

MatchDay
Test Plan
Versione 0.1



MatchDay

Data: 06/12/2024

Progetto: MatchDay	Versione: 0.1
Documento: Test Plan	Data: 06/12/2024

Coordinatore del progetto:

Nome	Matricola

Partecipanti:

Nome	Matricola
Vincenzo Vitale	0512113542
Nicola Moscufo	0512114886
Francesco Moscufo	0512115027

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

Revision History

Data	Versione	Descrizione	Autore

Sommario

1.	INTRODUZIONE	4
2.	PANORAMICA DEL SISTEMA	4
3.	FUNZIONALITA' DA TESTARE / DA NON TESTARE	4
4.	CRITERI PASS/FALL.....	4
5.	APPROCCIO.....	5
6.	STRUMENTI USATI PER IL TESTING	5

1. INTRODUZIONE

Questo documento ha l'obiettivo di specificare la pianificazione dell'attività di testing per l'app MatchDay.

2. PANORAMICA DEL SISTEMA

L'architettura è di tipo three-layer, i quali sono:

- Presentation Layer
- Business Data Layer
- Data Layer

Come riportato nell'SDD, il sistema è diviso nei seguenti sottosistemi:

- Gestione Prenotazione
- Gestione Campo
- Gestione Slot
- Accesso ai dati persistenti

3. FUNZIONALITA' DA TESTARE / DA NON TESTARE

Verranno testate le seguenti funzionalità per ogni sottosistema:

- Gestione Prenotazione:
 - Prenotazione
 - Modifica Prenotazione
 - Cancellazione Prenotazione
- Gestione Campo:
 - Aggiunta Campo
 - Rimozione Campo
- Gestione Slot:
 - Aggiunta Slot
 - Rimozione Slot
 - Cambio di stato dello slot
- Gestione Utente:
 - Registrazione Utente
 - Login Utente

4. CRITERI PASS/FALL

Lo scopo del testing è quello di dimostrare la presenza di faults nel sistema. Un test sarà considerato di successo se, dato un input del sistema, l'output osservato sarà diverso dall'output atteso, sarà quindi identificata una failure e verrà corretta. Un test sarà fallito quando l'output osservato corrisponde con l'output atteso.

5. APPROCCIO

L'approccio è diviso in tre fasi che si susseguiranno nel seguente ordine:

- Test di unità per le singole parti del sistema
- Test di integrazione
- Test di sistema per testare tutte le funzionalità implementate nel sistema

6. STRUMENTI USATI PER IL TESTING

- Flutter Test: per realizzare il testing si unità e quello di integrazione
- Mockito: è una libreria Java utilizzata per creare oggetti "mock" durante il testing, in particolare nei test unitari. Un oggetto mock è una versione simulata di un oggetto reale che permette di isolare e testare specifiche unità di codice senza dipendere dal comportamento di altri componenti del sistema.