

Università degli Studi di Salerno

Corso di Ingegneria del Software



UStrike!

Test Plan and Specifications

Versione 1.0

Data: 22/12/2025

Informazioni sul Progetto

Progetto: UStrike!
Documento: Test Plan and specifications
Versione: 1.0
Data: 22/12/2025

Coordinatore del progetto:

Nome
Matricola

Partecipanti:

Nome	Matricola
Russo Serena	0512119098
Stefanile Andrea	0512119557
Valito Marcello	0512119014

Scritto da:

Russo Serena, Stefanile Andrea, Valito Marcello

Revision History

Data	Versione	Descrizione	Autore
19/12/2025	0.1	Stesura iniziale del documento	Russo Serena, Stefanile Andrea
20/12/2025	0.2	Inserimento dei primi test cases e test frames	Stefanile Andrea, Valito Marcello
21/12/2025	0.3	Ultimazione di tutti i test cases e test frames e correzione di alcuni errori	Russo Serena, Stefanile Andrea, Valito Marcello
22/12/2025	1.0	Revisione e consegna del documento.	Russo Serena, Stefanile Andrea, Valito Marcello

Sommario	4
1.Introduzione	4
2.Relazioni con altri documenti	4
3.Funzionalità da testare	4
3.1Gestione Account ()	5
3.2Gestione Prenotazioni ()	5
3.3GestioneNotifiche()	5
4.Criteri Pass/Fail Testing	5
5.Approccio e Strategie di Testing.....	5
5.1Unit Testing (BLACK-BOX + Category Partition).....	5
5.2Integration Testing (WHITE-BOX + Path Coverage).....	6
5.3System Testing (Selenium).....	6
5.4Sospensione e Ripresa	6
5.5Materiali per il testing	7
6.Test Cases and Test frames.....	7
6.1 – Login	7
Test Frame.....	7
Test cases.....	8
6.2 – Aggiungi Prenotazione.....	9
Test Frame.....	10
Test cases	10
6.3-Registrazione	12
Test Frame.....	13
Test cases	14
6.4-Modifica profilo personale	17
Test Frame.....	17
Test cases	17
6.5-Rifiuta Prenotazione.....	19
Test Frame.....	19
Test Cases	19
6.6 - Annulla Prenotazione	20
Test Frame	20
Test cases	21
6.7-Assegna Risorsa	21
Test Frame.....	22
Test Cases	22
6.8 - Visualizza prenotazione personali.....	23

Test Frame	23
Test cases	23
6.9 Modifica Password	24
Test Frame.....	25
Test Cases	25

Sommario

Sommario

1.Introduzione

Il Test Plan è un documento che si focalizza sugli aspetti manageriali del testing: gestisce lo sviluppo e le attività di testing effettuate sul sistema UStrike! creato. Saranno identificati gli elementi e le funzionalità da testare, le strategie di testing, gli strumenti utilizzati per effettuarlo.

Lo scopo del testing è quello di rilevare gli errori in maniera pianificata all'interno del codice realizzato, in modo che essi non si ripetano durante l'utilizzo da parte dell'utente finale. I risultati dei test servono per intervenire nei punti in cui sono presenti défaillance.

2.Relazioni con altri documenti

Il test planning è in stretta relazione con i documenti prodotti fino ad ora:

- RAD (Requirement Analysis Document): i test case sono basati sulle funzionalità individuate nei 38 Use Cases
- SDD (System Design Document): la definizione dei sottosistemi (MVC a 13 servlet) e i servizi da sottoporre a testing
- ODD (Object Design Document): le classi e le interfacce definiscono il scope esatto del testing

3.Funzionalità da testare

Le funzionalità sottoposte a test, suddivise per gestione del sistema:

3.1 Gestione Account ()

- Login (cliente, staff, manager) x
- Registrazione(cliente)
- Modifica profilo personale(cliente)
- Modifica password(cliente)

3.2 Gestione Prenotazioni ()

- Aggiungi prenotazione (cliente)
- Annulla prenotazione (cliente)
- Accetta prenotazione (staff)
- Rifiuta prenotazione (staff)
- Assegna risorsa a prenotazione (staff)
- Visualizza prenotazioni personali (cliente)

3.3 GestioneNotifiche ()

- Invia Email conferma(sistema)
- Invia Email rifiuto(sistema)

4.Criteri Pass/Fail Testing

Pass: L'output osservato non corrisponde all'output atteso secondo le post-condizioni specificate nell'ODD.

Fail: L'output osservato è uguale all'output atteso.

Una volta riscontrata una failure:

1. Si registra il difetto
2. Si interviene per correggerlo
3. Si effettua il testing nuovamente per appurare che non abbia prodotto effetti collaterali

5.Approccio e Strategie di Testing

L'approccio della fase di testing si compone di tre livelli:

5.1 Unit Testing (BLACK-BOX + Category Partition)

Livello: Componenti singole (Service,Utills,Model)

Tecnica: BLACK-BOX testing

- Focus sul comportamento input/output senza considerare la struttura interna
- Utilizzo della strategia Category Partition:

- Decomposizione dello spazio di input in categorie
- Partizionamento delle categorie in classi di equivalenza ("scelte")
- Specificazione di "combinazioni" delle scelte per creare istanze di test case
- Determinazione dei risultati attesi

Esempio - UserService::authenticateUser():

- Categoria 1: Email (valida, invalida, inesistente)
- Categoria 2: Password (corretta, sbagliata, null)
- Categoria 3: Stato account (attivo, disabilitato)
- Combinazioni: $3 \times 3 \times 2 = 18$ test case

5.2 Integration Testing (WHITE-BOX + Path Coverage)

Livello: Integrazione tra Servlet ↔ Service ↔ Database

Tecnica: WHITE-BOX testing

- Testing della logica interna dei sottosistemi
- Concetto di "cammino": sequenza di istruzioni attraversata durante un'esecuzione
- Independent Path Coverage: numero finito di cammini indipendenti che coprono la maggior parte dei restanti percorsi

5.3 System Testing (Selenium)

Livello: Sistema completo end-to-end

Tecnica: Functional Testing

- Test dei flussi utente completi (UC)
- Interazione con browser (Chrome, Firefox)
- Verifica della UI response e comportamento

Test Scenarios:

1. Login → Dashboard → Visualizza Prenotazioni → Logout
2. Registrazione → Verifica email → Login → Crea Prenotazione
3. Manager: Crea Turni → Visualizza Turni → Modifica Turni

5.4 Sospensione e Ripresa

La fase di testing può essere sospesa se si riscontra un difetto critico nel sistema per essere poi ripresa dopo aver risolto il problema riscontrato. I criteri di sospensione:

- Critico: Errore che impedisce il funzionamento di un UC (es. DB non connesso)
- Maggiore: Errore che causa risultati errati (es. mora non calcolata)
- Minore: Errore estetico o di validazione (es. messaggio di errore incompleto)

Solo i difetti Critico e Maggiore sospendono il testing.

5.5 Materiali per il testing

Infrastruttura:

- Web Server: Apache Tomcat 9 in locale
- Client-web: Browser (Chrome, Firefox)
- Database: MySQL 8.0
- IDE: NetBeans o IntelliJ IDEA

Strumenti di Testing:

- JUnit 4 - Unit Testing framework
- Mockito 3 - Mock objects per isolamento
- Selenium IDE 4 - Browser automation
- MySQL Workbench - Gestione DB test
- Postman (opzionale) - API testing manuale
- JaCoCo - Code coverage analysis

Dati di Test:

- DB di test con dati di seed predefiniti
- Utenti di test per ogni ruolo (Cliente, Staff,)
- Servizi, Risorse, Premi predefiniti

6. Test Cases and Test frames

6.1 – Login

Parametro: Email	
Categorie	Scelte
Username nel database - EM	1. Non esiste nel database [property NotEM] 2. Username esistente [property EM]

Parametro: Password	
Categorie	Scelte
Password associata a username nel database - PA	1. Password associata a username nel database [property PA] 2. Password non associata a username nel database [property NotPA]

Test Frame

Codice	Combinazione	Esito
TC1	EM1.PA2	Visualizzazione pagina di autenticazione con messaggio "username o password non corretta"
TC2	EM2.PA1	Visualizzazione pagina utente registrato
TC3	EM2.PA2	Visualizzazione pagina di autenticazione con messaggio "username o password non corretta"

Test cases

Test case ID:	TC1
Precondizione:	
L'utente si trova sulla pagina di login.	
Flusso di eventi:	
L'utente compila il form:	
Input	Valore
Email	a.rossi24@mail.com
Password	wwwwwwwww
L'utente clicca sul pulsante Accedi	
Oracolo	
Autenticazione fallita perché non esiste una mail corrispondente nel database.	

Test case ID:	TC2
Precondizione:	
L'utente si trova sulla pagina di login.	
Flusso di eventi:	
L'utente compila il form:	
Input	Valore
Email	g.verdi@mail.com
Password	Gverdi12
L'utente clicca sul pulsante Accedi	
Oracolo	
Autenticazione riuscita essendo che la mail è registrata nel database e la password è associata alla mail	

Test case ID:	TC3
Precondizione:	
L'utente si trova sulla pagina di login.	
Flusso di eventi:	
L'utente compila il form:	
Input	Valore
Email	f.gialli@mail.com
Password	F1234
L'utente clicca sul pulsante Accedi	
Oracolo	
Autenticazione fallita essendo che la mail esiste nel database ma la password associata non è F1234	

6.2 – Aggiungi Prenotazione

Parametro: idServizio	
Categorie	Scelte
Servizio nel database - SE	1. Non esiste nel database [property NotSD] 2. Servizio esistente [property SD]

Parametro: data	
Categorie	Scelte
Data nel database - DD	1. Data non disponibile [property NotDD] 2. Data disponibile [property DD]

Parametro: fascia	
Categorie	Scelte
Fascia oraria nel database - FO	1. Fascia oraria non disponibile [property NotFD] 2. Fascia oraria disponibile [property FD]

Parametro: partecipanti	
Categorie	Scelte
Formato nome- FN	1. Formato nome dei partecipanti non corretto [property NotFN] 2. Formato nome dei partecipanti corretto [property FN]
Lunghezza nome- LN	1. Lunghezza nome dei partecipanti superiore a quella stabilita [property NotLN] 2. Lunghezza nome dei partecipanti inclusa in quella stabilita [property LN]

Test Frame

Codice	Combinazione	Esito
TC4	SE2.DD1	Visualizzazione pagina di prenotazione con messaggio "Data non disponibile"
TC5	SE2.DD2.FO1	Visualizzazione pagina di prenotazione con messaggio "Fascia oraria non disponibile"
TC6	SE2.DD2.FO2.FN1.LN2	Visualizzazione pagina di prenotazione con messaggio "Nome dei partecipanti non corretto"
TC7	SE2.DD2.FO2.FN2.LN1	Visualizzazione pagina di prenotazione con messaggio "Nome dei partecipanti non corretto"
TC8	SE2.DD2.FO2.FN2.LN2	Prenotazione avvenuta con successo ed inoltramento della richiesta

Test cases

Test case ID:	TC4
Precondizione:	
L'utente si trova sulla pagina di prenotazione bowling.	
Flusso di eventi:	
L'utente compila il form:	
Input	Valore
idServizio	1
data	2025-12-25
L'utente clicca sul pulsante Conferma	
Oracolo	
Prenotazione fallita in quanto la data inserita "2025-12-25" non è disponibile.	

Test case ID:	TC5
Precondizione:	
L'utente si trova sulla pagina di prenotazione bowling.	
Flusso di eventi:	
L'utente compila il form:	

Input	Valore
idServizio	1
data	2025-12-25
fascia	pomeriggio
L'utente clicca sul pulsante Conferma	
Oracolo	
Prenotazione fallita in quanto la fascia oraria "pomeriggio" non è disponibile.	

Test case ID:	TC6
Precondizione:	
L'utente si trova sulla pagina di prenotazione bowling.	
Flusso di eventi:	
L'utente compila il form:	
Input	Valore
idServizio	1
data	2025-12-25
fascia	pomeriggio
partecipanti	"Mario\$", "L&igi"
L'utente clicca sul pulsante Conferma	
Oracolo	
Prenotazione fallita in quanto i nomi dei partecipanti inseriti "Mario\$", "L&igi" presentano caratteri speciali.	

Test case ID:	TC7
Precondizione:	
L'utente si trova sulla pagina di prenotazione bowling.	
Flusso di eventi:	
L'utente compila il form:	
Input	Valore
idServizio	1
data	2025-12-25
fascia	pomeriggio
partecipanti	"Supercalifragilistichespinalidoso", "Luigi"
L'utente clicca sul pulsante Conferma	
Oracolo	

Prenotazione fallita in quanto i nomi dei partecipanti "Supercalifragilistichepsalidoso", "Luigi" non rispettano i termini di lunghezza.

Test case ID:	TC8										
Precondizione:											
L'utente si trova sulla pagina di prenotazione bowling.											
Flusso di eventi:											
L'utente compila il form:											
<table><tr><th>Input</th><th>Valore</th></tr><tr><td>idServizio</td><td>1</td></tr><tr><td>data</td><td>2025-12-25</td></tr><tr><td>fascia</td><td>pomeriggio</td></tr><tr><td>partecipanti</td><td>"Mario", "Luigi"</td></tr></table>		Input	Valore	idServizio	1	data	2025-12-25	fascia	pomeriggio	partecipanti	"Mario", "Luigi"
Input	Valore										
idServizio	1										
data	2025-12-25										
fascia	pomeriggio										
partecipanti	"Mario", "Luigi"										
L'utente clicca sul pulsante Conferma											
Oracolo											
Procedura di prenotazione avvenuta con successo.											

6.3-Registrazione

Parametro: Email	
Categorie	Scelte
Email nel database – EM	1. Email esiste nel database [property EM] 2.Email non esiste nel database [property NotEM]
Formato email – FE	1.Formato email corretto [property FM] 2.Formato email non corretto [property NotFM]

EM.FE.FN.FC.FP.LP.CP

Parametro: Nome	
Categorie	Scelte
Formato nome – FN	1.Formato nome corretto [property FN] 2.Formato nome non corretto [property NotFN]

Parametro: Cognome	
Categorie	Scelte
Formato cognome – FC	1.Formato cognome corretto [property FC] 2.Formato cognome non corretto [property NotFC]

Parametro: Password

Categorie	Scelte
Formato password- FP	1.Formato password corretto [property FP] 2.Formato password non corretto [property NotFP]
Lunghezza password- LP	1.Lunghezza password corretta [property LP] 2.Lunghezza password non corretta [property not LP]

Parametro: Conferma password	
Categorie	Scelte
Conferma password uguale a Password- CP	1.Le due password sono uguali [property CP] 2.Le due password non sono uguali [property NotCP]

Test Frame

Codice	Combinazione	Esito
TC9	EM1.FE1.FN1.FC1.FP1.LP1.CP1	Registrazione avvenuta con successo
TC10	EM2.FE1.FN1.FC1.FP1.LP1.CP1	Visualizzazione pagina di Registrazione con messaggio "email già presente"
TC11	EM1.FE2.FN1.FC1.FP1.LP1.CP1	Visualizzazione pagina di autenticazione con messaggio "formato email errato"
TC12	EM1.FE1.FN2.FC1.FP1.LP1.CP1	Visualizzazione pagina di autenticazione con messaggio "formato nome errato"
TC13	EM1.FE1.FN1.FC2.FP1.LP1.CP1	Visualizzazione pagina di autenticazione con messaggio "formato cognome errato"
TC14	EM1.FE1.FN1.FC1.FP2.LP1.CP1	Visualizzazione pagina di autenticazione con messaggio "formato password errato"
TC15	EM1.FE1.FN1.FC1.FP2.LP2.CP1	Visualizzazione pagina di autenticazione con messaggio "password troppo corta"
TC16	EM1.FE1.FN1.FC1.FP1.LP1.CP2	Visualizzazione pagina di autenticazione con messaggio "le due password devono combaciare"

Test cases

Test case ID:	TC9
Precondizione:	
L'utente si trova sulla pagina di Registrazione. La mail f.bianchi@mail.com non è registrata	
Flusso di eventi:	
L'utente compila il form:	
Input	Valore
Nome	Francesco
Cognome	Bianchi
Email	f.bianchi@mail.com
Password	Fbianchi12!
Conferma password	Fbianchi12!
L'utente clicca sul pulsante Registrati	
Oracolo	
Registrazione avvenuta correttamente	

Test case ID:	TC10
Precondizione:	
L'utente si trova sulla pagina di Registrazione.	
Flusso di eventi:	
L'utente compila il form:	
Input	Valore
Nome	Francesco
Cognome	Bianchi
Email	f.bia2e2e2e
Password	Fbianchi12!
Conferma password	Fbianchi12!
L'utente clicca sul pulsante Registrati	
Oracolo	
Registrazione fallita, formato email non rispettato	

Test case ID:	TC11
Precondizione:	
L'utente si trova sulla pagina di Registrazione. La mail f.bianchi@mail.com è registrata.	
Flusso di eventi:	
L'utente compila il form:	

Input	Valore
Nome	Francesco
Cognome	Bianchi
Email	f.bianchi@mail.com
Password	Fbianchi12!
Conferma password	Fbianchi12!
L'utente clicca sul pulsante Registrati	
Oracolo	
Registrazione fallita, email già presente nel database	

Test case ID:	TC12
Precondizione:	
L'utente si trova sulla pagina di Registrazione. La mail f.bianchi@mail.com non è registrata.	
Flusso di eventi:	
L'utente compila il form:	
Input	Valore
Nome	Fr!fd!!!!!!
Cognome	Bianchi
Email	f.bianchi@mail.com
Password	Fbianchi12!
Conferma password	Fbianchi12!
L'utente clicca sul pulsante Registrati	
Oracolo	
Registrazione fallita, formato nome errato	

Test case ID:	TC13
Precondizione:	
L'utente si trova sulla pagina di Registrazione. La mail f.bianchi@mail.com non è registrata	
Flusso di eventi:	
L'utente compila il form:	
Input	Valore
Nome	Francesco
Cognome	B!!!!!!.....
Email	f.bianchi@mail.com
Password	Fbianchi12!
Conferma password	Fbianchi12!
L'utente clicca sul pulsante Registrati	

Oracolo
Registrazione fallita, formato cognome errato

Test case ID:	TC14
Precondizione:	
L'utente si trova sulla pagina di Registrazione. La mail f.bianchi@mail.com non è registrata	
Flusso di eventi:	
L'utente compila il form:	
Input	Valore
Nome	Francesco
Cognome	Bianchi
Email	f.bianchi@mail.com
Password	adwsad
Conferma password	Fbianchi12!
L'utente clicca sul pulsante Registrati	
Oracolo	
Registrazione fallita, formato password errato	

Test case ID:	TC15
Precondizione:	
L'utente si trova sulla pagina di Registrazione. La mail f.bianchi@mail.com non è registrata	
Flusso di eventi:	
L'utente compila il form:	
Input	Valore
Nome	Francesco
Cognome	Bianchi
Email	f.bianchi@mail.com
Password	F
Conferma password	Fbianchi12!
L'utente clicca sul pulsante Registrati	
Oracolo	
Registrazione fallita, password troppo corta	

Test case ID:	TC16
Precondizione:	
L'utente si trova sulla pagina di Registrazione. La mail f.bianchi@mail.com non è registrata	
Flusso di eventi:	
L'utente compila il form:	

Input	Valore
Nome	Francesco
Cognome	Bianchi
Email	f.bianchi@mail.com
Password	Fbianchi12!
Conferma password	Fbianchi13!
L'utente clicca sul pulsante Registrati	
Oracolo	
Registrazione fallita, le due password non corrispondono	

6.4-Modifica profilo personale

Parametro: Nome	
Categorie	Scelte
Formato nome – FN	1.Formato nome corretto [property FN] 2.Formato nome non corretto [property NotFN]

Parametro: Cognome	
Categorie	Scelte
Formato cognome – FC	1.Formato cognome corretto [property FC] 2.Formato cognome non corretto [property NotFC]

Test Frame

Codice	Combinazione	Esito
TC17	FN1.FC1	Visualizzazione profilo personale aggiornato
TC18	FN2.FC1	Visualizzazione pagina di modifica con messaggio "formato nome non valido"
TC19	FN1.FC2	Visualizzazione pagina di modifica con messaggio "formato cognome non valido"

Test cases

Test case ID:	TC17
Precondizione:	

L'utente è autenticato. L'utente è nella sua pagina informazioni personali	
Flusso di eventi:	
L'utente inserisce:	
Input	Valore
Nome	Marinella
Cognome	Bianco
L'utente clicca sul pulsante Modifica informazioni	
Oracolo	
Le informazioni sono state modificate con successo.	

Test case ID:	TC18
Precondizione:	
L'utente è autenticato. L'utente è nella sua pagina informazioni personali	
Flusso di eventi:	
L'utente inserisce:	
Input	Valore
Nome	Merola!"!1re
Cognome	Bianco
L'utente clicca sul pulsante Modifica informazioni	
Oracolo	
Modifica fallita, formato nome errato	

Test case ID:	TC19
Precondizione:	
L'utente è autenticato. L'utente è nella sua pagina informazioni personali	
Flusso di eventi:	
L'utente inserisce:	
Input	Valore
Nome	Marinella
Cognome	Biancsda"!E£"T£TSc
L'utente clicca sul pulsante Modifica informazioni	
Oracolo	
Modifica fallita, formato nome errato	

6.5-Rifiuta Prenotazione

Parametro: ID Prenotazione	
Categorie	Scelte
Esistenza nel database - EP	1. Non esiste nel database [property NotEP] 2. Esiste nel database [property EP]

Parametro: Motivazione Rifiuto	
Categorie	Scelte
Lunghezza Motivazione - LM	1. Lunghezza = 0 – campo vuoto [property invalidLMValue] 2. Lunghezza >= 1 [property validLMValue]

Parametro: Stato Prenotazione	
Categorie	Scelte
Stato prenotazione - SP	1. Stato = "In attesa" [property validSPValue] 2. Stato != "In attesa" [property invalidSPValue]

Test Frame

Codice	Combinazione	Esito
TC20	EP2.LM2.SP1	Visualizzazione pagina di prenotazione con messaggio "Prenotazione rifiutata con successo"
TC21	EP2.LM1.SP2	Visualizzazione pagina di prenotazione con messaggio "Motivazione: campo obbligatorio"

Test Cases

Test case ID:	TC20
Precondizione:	
Lo staff si trova sulla pagina di gestione prenotazioni e una prenotazione con stato "In attesa" è selezionata.	
Flusso di eventi:	
Lo staff compila il form di rifiuto:	

Input	Valore
idPrenotazione	5
motivazione	"Pista/tavolo danneggiata/o"
Lo staff clicca sul pulsante "Rifiuta"	
Oracolo	
Prenotazione rifiutata con successo. Stato cambia a "Rifiutata", motivazione salvata nel DB, cliente riceve notifica di rifiuto, risorsa liberata.	

Test case ID:	TC21
Precondizione:	
Lo staff si trova sulla pagina di gestione prenotazioni.	
Flusso di eventi:	
Lo staff tenta di rifiutare una prenotazione senza inserire motivazione:	
Input	Valore
idPrenotazione	5
motivazione	(vuoto)
Lo staff clicca sul pulsante "Rifiuta"	
Oracolo	
Errore di validazione: "Motivazione: campo obbligatorio". Form ripristinato, prenotazione non modificata.	

6.6 - Annulla Prenotazione

Parametro: idPrenotazione	
Categorie	Scelte
Prenotazione nel database - PC	1. Non esiste nel database [property NotPC] 2. Servizio esistente [property PC]

Test Frame

Codice	Combinazione	Esito
TC22	PC1	Visualizzazione pagina della prenotazione con messaggio "Prenotazione inesistente"
TC23	PC2	Visualizzazione pagina della prenotazione con messaggio "Prenotazione annullata"

Test cases

Test case ID:	TC22
Precondizione:	
L'utente si trova sulla pagina delle sue prenotazioni.	
Flusso di eventi:	
L'utente seleziona la prenotazione che gli interessa:	
Input	Valore
idPrenotazione	110
L'utente clicca sul pulsante Conferma	
Oracolo	
Annullamento della prenotazione non riuscita in quanto l'IdPrenotazione non ha alcuna corrispondenza.	

Test case ID:	TC23
Precondizione:	
L'utente si trova sulla pagina delle sue prenotazioni.	
Flusso di eventi:	
L'utente seleziona la prenotazione che gli interessa:	
Input	Valore
idPrenotazione	1
L'utente clicca sul pulsante Conferma	
Oracolo	
Annullamento della prenotazione riuscita in quanto l'IdPrenotazione esiste nel database.	

6.7-Assegna Risorsa

Parametro: ID Prenotazione	
Categorie	Scelte
Esistenza nel database - EP	1. Non esiste nel database [property NotEP] 2. Esiste nel database [property EP]

Parametro: ID Risorsa	
Categorie	Scelte
Esistenza nel database - ER	1. Non esiste nel database [property NotER] 2. Esiste nel database [property ER]

Parametro: Disponibilità Risorsa per Fascia Oraria	
Categorie	Scelte
Stato risorsa per fascia oraria - SR	1. Risorsa occupata [property invalidSRValue] 2. Risorsa libera [property validSRValue]

Parametro: Compatibilità Risorsa-Servizio	
Categorie	Scelte
Servizio risorsa vs servizio prenotazione - CR	1. Servizio non corrisponde [property invalidCRValue] 2. Servizio corrisponde [if validSRValue] [property validCRValue]

Test Frame

Codice	Combinazione	Esito
TC24	EP2.ER2.SR2.CR2	"Risorsa assegnata con successo"
TC25	EP2.ER2.SR1.CR2	"Risorsa non disponibile per questa fascia oraria"

Test Cases

Test case ID:	TC24
Precondizione:	
Lo staff si trova sulla pagina di assegnazione risorse e una prenotazione Bowling con stato "Confermata" è selezionata per data "2025-12-25" fascia oraria "19:00-20:00".	
Flusso di eventi:	
Lo staff compila il form di assegnazione:	
Input	Valore
idPrenotazione	5
idRisorsa	2 (Pista Bowling - libera)
data	"2025-12-25"
fasciaOraria	"19:00-20:00"
Lo staff clicca sul pulsante "Assegna Risorsa"	
Oracolo	
Risorsa assegnata con successo. Prenotazione stato cambia a "Occupata risorsa", risorsa marcata come occupata per quella fascia oraria (19:00-20:00), cliente riceve notifica di assegnazione risorsa con numero della pista.	

Test case ID:	TC25
Precondizione:	
Lo staff si trova sulla pagina di assegnazione risorse e una prenotazione Bowling è selezionata per data "2025-12-25" fascia "19:00-20:00".	
Flusso di eventi:	
Lo staff tenta di assegnare una risorsa già occupata nello stesso orario:	
Input	Valore
idPrenotazione	5
idRisorsa	"Pista/tavolo danneggiata/o"
data	"2025-12-25"
fasciaOraria	"19:00-20:00"
Lo staff clicca sul pulsante "Assegna Risorsa"	
Oracolo	
Errore: "Risorsa non disponibile per questa fascia oraria". La pista è già occupata per quella data e fascia (19:00-20:00). Form ripristinato, assegnazione non completata.	

6.8 - Visualizza prenotazione personali

Parametro: idPrenotazione	
Categorie	Scelte
Prenotazione nel database - PC	1. Non esiste nel database [property NotPC] 2. Servizio esistente [property PC]

Test Frame

Codice	Combinazione	Esito
TC26	PC1	Visualizzazione pagina della prenotazione con messaggio "Prenotazione inesistente"
TC27	PC2	Visualizzazione pagina della prenotazione

Test cases

Test case ID:	TC26
Precondizione:	
L'utente si trova sulla pagina delle sue prenotazioni.	
Flusso di eventi:	
L'utente seleziona la prenotazione che gli interessa:	

Input	Valore
idPrenotazione	110
L'utente clicca sul pulsante Conferma	
Oracolo	
Visualizzazione della prenotazione non riuscita in quanto l'IdPrenotazione non ha alcuna corrispondenza.	

Test case ID:	TC27
Precondizione:	
L'utente si trova sulla pagina delle sue prenotazioni.	
Flusso di eventi:	
L'utente seleziona la prenotazione che gli interessa:	
Input	Valore
idPrenotazione	2
L'utente clicca sul pulsante Conferma	
Oracolo	
Visualizzazione della prenotazione riuscita in quanto l'IdPrenotazione esiste nel database.	

6.9 Modifica Password

Parametro: Autenticazione	
Categorie	Scelte
Autenticazione - AU	1. Utente non autenticato [property invalidAUValue] 2. Utente autenticato [property validAUValue]

Parametro: Vecchia Password	
Categorie	Scelte
Corrispondenza password attuale - VPC	1. Non corrisponde [property invalidVPCValue] 2. Corrisponde [property validVPCValue]

Parametro: Formato Nuova Password	
Categorie	Scelte
Formato nuova password - FNP	1. Formato non rispettato [property invalidFNPValue] 2. Formato valido [property validFNPValue]

Parametro: Nuova Password Confermata	
Categorie	Scelte
Corrispondenza conferma password - NPC	1. Non corrisponde a nuova password [property invalidNPCValue] 2. Corrisponde a nuova password [property validNPCValue]

Test Frame

Codice	Combinazione	Esito
TC28	AU1.VPC1.FNP1.NPC1	"Password modificata con successo"
TC29	AU2.VPC1.FNP1.NPC1	"Devi prima accedere per poter cambiare la password"
TC30	AU1.VPC2.FNP1.NPC1	"La vecchia password non corrisponde"
TC31	AU1.VPC1.FNP2.NPC1	"La nuova password non rispetta il formato richiesto"
TC32	AU1.VPC2.FNP1.NPC2	"La password ripetuta non è uguale a quella inserita"

Test Cases

Test case ID:	TC28
Precondizione:	
L'utente che invia la richiesta di modifica password è autenticato all'interno del sistema e la password dell'account è "gwE33@Q".	
Flusso di eventi:	
Modifica password:	
Input	Valore
VecchiaPassword	"gwE33@Q"
NuovaPassword	"EW2B4@D"
RipetizioneNuovaPassword	"EW2B4@D"
//	
Oracolo	
Password modificata con successo.	

Test case ID:	TC29								
Precondizione:									
L'utente che invia la richiesta di modifica password non è autenticato all'interno del sistema.									
Flusso di eventi:									
Tentativo accesso modifica password:									
<table border="1"> <thead> <tr> <th>Input</th><th>Valore</th></tr> </thead> <tbody> <tr> <td>VecchiaPassword</td><td>"vuoto"</td></tr> <tr> <td>NuovaPassword</td><td>"vuoto"</td></tr> <tr> <td>RipetizioneNuovaPassword</td><td>"vuoto"</td></tr> </tbody> </table>		Input	Valore	VecchiaPassword	"vuoto"	NuovaPassword	"vuoto"	RipetizioneNuovaPassword	"vuoto"
Input	Valore								
VecchiaPassword	"vuoto"								
NuovaPassword	"vuoto"								
RipetizioneNuovaPassword	"vuoto"								
Reindirizzato alla pagina di login.									
Oracolo									
Devi prima accedere per poter cambiare la password.									

Test case ID:	TC30								
Precondizione:									
L'utente che invia la richiesta di modifica password è autenticato all'interno del sistema e la password del sistema è "gwE33@Q ".									
Flusso di eventi:									
Modifica password:									
<table border="1"> <thead> <tr> <th>Input</th><th>Valore</th></tr> </thead> <tbody> <tr> <td>VecchiaPassword</td><td>"gwE22@q"</td></tr> <tr> <td>NuovaPassword</td><td>"EW2B4@D"</td></tr> <tr> <td>RipetizioneNuovaPassword</td><td>"EW2B4@D"</td></tr> </tbody> </table>		Input	Valore	VecchiaPassword	"gwE22@q"	NuovaPassword	"EW2B4@D"	RipetizioneNuovaPassword	"EW2B4@D"
Input	Valore								
VecchiaPassword	"gwE22@q"								
NuovaPassword	"EW2B4@D"								
RipetizioneNuovaPassword	"EW2B4@D"								
//									
Oracolo									
La vecchia password non corrisponde.									

Test case ID:	TC31
Precondizione:	
L'utente che invia la richiesta di modifica password è autenticato all'interno del sistema e la password del sistema è "gwE33@Q ".	
Flusso di eventi:	
Modifica password con formato non rispettato:	

Input	Valore
VecchiaPassword	"gwE22@q"
NuovaPassword	"EW2B4D"
RipetizioneNuovaPassword	"EW2B4D"
//	
Oracolo	
La nuova password non rispetta il formato richiesto.	

Test case ID:	TC32
Precondizione:	
L'utente che invia la richiesta di modifica password è autenticato all'interno del sistema e la password del sistema è "gwE33@Q ".	
Flusso di eventi:	
Modifica password con conferma non corrispondente:	
Input	Valore
VecchiaPassword	"gwE22@q"
NuovaPassword	"EW2B4@D"
RipetizioneNuovaPassword	"EwB4@D"
//	
Oracolo	
La password ripetuta non è uguale a quella inserita.	