Conferma Prenotazione Gestore



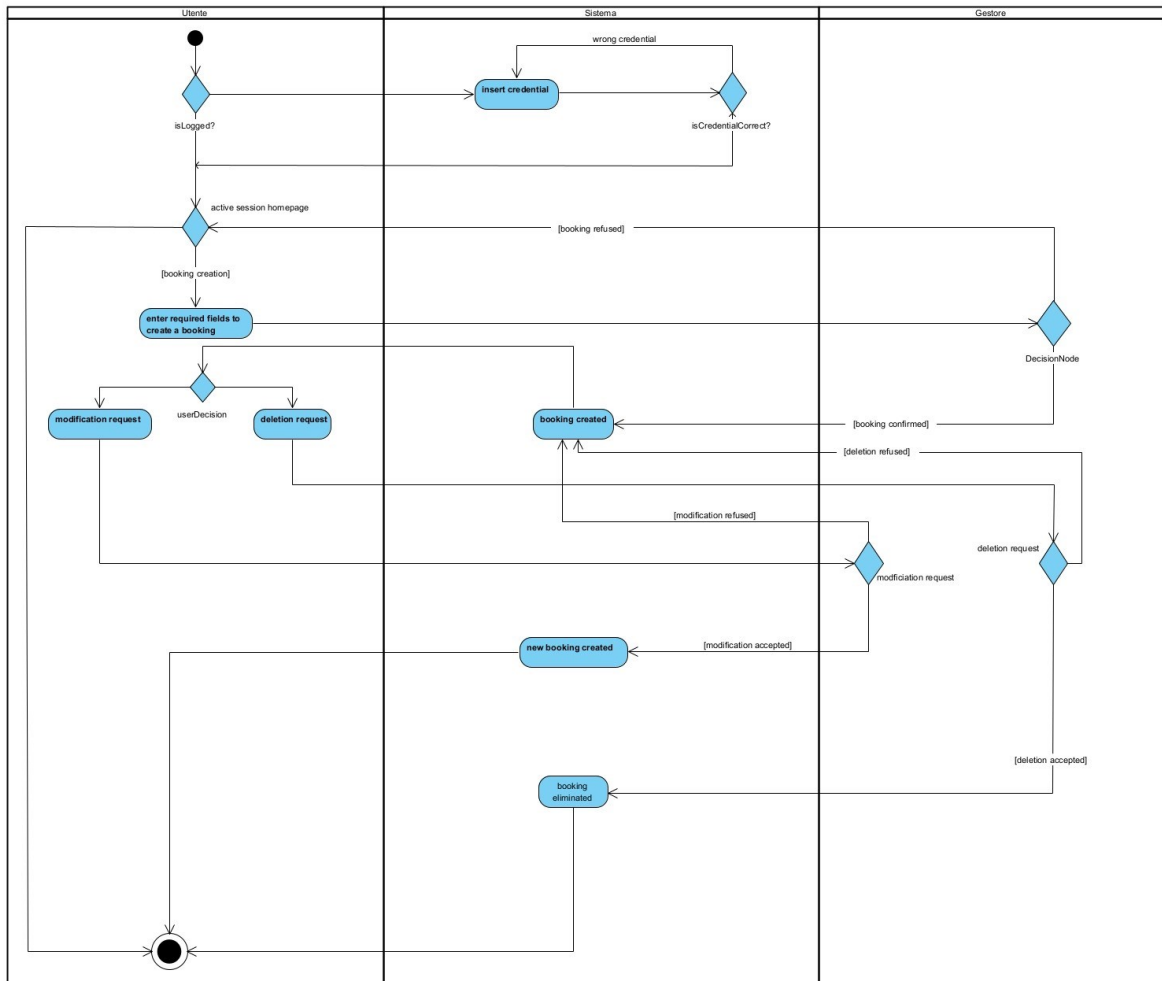
## SD\_5 Conferma Modifiche Gestore



## SD\_6 Conferma Cancellazione Prenotazione Gestore

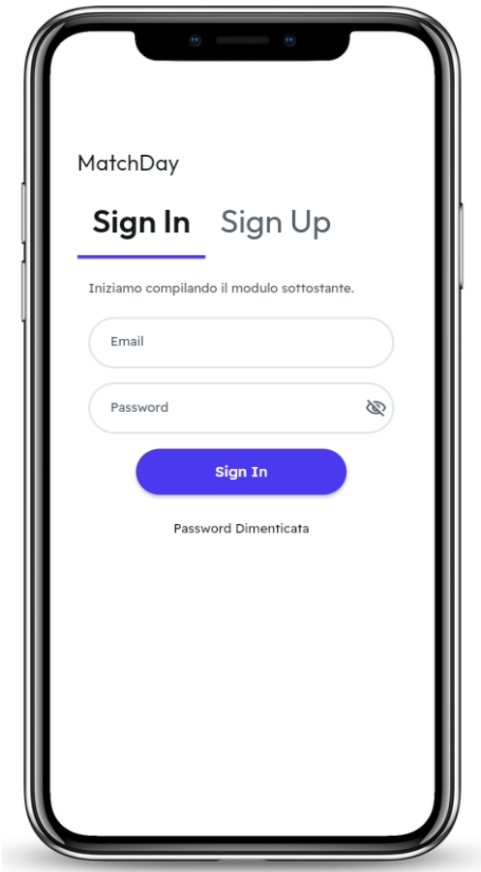


## 2.4.5. Activity Diagram

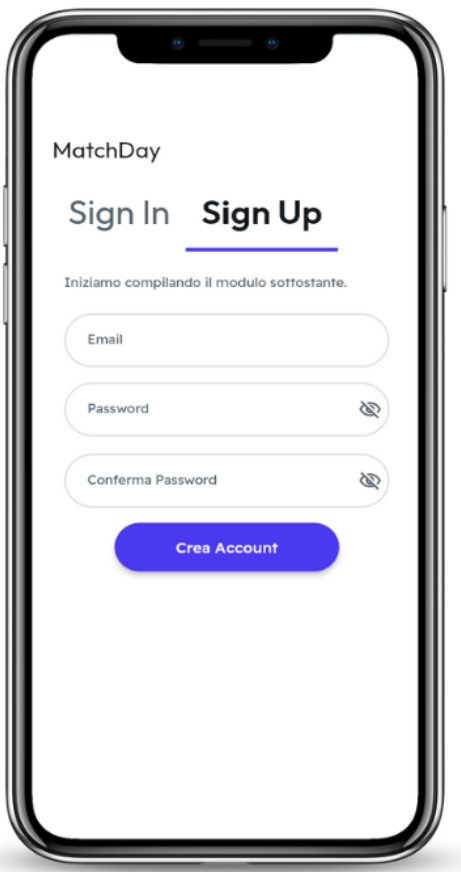


2.4.6. Mock-ups

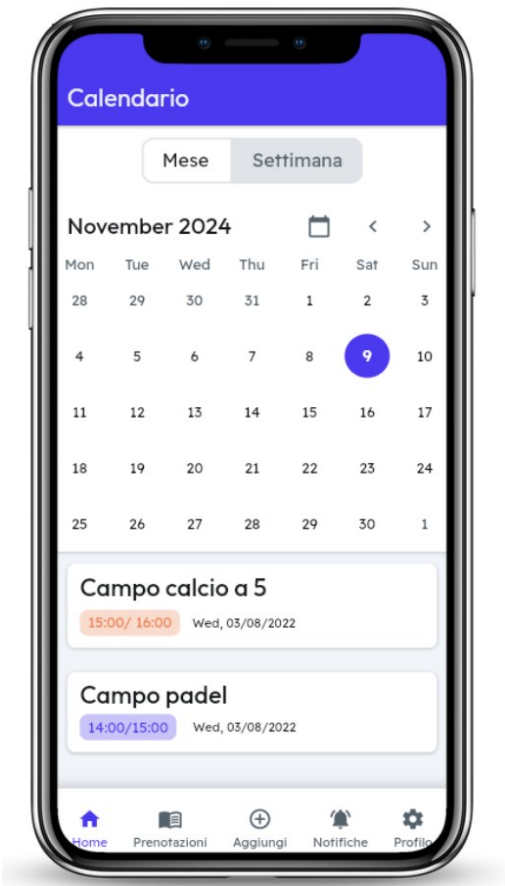
MU\_1 Login



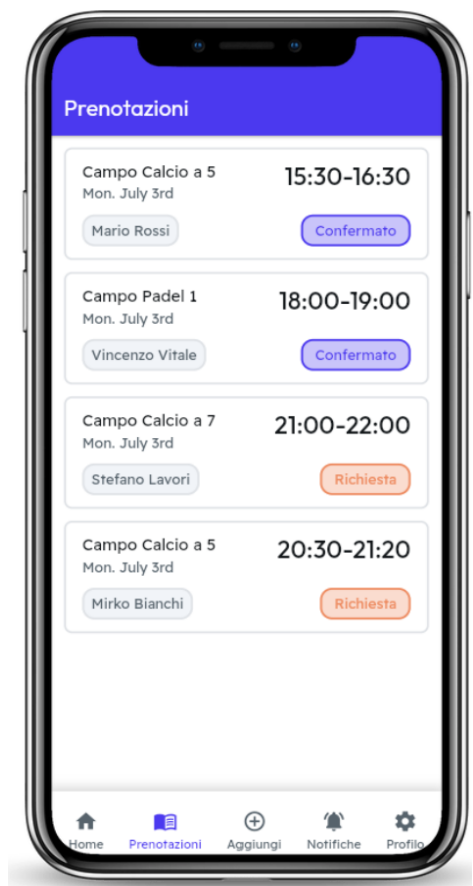
MU\_2 Registrazione



MU\_3 Home



MU\_4 Lista prenotazioni Admin



MU\_5 Aggiungi campo

