

**MatchDay**  
**Test Plan**  
**Versione 0.1**



**MatchDay**

Data: 06/12/2024

Documento: Test Plan	Data: 06/12/2024
----------------------	------------------

**Coordinatore del progetto:**

Nome	Matricola

**Partecipanti:**

Nome	Matricola
Vincenzo Vitale	0512113542
Nicola Moscufo	0512114886
Francesco Moscufo	0512115027

<b>Scritto da:</b>	Vincenzo Vitale, Nicola Moscufo, Francesco Moscufo
--------------------	--

**Revision History**

Data	Versione	Descrizione	Autore
		Ingegneria del Software	Pagina 2 di 5

## Sommario

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

### 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:

#### 1. Gestione Prenotazione:

- Rifiuto prenotazione
- Richiesta modifica prenotazione
- Richiesta prenotazione

#### 2. Gestione slot:

- Aggiunta slot
- Rimozione slot
- Cambio di stato dello slot

#### 3. 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

Il Testing sarà condotto utilizzando tecniche di Testing Black Box, Partizionamento in Classi di Equivalenza e Analisi dei valori limite.

## 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.

## 7. TEST CASES

### Functionality test

Non tutte le funzionalità richiedono test approfonditi, in quanto alcune rappresentano semplici interazioni con il sistema e non necessitano di molteplici casi d'uso. Ad esempio, funzionalità come la cancellazione di una prenotazione si limitano ad eseguire un'azione diretta senza particolari condizioni o complessità, e quindi vengono testate con un singolo caso d'uso per verificarne il comportamento atteso.

Altre funzionalità non vengono testate esplicitamente perché il sistema è progettato per prevenire azioni non valide. Per esempio: Impossibilità di cliccare su uno slot non disponibile: il sistema filtra automaticamente gli slot e consente di selezionare solo quelli associati al campo e alla data scelti. Di conseguenza, non è possibile che l'utente scelga uno slot già occupato, rendendo superfluo un test per questa situazione.

### LOGIN

PARAMETERS	Email, password
OBJECTS OF THE ENVIRONMENT	Database
CATEGORIES	CHOICES
Username presente nel database	E1: email presente nel database E2: email non presente nel database
Password presente nel database	P1: password associata all'email nel database P2: password non associata all'username nel database

### TEST CASES

COMBINATION	ORACLE
E1-P1	Utente verrà autenticato correttamente e reindirizzato alla homepage
E2-P1	Visualizzazione errore "Errore durante il login"
E1-P2	Visualizzazione errore "Errore durante il login"

## REGISTRAZIONE

<b>PARAMETERS</b>	Nome, cognome, email, password, telefono
<b>OBJECTS OF THE ENVIRONMENT</b>	Database
<b>CATEGORIES</b>	<b>CHOICES</b>
Formato Nome	N1: nome rispetta il formato del nome N2: nome vuoto
Formato Cognome	C1: cognome rispetta il formato di cognome C2: cognome non rispetta il formato di cognome
Formato Email	E1: l'email corretta E2: l'email non valida
Formato Password	P1: la password rispetta il formato prestabilito P2: la password non rispetta il formato
Formato Telefono (solo numeri)	T1: telefono rispetta il formato T2: telefono non rispetta il formato

## TEST CASES

<b>COMBINATION</b>	<b>ORACLE</b>
N1-C1-E1-P1-T1	Utente registrato correttamente
N2-C1-E1-P1-T1	Visualizzazione errore "Inserisci il tuo nome"
N1-C2-E1-P1-T1	Visualizzazione errore "Inserisci il tuo cognome"
N1-C1-E2-P1-T1	Visualizzazione errore "Inserisci una email valida"
N1-C1-E1-P2-T1	Visualizzazione errore "Password non valida"
N1-C1-E1-P1-T2	Visualizzazione errore "Numero di telefono" non valido

## Unit testing

### Slot

<b>METHOD NAME</b>	addSlots
<b>PARAMETERS</b>	campold: String, selectedDay: DateTime
<b>CATEGORIES</b>	<b>CHOISES</b>
Valore campold	C1: campold != null C2: campold = null
Valore selectedDay	D1: date != null D2: date = null
Valore slot	S1: slot != null S2: slot = null

<b>COMBINATIONS</b>	<b>ORACLE</b>
C1-D1-S1	Corretto: slot aggiunto con successo
C2-D1-S1	Error: "impossibile trovare il campo"

C1-D2-S1	Error: "giorno non disponibile"
C1-D1-S2	Error: "impossibile creare slot"

## Slot

<b>METHOD NAME</b>	fetchSlots
<b>PARAMETERS</b>	campold: String, selectedDay: DateTime
<b>CATEGORIES</b>	<b>CHOISES</b>
Valore campold	C1: campold != null C2: campold = null
Valore selectedDay	D1: selectedDay != null D2: selectedDay = null

<b>COMBINATIONS</b>	<b>ORACLE</b>
C1-D1	Corretto: vengono mostrati tutti gli slot per il giorno selezionato
C2-D1	Error: "impossibile trovare il campo"
C1-D2	Error: "giorno non disponibile"

## Slot

<b>METHOD NAME</b>	removeSlots
<b>PARAMETERS</b>	campold: String, selectedDay: DateTime, slot: Slot
<b>CATEGORIES</b>	<b>CHOISES</b>
Valore campold	C1: campold != null C2: campold = null
Valore selectedDay	D1: date != null D2: date = null
Valore slot	S1: slot != null S2: slot = null

<b>COMBINATIONS</b>	<b>ORACLE</b>
C1-D1-S1	Corretto: slot rimosso con successo
C2-D1-S1	Error: "impossibile trovare il campo"
C1-D2-S1	Error: "giorno non disponibile"
C1-D1-S2	Error: "slotId non trovato"

## Slot

<b>METHOD NAME</b>	updateSlotAvailability
<b>PARAMETERS</b>	campold: String, selectedDay: DateTime, slot: Slot
<b>CATEGORIES</b>	<b>CHOISES</b>
Valore campold	C1: campold != null C2: campold = null
Valore selectedDay	D1: date != null D2: date = null
Valore slot	S1: slot != null S2: slot = null

<b>COMBINATIONS</b>	<b>ORACLE</b>
C1-D1-S1	Corretto: lo stato dello slot viene aggiornato con successo
C2-D1-S1	Error: "impossibile trovare il campo"
C1-D2-S1	Error: "giorno non disponibile"
C1-D1-S2	Error: "slotId non trovato"

## Slot

<b>METHOD NAME</b>	removePastSlot
<b>PARAMETERS</b>	campold: String
<b>CATEGORIES</b>	<b>CHOISES</b>
Valore campold	C1: campold != null C2: campold = null

<b>COMBINATIONS</b>	<b>ORACLE</b>
C1	Corretto: gli slot passati sono stati rimossi con successo
C2	Error: "impossibile trovare il campo"

## Slot

<b>METHOD NAME</b>	generateHourlySlots
<b>PARAMETERS</b>	campold: String, selectedDay: dateTime, startHour: DateTime, endHour: DateTime
<b>CATEGORIES</b>	<b>CHOISES</b>
Valore campold	C1: campold != null C2: campold = null
Valore selectedDay	D1: date != null D2: date = null
Valore startHour	S1: startHour != null

	S2: startHour = null
Valore endHour	E1: endHour != null E2: endHour = null

COMBINATIONS	ORACLE
C1-D1-S1-E1	Corretto: vengono creati slot di un ora a partire dall'orario di inizio
C2-D1-S1-E1	Error: "impossibile trovare il campo"
C1-D2-S1-E1	Error: "giorno non disponibile"
C1-D1-S2-E1	Error: "orario di inizio non valido"
C1-D1-S1-E2	Error: "orario di fine non valido"

## Prenotazione

<b>METHOD NAME</b>	prenotazioneProvider.rifiutaPrenotazione
<b>PARAMETERS</b>	prenotazioneId : String, slotId: Slot, campold: String, dataPrenotazione: String
<b>CATEGORIES</b>	<b>CHOISES</b>
Valore prenotazioneId	P1 != null P2 = null
Valore campold	C1 != null C2 = null
Valore slotId	S1 != null S2 = null
Valore dataPrenotazione	D1 != null D2 = null

COMBINATIONS	ORACLE
P1-C1-S1-D1	Corretto: Prenotazione rifiutata con successo
P2-C1-S1-D1	Error: "prenotazione non trovata"
P1-C2-S1-D1	Error: "campold non valido"
P1-C1-S2-D1	Error: "Slot non esistente"
P1-C1-S1-D2	Error: "Data prenotazione non valida"

## Prenotazione

<b>METHOD NAME</b>	prenotazioneProvider.modificaPrenotazione
<b>PARAMETERS</b>	prenotazioneId : String, dataPrenotazione: String, idCampoPrecedente: String, slotPrecedente: Slot, selectedSlot: Slot
<b>CATEGORIES</b>	<b>CHOISES</b>
Valore prenotazioneId	P1 != null P2 = null
Valore idCampoPrecedente	F1 != null F2 = null



Valore dataPrenotazione	D1 != null D2 = null
Valore slotPrecedente	S1 != null S2 = null
Valore selectedSlot	L1 != null L2 = null

COMBINATIONS	ORACLE
P1-F1-D1-S1-L1	Corretto: Prenotazione modificata con successo
P2-F1-D1-S1-L1	Error: "prenotazione non trovata"
P1-F2-D1-S1-L1	Error: "Id campo precedente non trovato"
P1-F1-D2-S1-L1	Error: "Data prenotazione non valida"
P1-F1-D1-S2-L1	Error: "Slot precedente non valido"
P1-F1-D1-S1-L2	Error: "Lo slot selezionato non è disponibile"