

**Università degli Studi di Salerno**

**Corso di Ingegneria del Software**

**MatchDay**  
**System Design Document**  
**Versione 0.2**



**MatchDay**

Data: 18/11/2024

Progetto: MatchDay	Versione: 0.2
Documento: System Design Document	Data: 18/11/2024

**Coordinatore del progetto:**

Nome	Matricola

**Partecipanti:**

Nome	Matricola
Vincenzo Vitale	0512113542
Nicola Moscufo	0512114886
Francesco Moscufo	0512115027

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

**Revision History**

Data	Versione	Descrizione	Autore
12/12/2024	0.2	Modifica della decomposizione in sottosistemi	Vincenzo Vitale

## 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
1.3.1.	<i>Criteri di performance</i> .....	4
1.3.2.	<i>Criteri di affidabilità</i> .....	4
1.3.3.	<i>Criteri di manutenzione</i> .....	4
1.4.	Definizioni, Acronimi, Abbreviazioni .....	4
2.	SISTEMA PROPOSTO .....	5
2.1.	Panoramica .....	5
2.2.	Decomposizione in sottosistemi.....	5
2.3.	Mapping Hardware/Software .....	6
2.3.1.	<i>Deployment Diagram</i> .....	7
2.4.	Gestione dei dati persistenti .....	7
2.5.	Controllo degli accessi e sicurezza .....	7
2.6.	Controllo flusso globale del software .....	8
2.7.	Boundary Condition .....	8

# 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 di un centro sportivo, 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
- Visualizzazione dello storico delle prenotazioni

## ***1.3.Obiettivi e criteri di successo del sistema***

### ***1.3.1. Criteri di performance***

- Tempo di risposta: Il sistema deve rispondere in poco tempo in modo tale da poter garantire una navigazione fluida all'utente.
- Memoria: La memoria fornita dal database deve essere scalabile in modo tale da poter gestire sia l'accesso di più utenti che la registrazione da parte di nuovi utenti.

### ***1.3.2. Criteri di affidabilità***

- Disponibilità: Il sito web dovrà essere disponibile 24/7.
- Sicurezza e privacy: L'applicazione richiede il login (tramite email e password), nel caso si tratti di un utente già registrato, altrimenti richiede i dati dell'utente per registrarsi.
- Robustezza: Input non validi verranno individuati e verrà mostrato all'utente un messaggio di errore.

### ***1.3.3. Criteri di manutenzione***

- Portabilità: La portabilità sarà garantita in quanto l'applicazione è utilizzabile sia su web che su dispositivi mobili.

## ***1.4. Definizioni, Acronimi, Abbreviazioni***

SDD: System Design Document

DB: Database

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

## 2. SISTEMA PROPOSTO

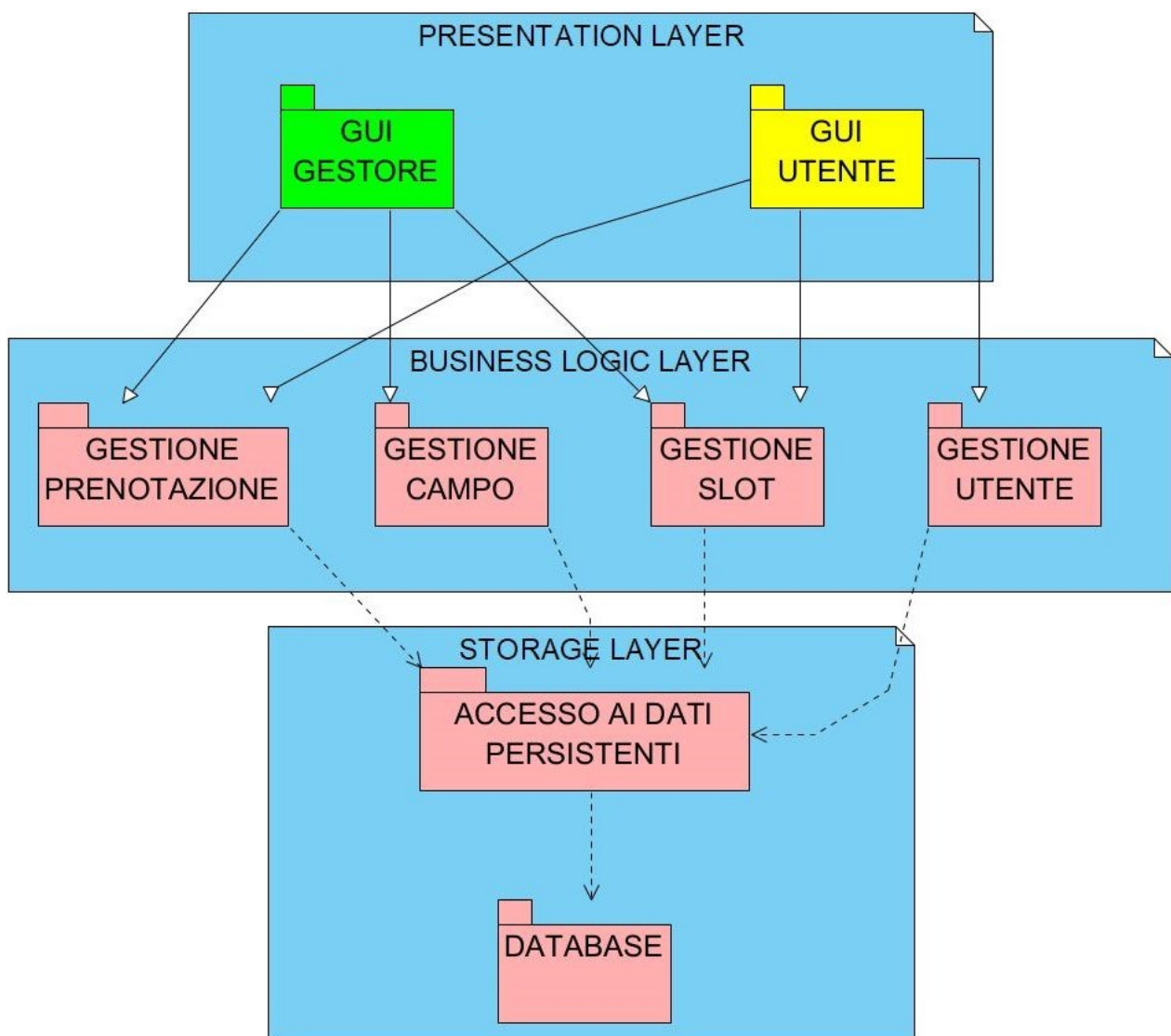
### 2.1. Panoramica

Il sistema proposto consente agli utenti di prenotare campi sportivi in modo semplice e veloce, sia da dispositivi mobile che da web. All'avvio, l'app richiede agli utenti di effettuare il login o di registrarsi, assicurando così un accesso sicuro e personalizzato. Dopo l'autenticazione, gli utenti possono visualizzare la disponibilità dei campi, selezionare orari e date, e completare le prenotazioni con pochi click.

Il sistema si basa su tre layer:

- Presentation Layer
- Business Logic Layer
- Data Access Layer

### 2.2. Decomposizione in sottosistemi



Il sistema si compone di tre layer:

- **Il Presentation Layer** si occupa di presentare le informazioni ai diversi utenti e consente agli stessi di interagire con il sistema
- **Il Business Logic Layer** definisce la logica applicativa
- **Data Layer** si occupa della gestione dei dati persistenti

Il Presentation Layer presenta 2 sottosistemi:

- **Gui Gestore:** rappresenta l'insieme delle pagine che il gestore può visualizzare per gestire le prenotazioni, aggiungere campi e aggiungere o modificare slot per le prenotazioni
- **Gui Utente:** rappresenta l'insieme delle pagine che un utente correttamente registrato può visualizzare per poi procedere alla prenotazione

Il Business Logic Layer è diviso in 3 sottosistemi:

- **Gestione Prenotazione:** fornisce le funzionalità per effettuare, modificare o eliminare una prenotazione e visualizzare le prenotazioni effettuate, questa funzionalità riguarda l'utente mentre il gestore ha la possibilità di confermare le prenotazioni, modifiche o cancellazioni effettuate dall'utente.
- **Gestione Campo:** fornisce le funzionalità per aggiungere, modificare o rimuovere un campo. Questa funzionalità è disponibile solo per il gestore.
- **Gestione Slot:** fornisce al gestore la funzionalità di aggiungere, rimuovere o modificare uno slot prenotabile dall'utente. L'utente prenota lo slot che più preferisce modificandone lo stato.

Il Data Layer è costituito da un unico sottosistema:

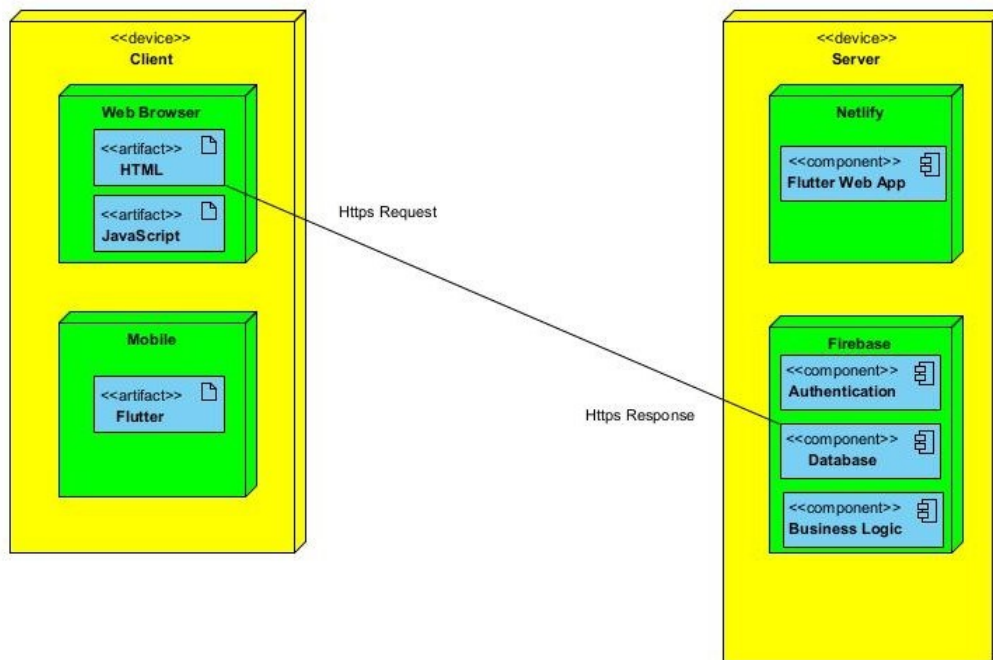
- **Accesso ai dati persistenti:** si occupa di leggere e modificare i dati presenti nel database

### ***2.3.Mapping Hardware/Software***

Per il sistema, basato su un'architettura three-layer, è stata scelta questa configurazione:

- Per il Presentation Layer utilizziamo Flutter
- Il Business Logic Layer si divide in:
  - Lato Client: Gestisce funzioni come la selezione di date e la validazione base degli input
  - Firebase (Server): Esegue la logica di business principale, come il controllo della disponibilità dei campi e la conferma delle prenotazioni.
- Per il Data Access Layer utilizziamo Firebase Firestore
- Come protocollo di comunicazione utilizziamo HTTP, TCP/IP.

### 2.3.1. Deployment Diagram



### 2.4. Gestione dei dati persistenti

Per la gestione della persistenza dei dati e dell'autenticazione, abbiamo adottato una soluzione integrata basata su Firebase. Firebase Authentication si occupa della gestione sicura degli utenti, consentendo l'autenticazione e l'accesso personalizzato alle risorse. Per quanto riguarda la persistenza dei dati, utilizziamo Firebase Firestore, un database NoSQL che organizza le informazioni in raccolte e documenti. Questa combinazione ci permette di garantire sincronizzazione in tempo reale, una gestione semplificata degli accessi e un'applicazione reattiva, capace di scalare per gestire un elevato numero di richieste.

### 2.5. Controllo degli accessi e sicurezza

Attori\Oggetti	Gestione Prenotazioni	Gestione Campo	Gestione Slot
Utente	<ul style="list-style-type: none"> <li>Prenota il campo desiderato</li> <li>Visualizza lo storico delle sue prenotazioni</li> <li>Possibilità di</li> </ul>	<ul style="list-style-type: none"> <li>Seleziona la tipologia di campo che preferisce</li> </ul>	<ul style="list-style-type: none"> <li>Modificando la prenotazione ha la possibilità di selezionare uno slot differente</li> </ul>

	modifica della prenotazione • Possibilità di cancellazione della prenotazione		
Gestore	• Conferma o rifiuta le prenotazioni effettuate dagli utenti • Possibilità di confermare o rifiutare le modifiche alle prenotazioni degli utenti • Possibilità di confermare o rifiutare le richieste di cancellazione della prenotazione da parte degli utenti	• Aggiunge nuovi campi • Modifica campi già esistenti • Elimina campi già esistenti	• Aggiunta di nuovi slot disponibili da prenotare • Rimozione degli slot • Modifica degli slot

## 2.6. Controllo flusso globale del software

Il global control flow del software è event-driven e reattivo, con un'ampia gestione asincrona delle operazioni di rete. Questo significa che l'app risponde agli eventi generati dalle interazioni degli utenti (come login, prenotazioni e logout) e agli aggiornamenti in tempo reale provenienti da Firebase.

## 2.7. Boundary Condition

### Registrazione Obbligatoria:

- L'utente deve completare il processo di registrazione o di login tramite Firebase Authentication per poter accedere all'app.
- L'accesso è possibile solo dopo aver fornito credenziali valide. Se l'utente non è registrato, sarà obbligato a completare la registrazione prima di poter utilizzare l'app.
- La registrazione è vincolata alla connessione a Internet, in quanto richiede la comunicazione con il server Firebase per l'autenticazione dell'utente.

### Funzionalità Solo Online:

- L'app non supporta l'uso offline. Ogni interazione con il database (come la verifica della disponibilità dei campi o la conferma di una prenotazione) richiede una connessione attiva a Internet.
- Se l'utente perde la connessione a Internet durante l'uso dell'app, le funzionalità non saranno disponibili e l'app non sarà in grado di effettuare operazioni come registrazioni, login, lettura



o scrittura di dati nel database.

**Accesso ai Dati e Operazioni:**

- Tutti i dati (come le prenotazioni, la disponibilità dei campi e le informazioni sugli utenti) sono recuperati e gestiti tramite Firebase Firestore, che richiede una connessione stabile a Internet per ogni interazione.
- Le operazioni di prenotazione e gestione dei dati vengono eseguite in tempo reale, con aggiornamenti automatici quando i dati nel database vengono modificati.