

Università degli Studi di Salerno

Corso di Ingegneria del Software

**UniMeet
TestPlan
Versione 1.0**



GitHub del progetto: [/UniMeet](#)

Data: 13/12/2024

Progetto: UniMeet	Versione: 1.0
Documento: Test plan	Data: 13/12/2024

Partecipanti:

Nome	Matricola
Lorenza Rosa Pia Natale	0512116647
Giovanni Tufano	05121120227
Antonio Del Vecchio	0512118501
Ciro Danzilli	0512111007

Scritto da:	Tutti i membri del gruppo
--------------------	---------------------------

Revision History

Data	Versione	Descrizione	Autore
13/12/2024	1.0	TestPlan	Tutto il gruppo

1. INTRODUZIONE.....	4
2. RELAZIONI CON ALTRI DOCUMENTI	4
3. FUNZIONALITA' DA TESTARE	5
4. CRITERI PASS/FAIL.....	5
5. APPROCCIO	6
6. SOSPENSIONE E RIPRESA	7
7. MATERIALE PER IL TESTING	7
8. CASI DI TEST	8
8.1 Registrazione studente.....	8
8.2. Registrazione docente.....	10
8.3 Login.....	12
8.4 Logout.....	13
8.5 Prenotazione ricevimento.....	13
8.6 Conferma prenotazione	14
8.7 Rifiuto ricevimento	14
8.8 Aggiunta ricevimento	15
8.9 Modifica ricevimento	15
8.10 Modifica password	16

1. INTRODUZIONE

Il Test Plan è un documento che si concentra sugli aspetti gestionali del processo di testing. Serve a pianificare e monitorare le attività di testing effettuate sul sistema sviluppato. Attraverso il Test Plan saranno individuati gli elementi e le funzionalità da verificare, le strategie di testing adottate e gli strumenti impiegati per la loro esecuzione. L'obiettivo principale è rilevare e correggere gli errori nel codice realizzato in modo sistematico, assicurandosi che tali errori non emergano durante l'utilizzo da parte dell'utente finale.

2. RELAZIONI CON ALTRI DOCUMENTI

Il **Test Planning** è strettamente correlato ai documenti prodotti durante le precedenti fasi di sviluppo. I test case vengono progettati sulla base delle funzionalità definite nel **Documento di Raccolta e Analisi dei Requisiti (RAD)**. La selezione dei sottosistemi e dei servizi da testare deriva dal **System Design Document (SDD)**, mentre la definizione delle classi coinvolte nei test è stata effettuata utilizzando le specifiche contenute nell'**Object Design Document (ODD)**.

Questa interconnessione garantisce che il processo di testing sia allineato agli obiettivi progettuali e ai requisiti del sistema, assicurando una copertura completa e coerente delle funzionalità.

3. FUNZIONALITA' DA TESTARE

Nella seguente fase testiamo le seguenti funzionalità:

- Registrazione studente
- Registrazione docente
- Login
- Logout
- Modifica password
- Prenotazione ricevimento
- Conferma ricevimento
- Rifiuto ricevimento
- Aggiunta ricevimento
- Modifica ricevimento

4. CRITERI PASS/FAIL

Lo scopo del testing è identificare eventuali errori presenti nel sistema.

Un test può essere considerato riuscito se il suo output corrisponde a quello previsto dall'oracolo, il quale definisce sia le condizioni di errore sia i risultati attesi in caso di successo.

Durante la fase di testing, è fondamentale raggiungere i seguenti obiettivi:

- Verificare tutti i requisiti di alta priorità;
- Garantire una copertura del 100% dei metodi da testare;
- Ripetere i test ogni volta che vengono apportate modifiche al sistema.

5. APPROCCIO

L'approccio della fase di testing si compone di tre fasi. Si inizia con il testing di unità che ha lo scopo di testare le componenti del sistema singolarmente, poi si passa al testing di integrazione in cui le componenti del sistema verranno combinate e testate come un unico gruppo ed infine, si effettua il testing di sistema che mira a verificare il funzionamento dell'intero sistema. Per il testing di unità si utilizzeranno **JUnit** e **Mockito**. Con **JUnit** ci focalizzeremo sul comportamento delle singole componenti testando gli input e gli output senza tener conto della loro struttura interna. Per simulare le dipendenze esterne di ciascuna componente utilizzeremo **Mockito**, isolando le parti da testare e creando dei mock per verificare i comportamenti in condizioni specifiche.

A causa della mancanza di fattibilità di effettuare un test esaustivo per l'ingente quantitativo di dati di input, si definiranno scenari rappresentativi e casi di test mirati per coprire le principali classi di equivalenza dei dati, al termine saranno specificati i casi di test dettagliati con valori effettivi per ciascun dato di input, determinando i risultati attesi e verificando quelli ottenuti.

Per il testing di integrazione si utilizzeranno ancora **JUnit** e **Mockito** per testare la logica interna del sottosistema o dell'oggetto considerato. Per trovare un errore nel codice, verranno definiti dati di test in grado di percorrere le parti errate del programma. Per testare le diverse parti del sistema si introduce il concetto di "cammino", ovvero, una sequenza di istruzioni attraversata durante un'esecuzione. Non sarà possibile testare ogni singolo cammino (dato l'elevato numero di questi ultimi), ma sarà identificato un numero finito di cammini indipendenti che combinati tra loro forniscono la copertura della maggior parte dei restanti cammini. Inoltre, sarà verificata l'interazione con il database utilizzando un'istanza di test di **MySQL**, testando query e operazioni sulle tabelle per garantire che i dati siano gestiti correttamente.

Per determinare se il sistema rispecchia tutti i requisiti funzionali e globali, sarà effettuato il testing tramite il tool **Selenium**. Con Selenium si simuleranno flussi utente che attraversano i principali percorsi funzionali del sistema, verificando il comportamento dell'interfaccia grafica e le interazioni con il backend. Saranno definiti casi di test per garantire che ogni operazione utente produca i risultati attesi. Verrà inoltre verificata la coerenza tra le azioni svolte dall'interfaccia e le operazioni di lettura e scrittura nel database **MySQL**, assicurando che il sistema funzioni come previsto in un contesto completo.

6. SOSPENSIONE E RIPRESA

La fase di testing viene sospesa nel momento in cui si riscontra un problema nel sistema e viene ripresa solo dopo aver risolto quest'ultimo.

7. MATERIALE PER IL TESTING

Gli strumenti utilizzati per il testing:

- MySQL
- Selenium
- JUnit
- Mockito

8. CASI DI TEST

8.1 Registrazione studente

Parametro: matricola	
Categorie	Scelte
Lunghezza matricola – LM	1. Lunghezza != 10 – campo vuoto [property invalidLMValue] 2. Lunghezza = 10 [property validLMValue]
Formato matricola – FM	1. Non rispetta il formato “05121” [property invalidFMValue] 2. Rispetta il formato “05121” [if validLEValue] [property validFMValue]
Esistenza nel database - EXMA	1. Esiste nel database [property NotEXMA] 2. Non esiste nel database [if validFEValue] [property EXMA]

Parametro: e-mail	
Categorie	Scelte
Lunghezza E-mail – LE	1. Lunghezza = 0 – campo vuoto [property invalidLEValue] 2. Lunghezza >= 1 [property validLEValue]
Formato E-mail – FE	1. Non rispetta il formato [property invalidLEValue] 2. Rispetta il formato [if validLEValue] [property validFEValue]
Esistenza nel database - EXEM	1. Esiste nel database [property NotEXEM] 2. Non esiste nel database [if validFEValue] [property EXEM]

Parametro: password	
Categorie	Scelte
Lunghezza Password – LP	1. Lunghezza = 0 – campo vuoto [property invalidLPValue]

	2. Lunghezza >= 1 [property validLPValue]
Formato Password – FP	1. Non rispetta il formato [property invalidFPValue] 2. Rispetta il formato [if validLPValue] [property validFPValue]

Parametro: ripeti password	
Categorie	Scelte
Lunghezza Ripeti Password – LRP	1. Lunghezza = 0 – campo vuoto [property invalidLRPValue] 2. Lunghezza >= 1 [property validLRPValue]
Coincidenza - OVLP	1. Il campo non coincide con il campo Password [property NotOverlap] 2. il campo coincide con il campo Password [if validFRPValue && validLPValue] [property Overlap]

CODICE	COMBIANZIONE	ORACOLO
TC_RegistrazioneSuccess	LM2.FM2.EXMA2.LE2.FE2.EXEM2.LP2.FP2.LRP2.OVLP2	“Registrazione avvenuta con successo!”
TC_RegistrazioneFailed1	LM2.FM2.EXMA2.LP2.FP2.LRP2.OVLP2.LE2.FE2.EXEM1	“E-mail già esistente”
TC_RegistrazioneFailed2	LM2.FM2.EXMA2.LP2.FP2.LRP2.OVLP2.LE2.FE1	“Formato e-mail non valido”
TC_RegistrazioneFailed3	LM2.FM2.EXMA2.LP2.FP2.LRP2.OVLP2.LE1	“E-mail - campo obbligatorio”
TC_RegistrazioneFailed4	LM2.FM2.EXMA2.LP2.FP2.LRP2.OVLP1.LE2.FE2.EXEM2	“Le password non coincidono”
TC_RegistrazioneFailed5	LM2.FM2.EXMA2.LP2.FP2.LRP1.LE2.FE2.EXEM2	“Ripeti password - campo obbligatorio”
TC_RegistrazioneFailed6	LM2.FM2.EXMA2.LP2.FP1.LRP2.OVLP2.LE2.FE2.EXEM2	“Formato password non valido”
TC_RegistrazioneFailed7	LM2.FM2.EXMA2.LP1.LRP2.OVLP1.LE2.FE2.EXEM2	“Password - campo

		<i>obbligatorio</i>
TC_RegistrazioneFailed 8	LM2.FM2.EXMA1.LE2.FE2.EXEM2.LP2.FP2.LRP2.OV LP2	<i>“Matricola già inserita”</i>
TC_RegistrazioneFailed 9	LM2.FM1.EXMA2.LE2.FE2.EXEM2.LP2.FP2.LRP2.OV LP2	<i>“Formato matricola non valido”</i>
TC_RegistrazioneFailed 10	LM1.FM2.EXMA2.LE2.FE2.EXEM2.LP2.FP2.LRP2.OV LP2	<i>“Matricola - campo obbligatorio”</i>

8.2. Registrazione docente

Parametro: codice docente	
Categorie	Scelte
Lunghezza Codice Docente – LCD	1. Lunghezza = 0 – campo vuoto [property invalidLCDValue] 2. Lunghezza = 6 [property validLCDValue]
Esistenza nel database - EXCD	1. Esiste nel database [property NotEXCD] 2. Non esiste nel database [if validFEValue] [property EXCD]

Parametro: e-mail	
Categorie	Scelte
Lunghezza E-mail – LE	1. Lunghezza = 0 – campo vuoto [property invalidLEValue] 2. Lunghezza >= 1 [property validLEValue]
Formato E-mail – FE	1. Non rispetta il formato [property invalidLEValue] 2. Rispetta il formato [if validLEValue] [property validFEValue]
Esistenza nel database - EXEM	1. Esiste nel database [property NotEXEM] 2. Non esiste nel database [if validFEValue] [property EXEM]

Parametro: password	
Categorie	Scelte
Lunghezza Password – LP	1. Lunghezza = 0 – campo vuoto [property invalidLPValue] 2. Lunghezza >= 1 [property validLPValue]
Formato Password – FP	1. Non rispetta il formato [property invalidFPValue] 2. Rispetta il formato [if validLPValue] [property validFPValue]

Parametro: ripeti password	
Categorie	Scelte
Lunghezza Ripeti Password – LRP	1. Lunghezza = 0 – campo vuoto [property invalidLRPValue] 2. Lunghezza >= 1 [property validLRPValue]
Coincidenza - OVLP	1. Il campo non coincide con il campo Password [property NotOverlap] 2. il campo coincide con il campo Password [if validFRPValue && validLPValue] [property Overlap]

CODICE	COMBIANZIONE	ORACOLO
TC_RegistrazioneSuccess	LCD2.EXCD2.LE2.FE2.EXEM2.LP2.FP2.LRP2.OVLP2	“Registrazione avvenuta con successo!”
TC_RegistrazioneFailed1	LCD2.EXCD2.LP2.FP2.LRP2.OVLP2.LE2.FE2.EXEM1	“E-mail già esistente”
TC_RegistrazioneFailed2	LCD2.EXCD2.LP2.FP2.LRP2.OVLP2.LE2.FE1	“Formato e-mail non valido”
TC_RegistrazioneFailed3	LCD2.EXCD2.LP2.FP2.LRP2.OVLP2.LE1	“E-mail - campo obbligatorio”
TC_RegistrazioneFailed4	LCD2.EXCD2.LP2.FP2.LRP2.OVLP1.LE2.FE2.EXEM2	“Le password non coincidono”
TC_RegistrazioneFailed5	LCD2.EXCD2.LP2.FP2.LRP1.LE2.FE2.EXEM2	“Ripeti

		<i>password - campo obbligatorio</i>
TC_RegistrazioneFailed6	LCD2.EXCD2.LP2.FP1.LRP2.OVLP2.LE2.FE2.EXEM2	<i>“Formato password non valido”</i>
TC_RegistrazioneFailed7	LCD2.EXCD2.LP1.LRP2.OVLP1.LE2.FE2.EXEM2	<i>“Password - campo obbligatorio”</i>
TC_RegistrazioneFailed8	LCD2.EXCD1.LE2.FE2.EXEM2.LP2.FP2.LRP2.OVLP2	<i>“Codice docente già inserito”</i>
TC_RegistrazioneFailed9	LCD1.EXCD2.LE2.FE2.EXEM2.LP2.FP2.LRP2.OVLP2	<i>“Codice docente – campo obbligatorio”</i>

8.3 Login

Parametro: e-mail	
Categorie	Scelte
Esistenza nel database – EXEM	1. Non esiste nel database [property NotEXEM] 2. Esiste nel database [property EXEM]

Parametro: password	
Categorie	Scelte
Associazione all'utente – AS	1. Non è correttamente associata all'e-mail [property invalidASValue] 2. È correttamente associata all'e-mail [if EXEM] [property validASValue]

CODICE	COMBIANZIONE	ORACOLO
TC_LoginSuccess	EXEM2.AS2	<i>“Login effettuato!”</i>
TC_LoginFailed1	EXEM2.AS1	<i>“Password errata”</i>
TC_LoginFailed2	EXEM1	<i>“L'utente non esiste”</i>

8.4 Logout

Parametro: login	
Categorie	Scelte
Utente loggato – UL	1. Utente non loggato [property NotUL]
	2. Utente loggato [property UL]

CODICE	COMBIANZIONE	ORACOLO
TC_LoginoutSuccess	USIN2	<i>“Logout effettuato!”</i>
TC_LoginoutFailed	USIN1	<i>“Devi essere loggato per poter effettuare il logout”</i>

8.5 Prenotazione ricevimento

categorie	Scelte
Giorno Ricevimento - GR	1. Giorno ancora non presente nel database [property EXGR] 2. Giorno già presente nel database [property NotEXGR]
Ora Ricevimento - OR	1. Ora ancora non presente nel database [property EXOR] 2. Ora già nel database [property NotExor]
Lunghezza Note - LN	1. Lunghezza =0 – campo vuoto [property NotEXLN] 2. Lunghezza >= 1 [property EXLN]

codice	combinazione	oracolo
TC_Prenotazione_Ricevimento_Ok1	GR1.OR1.NotEXLN	“prenotazione avvenuta”
TC_Prenotazione_Ricevimento_Ok2	GR1.OR1.EXLN	“prenotazione avvenuta”
TC_Prenotazione_Ricevimento_Ok3	GR2.OR1.EXLN	“prenotazione avvenuta”
TC_Prenotazione_Ricevimento_Ok4	GR2.OR1.NotEXLN	“prenotazione avvenuta”
TC_Prenotazione_Ricevimento_Failed1	GR2.OR2.EXLN	“orario già prenotato”
TC_Prenotazione_Ricevimento_Failed2	GR2.OR2.NotEXLN	“orario già prenotato”
TC_Prenotazione_Ricevimento_Failed3	GR1.OR2.EXLN	“prenotazione avvenuta”
TC_Prenotazione_Ricevimento_Failed4	GR1.OR2.NotEXLN	“prenotazione avvenuta”

8.6 Conferma prenotazione

categorie	scelte
Prenotazione - PR	<ol style="list-style-type: none">1. Prenotazione presente nel database e stato prenotazione “in sospeso” [property EXPR AND stato=” in sospeso”]2. Prenotazione non presente nel database [property NotEXPR]

codice	combinazione	oracolo
TC_AccettazionePrenotazione_Ok	PR1	“prenotazione ricevimento accettata”

8.7 Rifiuto ricevimento

categorie	scelte
Prenotazione - PR	<ol style="list-style-type: none">1. Prenotazione presente nel database e stato prenotazione “in sospeso” [property EXPR AND stato=” in sospeso”]2. Prenotazione non presente nel database [property NotEXPR]

codice	combinazione	oracolo
TC_AccettazionePrenotazione_Ok	PR1	“prenotazione ricevimento rifiutata”

8.8 Aggiunta ricevimento

categorie	scelte
Giorno Ricevimento - GR	<ol style="list-style-type: none"> 1. Lunghezza\geq1 [property ValidGRValue] 2. Lunghezza=0 - campo vuoto [property InvalidGRValue]
Ora Ricevimento - OR	<ol style="list-style-type: none"> 1. Rispetta il formato [property ValidORValue] 2. Non rispetta il formato [property InvalidORvalue]
Insegnamento Ricevimento - IR	<ol style="list-style-type: none"> 1. Lunghezza\geq1 [property ValidIRValue] 2. Lunghezza=0 – campo vuoto [property InvalidIRValue]

codice	combinazione	oracolo
TC_AggiuntaRicevimento_Ok	GR1.OR1.IR1	“Ricevimento aggiunto con successo”
TC_AggiuntaRicevimento_Failed1	GR2.OR1.IR1	“inserire giorno ricevimento”
TC_AggiuntaRicevimento_Failed2	GR1.OR2.IR1	“Formato data errato”
TC_AggiungiRicevimento_Failed3	GR1.OR2.IR2	“inserire insegnamento”
TC_AggiungiRicevimento_Failed4	GR2.OR2.IR2	“inserire correttamente i dati”

8.9 Modifica ricevimento

Giorno Ricevimento - GR	<ol style="list-style-type: none"> 1. Lunghezza\geq1 [property ValidGRValue] 2. Lunghezza=0 - campo vuoto [property InvalidGRValue]
Ora Ricevimento - OR	<ol style="list-style-type: none"> 1. Rispetta il formato [property ValidORValue] 2. Non rispetta il formato [property InvalidORvalue]
Insegnamento Ricevimento - IR	<ol style="list-style-type: none"> 1. Lunghezza\geq1 [property ValidIRValue] 2. Lunghezza=0 – campo vuoto [property InvalidIRValue]

codice	combinazione	oracolo
TC_ModificaRicevimento_Ok	GR1.OR1.IR1	“Ricevimento modificato con successo”
TC_ModificaRicevimento_Failed1	GR2.OR1.IR1	“inserire giorno ricevimento”
TC_ModificaRicevimento_Failed2	GR1.OR2.IR1	“Formato data errato”
TC_ModificaRicevimento_Failed3	GR1.OR2.IR2	“inserire insegnamento”
TC_ModificaRicevimento_Failed4	GR2.OR2.IR2	“inserire correttamente i dati”

8.10 Modifica password

Categorie	scelte
Associata all'utente - AU	<ol style="list-style-type: none"> la password non è associata all'utente nella sessione [property InvalidAUValue] La password è associata all'utente nella sessione [property ValidAUValue]
Lunghezza Nuova Password - LNP	<ol style="list-style-type: none"> Lunghezza =0 campo vuoto [property invalidLNPValue] Lunghezza >=1 [property ValidLNPValue]
Formato Nuova Password - FNP	<ol style="list-style-type: none"> Non rispetta il formato [property invalidFNPValue] Rispetta il formato [property ValidFNPValue]
Lunghezza Conferma Password - LCP	<ol style="list-style-type: none"> Lunghezza =0 campo vuoto [Property invalidLCPValue] Lunghezza >=1 [property ValidLCPValue]
Coincidono - OVLP	<ol style="list-style-type: none"> Non coincide con la nuova password [property NotOverlap] Coincide con la nuova password [if ValidFNPValue && ValidLCP] [property Overlap]

Codice	combinazione	oracolo
TC_ModificaPassword_Ok	AU2.LNP2.FNP2.LCP2.OVLP2	“password modificata”
TC_ModificaPassword_Failed1	AU2.LNP2.LCP2.OVLP1	“le password non coincidono”
TC_ModificaPassword_Failed2	AU2.LNP2.FNP1	“formato nuova password errato”
TC_ModificaPassword_Failed2	AU2.LNP1.LCP2	“inserire nuova password”
TC_ModificaPassword_Failed3	AU1.LNP2.LCP2.OVLP2	“password errata”