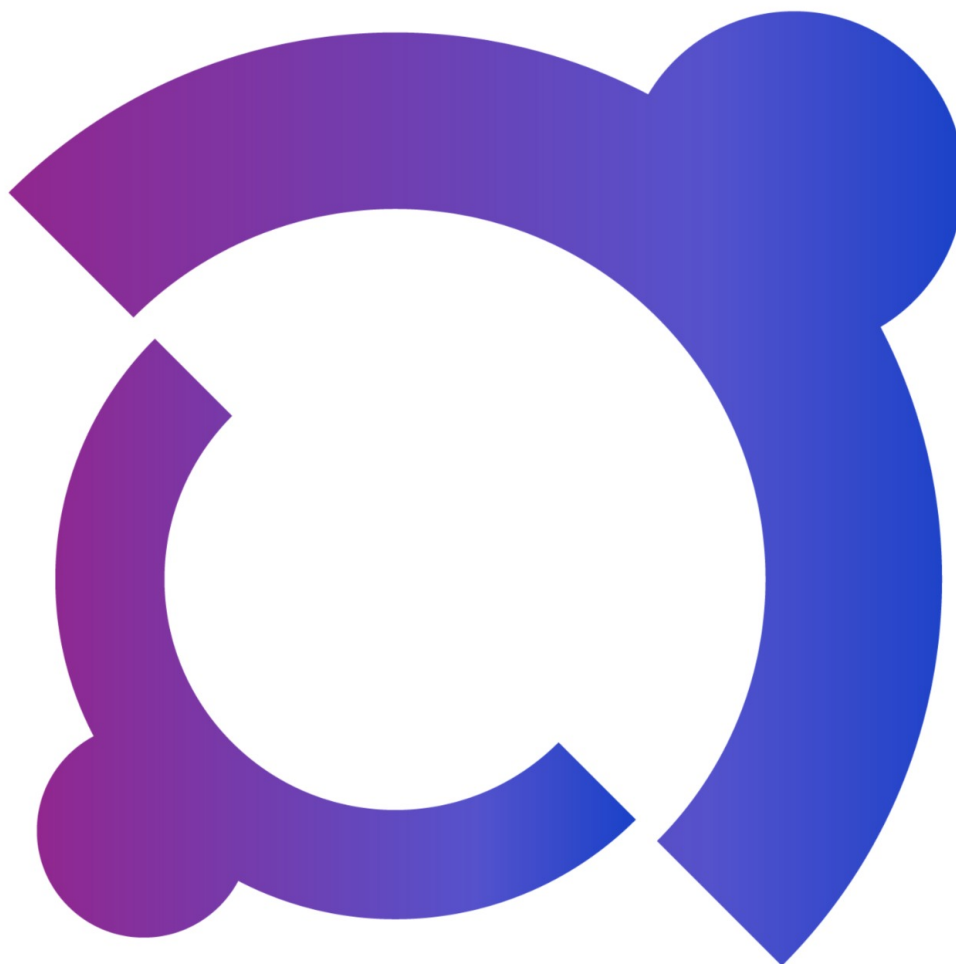




TEST CASE PLAN - OPENMEET



Test Case Plan

All'attenzione di: Prof. Carmine Gravino

Preparato da: Angelo Nazzaro, Yuri Brandi, Roberto Della Rocca, Francesco Granozio

Versione: 0.3



TEST CASE PLAN - OPENMEET

Sommario

Revision History	3
Team Members	4
1. Introduzione	5
2. Relazione con gli altri Documenti	5
3. Panoramica del Sistema	6
4. Funzionalità da Testare/Non Testare	7
5. Pass/Fail Criteria	7
6. Approccio	8
6.1 Testing di Unità	8
6.2 Testing Funzionale	8
6.3 Ispezione del Codice	8
6.4 Altre Tecnologie Utilizzate	9
7. Strumenti per il Testing	9
8. Test Cases	9
9. Specifica dei Test Cases	10
9.1 Gestione Piattaforma - GP	10
9.1.1 Ban Meeter	10
9.2 Gestione Meeter - MT	11
9.2.1 Segnalazione Meeter	11
9.2.2 Recupero Password	12
9.2.3 Registrazione	14
10. Pianificazione e Assegnazione ruoli	18



TEST CASE PLAN - OPENMEET

Revision History

Data	Versione	Descrizione	Autori
15/12/2022	0.1	Prima stesura.	Membri del Team
16/12/2022	0.2	Aggiunta delle sezioni: 3. Panoramica del sistema, 2. Relazione con gli altri documenti, 5. Pass/Fail Criteria e 8. Test Cases	AN
17/12/2022	0.3	Aggiunta delle sezioni: 1. Introduzione, 6. Approccio, 7. Strumenti per il Testing, 10. Pianificazione e Assegnazione ruoli	AN



TEST CASE PLAN - OPENMEET

Team Members

Nome e Cognome	Ruolo	Informazioni di contatto	Matricola	Acronimo
Angelo Nazzaro	Membro del Team	a.nazzaro13@studenti.unisa.it	0512110391	AN
Yuri Brandi	Membro del Team	y.brandi@studenti.unisa.it	0512109740	YB
Roberto Della Rocca	Membro del Team	r.dellarocca5@studenti.unisa.it	0512110802	RD
Francesco Granozio	Membro del Team	f.granozio1@studenti.unisa.it	0512111903	FG



TEST CASE PLAN - OPENMEET

1. Introduzione

OpenMeet mira alla creazione di una piattaforma che incentivi gli utenti, con particolare riguardo ai giovani, a fare nuove conoscenze ed esperienze a livello sociale nel rispetto dell'identità e della privacy del singolo.

Il seguente documento ha lo scopo di definire l'ambito, l'obiettivo e l'approccio applicati al fine di testare l'applicativo software e garantire un funzionamento corretto di quest'ultimo.

Le attività di testing pianificate sono state divise per gestioni, come specificate nel RAD. Si è deciso di testare le seguenti gestioni:

- **Gestione Meeter - GMT**
- **Gestione Piattaforma - GP**

2. Relazione con gli altri Documenti

Il seguente documento, ai fini della corretta individuazione dei test cases, è strettamente correlato agli atti:

- **Requirement Analysis Document - RAD**
I test cases pianificati sono elaborati sulla base dei requisiti funzionali descritti nel RAD.
Riferimento: [openMeet_RAD](#)
- **System Document Design - SDD**
I test cases dovranno rispettare la suddivisione in sottosistemi stabilita nell'SDD.
Riferimento: [openMeet_SDD](#)
- **Referenze RFC**
I test riguardanti i pattern degli indirizzi email sono stati strutturati sulla base dei seguenti standard:
 - [RFC 3696](#)
 - [RFC 5321](#)
 - [RFC 5322](#)
 - [RFC 6530](#)

TEST CASE PLAN - OPENMEET

3. Panoramica del Sistema

Il sistema proposto è basato sullo stile architettura Three-Tier e verrà dunque suddiviso nei seguenti livelli:

- **Presentation Tier:** include tutte le componenti grafiche che si interfacciano e presentano i dati all'utente;
- **Logic Tier:** include tutti gli oggetti relativi al controllo e alle entità che realizzano l'elaborazione, le regole di verifica richieste dall'applicazione;
- **Data Layer:** include tutti i componenti dedicati alla memorizzazione, al recupero e all'interrogazione di oggetti persistenti.

Per la gestione dei dati persistenti si è deciso di utilizzare un DBMS relazionale in quanto ci garantisce un accesso concorrente e consistente ai dati.

Il DBMS relazionale facilita la coerenza rispetto ai design goals stabiliti, in quanto garantisce:

- **Affidabilità dei dati:** è in grado di conservare intatto il contenuto di una base di dati e/o di permetterne la ricostruzione in caso di malfunzionamenti hardware e/o software.
- **Privatezza dei dati:** offre meccanismi di autenticazione e autorizzazione per l'accesso al DBMS stesso.
- **Atomicità delle operazioni:** consente l'esecuzione di transazioni, ovvero sequenze di operazioni atomiche, che se eseguite in modo lecito, producono una variazione nello stato della base di dati. In caso di insuccesso, la base di dati ritorna allo stato precedente rimanendo invariata.

Differentemente, la componente di messaggistica tra i client Mobile sfrutta una persistenza locale, specificamente SQLite/Room, al fine di garantire una conversazione peer-to-peer più sicura, senza sacrificare i vantaggi di un DBMS.

TEST CASE PLAN - OPENMEET

4. Funzionalità da Testare/Non Testare

Di seguito è riportato un elenco delle funzionalità sulle quali verranno effettuate le operazioni di testing suddivise per gestioni:

- **Gestione Meeter - GMT:**
 - Recupero Password
 - Registrazione
 - Segnalazione Meeter
- **Gestione Piattaforma - GP:**
 - Ban Meeter

Le altre funzionalità del sistema non verranno sottoposte a una fase di testing nel rispetto dei vincoli progettuali e tecnici imposti.

5. Pass/Fail Criteria

Il testing ha lo scopo di individuare discrepanze tra il comportamento atteso descritto tramite il modello del sistema e il comportamento osservato dal sistema implementato.

Lo scopo della fase di testing è quello di progettare test al fine di provare il sistema, rilevarne i problemi e risolverli.

L'esito di ogni test verrà descritto da un oracolo, ovvero la sequenza di dati in output o comandi che rappresentano la corretta esecuzione del test.

Un test avrà successo (**pass**) se, dato un input al sistema, l'output ottenuto è diverso dall'output atteso dall'oracolo. Il test avrà avuto successo nel rilevare problemi del sistema.

Un test fallisce (**fail**) se, dato un input al sistema, l'output ottenuto è equivalente a quello atteso dall'oracolo. Il test avrà fallito nel rilevare problemi del sistema.

TEST CASE PLAN - OPENMEET

6. Approccio

La procedura di testing sarà suddivisa in:

- Testing di Unità
- Testing Funzionale

Prima della fase di implementazione del sistema, avverrà la progettazione dei casi di test di sistema, perfezionati in seguito nella loro fase di esecuzione; durante la fase d'implementazione avverrà la progettazione dei casi di test di unità.

6.1 Testing di Unità

Il Testing di Unità ha lo scopo di trovare le differenze tra l'object design model e le corrispondenti componenti implementate.

I casi di test saranno definiti attraverso un approccio **whitebox** e saranno sviluppati e documentati in contemporanea allo sviluppo del codice, supportato dall'utilizzo del framework **JUnit**.

Le classi di test rispetteranno la nomenclatura **ProductionClassNameTest** e saranno revisionate e modificate da sviluppatori diversi.

6.2 Testing Funzionale

Il Testing Funzionale ha lo scopo di trovare le differenze tra lo use case model specificato nel RAD e il sistema implementato.

Nello specifico, si andranno a valutare e validare i requisiti funzionali al fine di individuare eventuali discrepanze tra il comportamento atteso e il comportamento osservato.

6.3 Ispezione del Codice

Durante lo sviluppo del sistema saranno eseguite periodiche attività di revisione del codice per assicurarsi che questo sia conforme al codestyle di Kotlin definito da Google LLC.

Le attività di revisione del codestyle avverranno tramite tool di Checkstyle, in particolare ci si affiderà alle built-in inspections di Android Studio.

TEST CASE PLAN - OPENMEET

6.4 Altre Tecnologie Utilizzate

Di seguito un elenco delle altre tecnologie che saranno utilizzate nella fase di testing:

- **Github Actions:** un tool di continuous integration e delivery (CI/CD) fornito da [Github](#) che verrà utilizzato per automatizzare la costruzione, il testing e il deployment.
- **JaCoco:** una libreria open-source che verrà utilizzata per la creazione di report sul code coverage.
- **Mockito:** per la costruzione degli stub e l'isolamento della componente testata.

7. Strumenti per il Testing

L'attività di testing verrà svolta tramite il sistema di emulazione di dispositivi mobile fornito da Android Studio e/o su dispositivi mobile reali Android based.

8. Test Cases

Per sviluppare i test cases sarà utilizzato il metodo del **Category Partition**. Tale metodo consiste nell'identificare per ogni funzionalità da testare dei parametri; per ogni parametro verranno individuate delle categorie, le quali saranno poi suddivise in scelte. Ad ogni scelta verrà assegnato un valore.

I test cases verranno definiti nel documento di **Test Case Specification (TCS)**.

TEST CASE PLAN - OPENMEET

9. Specifica dei Test Cases

9.1 Gestione Piattaforma - GP

9.1.1 Ban Meeter

Parametro: Motivazione	
Nome categoria	Scelte per la categoria
Lunghezza [LN]	1. Lunghezza < 1 OR Lunghezza > 255 ⇒ [error] 2. Lunghezza >= 1 AND Lunghezza <= 255 ⇒ [PROPERTY_LN_OK]
Parametro: Data Ora Fine	
Nome categoria	Scelte per la categoria
Correttezza [CT]	1. Data Ora Fine < Data Ora Corrente ⇒ [error] 2. Data Ora Fine >= Data Ora Corrente ⇒ [PROPERTY_CT_OK]

TEST FRAMES

Test Case ID	Test Frame	Esito
TC_1.1_1	LN1	Errato: Lunghezza motivazione invalida.
TC_1.2_1	LN2, CT1	Errato: Data Ora Fine antecedente alla Data Ora Corrente.
TC_1.2_2	LN2, CT2	Corretto.

TEST CASE PLAN - OPENMEET

9.2 Gestione Meeter - MT

9.2.1 Segnalazione Meeter

Parametro: Motivazione	
Nome categoria	Scelte per la categoria
Lunghezza [LN]	1. Lunghezza < 1 OR Lunghezza > 255 ⇒ [error] 2. Lunghezza >= 1 AND Lunghezza <= 255 ⇒ [PROPERTY_LN_OK]

TEST FRAMES

Test Case ID	Test Frame	Esito
TC_2.1_1	LN1	Errato: Lunghezza motivazione invalida.
TC_2.1_2	LN2	Corretto.

TEST CASE PLAN - OPENMEET

9.2.2 Recupero Password

Parametro: Email	
Nome categoria	Scelete per la categoria
Presenza Database [PD]	<ol style="list-style-type: none"> 1. Presente = false ⇒ [error] 2. Presente = true ⇒ [PROPERTY_PD_OK]
Parametro: Password	
Nome categoria	Scelte per la categoria
Lunghezza [LN]	<ol style="list-style-type: none"> 1. Lunghezza < 8 OR Lunghezza > 16 ⇒ [error] 2. Lunghezza >= 8 AND Lunghezza <= 16 ⇒ [PROPERTY_LN_OK]
Presenza Carattere Speciale [PCS]	<ol style="list-style-type: none"> 1. Presente = false ⇒ [error] 2. Presente = true ⇒ [PROPERTY_PCS_OK]
Presenza Numero [PN]	<ol style="list-style-type: none"> 1. Presente = false ⇒ [error] 2. Presente = true ⇒ [PROPERTY_PN_OK]
Presenza Lettera Maiuscola [LMAI]	<ol style="list-style-type: none"> 1. Presente = false ⇒ [error] 2. Presente = true ⇒ [PROPERTY_LMAI_OK]
Presenta Lettera Minuscola [LMIN]	<ol style="list-style-type: none"> 1. Presente = false ⇒ [error] 2. Presente = true ⇒ [PROPERTY_LMIN_OK]
Parametro: Conferma Password	
Nome categoria	Scelte per la categoria
Password Identica [PI]	<ol style="list-style-type: none"> 1. Conferma Password <> Password ⇒ [error] 2. Conferma Password == Password ⇒ [PROPERTY_PI_OK]



TEST CASE PLAN - OPENMEET

TEST FRAMES

Test Case ID	Test Frame	Esito
TC_3.1_1	PD1	Errato: L'email non presente nel database.
TC_3.1_2	PD2	Corretto: Il test case passa alla fase successiva.
TC_3.2_1	PD2, LN1	Errato: Lunghezza password invalida.
TC_3.2_2	PD2, LN2, PCS1	Errato: La password non contiene almeno un carattere speciale.
TC_3.2_3	PD2, LN2, PCS2, PN1	Errato: La password non contiene almeno un numero.
TC_3.2_4	PD2, LN2, PCS2, PN2, LMAI1	Errato: La password non contiene almeno una lettera maiuscola.
TC_3.2_5	PD2, LN2, PCS2, PN2, LMAI2, LMIN1	Errato: La password non contiene almeno una lettera minuscola.
TC_3.3_1	PD2, LN2, PCS2, PN2, LMAI2, LMIN2, PI1	Errato: Le due password non corrispondono.
TC_3.3_2	PD2, LN2, PCS2, PN2, LMAI2, LMIN2, PI2	Corretto.



TEST CASE PLAN - OPENMEET

9.2.3 Registrazione

Parametro: Email	
Nome categoria	Scelte per la categoria
Lunghezza Parte Locale [LNPL]	1. Lunghezza < 1 OR Lunghezza > 64 ⇒ [error] 2. Lunghezza >= 1 AND Lunghezza <= 64 ⇒ [PROPERTY_LNPL_OK]
Lunghezza Parte Dominio [LNPDP]	1. Lunghezza < 1 OR Lunghezza > 255 ⇒ [error] 2. Lunghezza >= 1 AND Lunghezza <= 255 ⇒ [PROPERTY_LNPDP_OK]
Presenza singola Chiocciola [PC]	1. Presente = false ⇒ [error] 2. Presente = true ⇒ [PROPERTY_PC_OK]
Legalità Punto Parte Locale [LPPL]	1. primoCarattere == '.' OR primoCarattere == '.' ⇒ [error] 2. primoCarattere <> '.' AND primoCarattere <> '.' ⇒ [PROPERTY_LPPL_OK]
Legalità Punto Parte Dominio [LPPDP]	1. primoCarattere == '.' OR primoCarattere == '.' ⇒ [error] 2. primoCarattere <> '.' AND primoCarattere <> '.' ⇒ [PROPERTY_LPPDP_OK]
Non-Consecutività Punti [NCP]	1. Non-Consecutività = false ⇒ [error] 2. Non-Consecutività = true ⇒ [PROPERTY_NCP_OK]
Legalità Caratteri Speciali Parte Locale [LCSPL]	1. carattereSpeciale not in "!#\$%&'*+./=?^_`{ }" ⇒ [error] 2. carattereSpeciale in "!#\$%&'*+./=?^_`{ }" ⇒ [PROPERTY_LCSPL_OK]
Legalità Caratteri Parte Dominio [LCPDP]	1. carattere not in [a-z] AND carattere not in [0-9] ⇒ [error] 2. carattere in [a-z] OR carattere in [0-9] ⇒ [PROPERTY_LCPDP_OK]
Assenza Numeri Dominio Primo Livello [ANDPL]	1. Assenza = false ⇒ [error] 2. Assenza = true ⇒ [PROPERTY_ANDPL_OK]
Presenza Database [PD]	1. Presente = true ⇒ [error] 2. Presente = false ⇒ [PROPERTY_PD_OK]



TEST CASE PLAN - OPENMEET

Parametro: Password	
Nome categoria	Scelte per la categoria
Lunghezza [LN]	1. Lunghezza < 8 OR Lunghezza > 16 ⇒ [error] 2. Lunghezza >= 8 AND Lunghezza <= 16 ⇒ [PROPERTY_LN_OK]
Presenza Carattere Speciale [PCS]	1. Presente = false ⇒ [error] 2. Presente = true ⇒ [PROPERTY_PCS_OK]
Presenza Numero [PN]	1. Presente = false ⇒ [error] 2. Presente = true ⇒ [PROPERTY_PN_OK]
Presenza Lettera Maiuscola [LMAI]	1. Presente = false ⇒ [error] 2. Presente = true ⇒ [PROPERTY_LMAI_OK]
Presenza Lettera Minuscola [LMIN]	1. Presente = false ⇒ [error] 2. Presente = true ⇒ [PROPERTY_LMIN_OK]
Parametro: Conferma Password	
Nome categoria	Scelte per la categoria
Password Identica [PI]	1. Conferma Password <> Password ⇒ [error] 2. Conferma Password == Password ⇒ [PROPERTY_PI_OK]



TEST CASE PLAN - OPENMEET

TEST FRAMES

Test Case ID	Test Frame	Esito
TC_4.1_1	LNPL1	Errato: Lunghezza parte locale invalida.
TC_4.2_1	LNPL2, LNPDP1	Errato: Lunghezza parte dominio invalida.
TC_4.3_1	LNPL2, LNPDP1, PC1	Errato: L'email presenta nessuna o più di una chiocciola.
TC_4.4_1	LNPL2, LNPDP1, PC2, LPPL1	Errato: Legalità punto della parte locale non rispettata.
TC_4.5_1	LNPL2, LNPDP1, PC2, LPPL2, LPPD1	Errato: Legalità punto della parte dominio non rispettata.
TC_4.6_1	LNPL2, LNPDP1, PC2, LPPL2, LPPD2, NCP1	Errato: Non-Consecutività dei punti non rispettata.
TC_4.7_1	LNPL2, LNPDP1, PC2, LPPL2, LPPD2, NCP2, LCSPL1	Errato: Legalità di caratteri speciali parte locale non rispettata.
TC_4.8_1	LNPL2, LNPDP1, PC2, LPPL2, LPPD2, NCP2, LCSPL2, LCPD1	Errato: Legalità di caratteri speciali parte dominio non rispettata.
TC_4.9_1	LNPL2, LNPDP1, PC2, LPPL2, LPPD2, NCP2, LCSPL2, LCPD2, ANDPL1	Errato: Assenza dei numeri nella parte del dominio non rispettata.
TC_4.10_1	LNPL2, LNPDP1, PC2, LPPL2, LPPD2, NCP2, LCSPL2, LCPD2, ANDPL2, PD1	Errato: L'email è già presente nel database.
TC_4.11_1	LNPL2, LNPDP1, PC2, LPPL2, LPPD2, NCP2, LCSPL2, LCPD2, ANDPL2, PD1, LN1	Errato: Lunghezza password invalida.
TC_4.12_1	LNPL2, LNPDP1, PC2, LPPL2, LPPD2, NCP2, LCSPL2, LCPD2, ANDPL2, PD1, LN2, PCS1	Errato: La password non contiene almeno un carattere speciale.
TC_4.13_1	LNPL2, LNPDP1, PC2, LPPL2, LPPD2, NCP2, LCSPL2, LCPD2, ANDPL2, PD1, LN2, PCS2, PN1	Errato: La password non contiene almeno un numero,
TC_4.14_1	LNPL2, LNPDP1, PC2, LPPL2, LPPD2, NCP2, LCSPL2, LCPD2, ANDPL2, PD1, LN2, PCS2, PN1, LMAI1	Errato: La password non contiene almeno una lettera maiuscola.



TEST CASE PLAN - OPENMEET

Test Case ID	Test Frame	Esito
TC_4.15_1	LNPL2, LNPD1, PC2, LPPL2, LPPD2, NCP2, LCSPL2, LCPD2, ANDPL2, PD1, LN2, PCS2, PN1, LMAI2, LMIN1	Errato: La password non contiene almeno una lettera minuscola.
TC_4.16_1	LNPL2, LNPD1, PC2, LPPL2, LPPD2, NCP2, LCSPL2, LCPD2, ANDPL2, PD1, LN2, PCS2, PN1, LMAI2, LMIN2, PI1	Errato: Le due password non corrispondono.
TC_4.16_2	LNPL2, LNPD1, PC2, LPPL2, LPPD2, NCP2, LCSPL2, LCPD2, ANDPL2, PD1, LN2, PCS2, PN1, LMAI2, LMIN2, PI2	Corretto



TEST CASE PLAN - OPENMEET

10. Pianificazione e Assegnazione ruoli

L'attività di pianificazione del testing avverrà dopo la fase di design.

Le scritture dei casi di test avverrà in contemporanea allo sviluppo del codice.

L'esecuzione dei test avverrà sia durante che dopo l'implementazione del sistema. Una volta concluso lo sviluppo, tutti i test saranno eseguiti nuovamente per garantirne il corretto funzionamento e produrre i report finali.

Tutti i soggetti coinvolti nella fase di testing saranno responsabili di sviluppare le classi di test a loro assegnate e di testare le classi di test sviluppate dagli altri.