|  |
| --- |
| **Università degli Studi di Salerno Corso di Ingegneria del Software** |

**LetsMeet  
TestPlan  
Versione 1.1**

****

**Docente:**

Andrea De Lucia

**Studenti:**

Vittorio Aiello

Gerardo Benevento

Raffaele Sansone

Data: 10/02/2019

|  |  |
| --- | --- |
| Progetto: LetsMeet | Versione: 1.0 |
| Documento: Test Plan | Data: 10/03/2019 |

**Coordinatore del progetto:**

|  |  |
| --- | --- |
| Nome | Matricola |
|  |  |
| Vittorio Aiello | 0512104584 |

**Partecipanti:**

|  |  |
| --- | --- |
| Nome | Matricola |
|  |  |
| Vittorio Aiello | 0512104524 |
| Gerardo Benevento | 0512104584 |
| Raffaele Sansone | 0512104974 |

|  |  |
| --- | --- |
| **Scritto da:** | Gerardo Benevento, Raffaele Sansone, Vittorio Aiello |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Versione | Descrizione | Autore |
| 10/02/2019 | 1.0 | Prima stesura del TetsPlan | GB; VA; RS |
| 24/03/2019 | 1.1 | Revisione TestPan | GB; VA; RS |

1. IntroduzioneLo scopo di questo documento è quello di analizzare e gestire lo sviluppo e le attività di  
Testing riguardanti il software LetsMeet.  
Questa sessione di lavoro deve verificare il corretto funzionamento di LetsMeet in diversi casi,  
studiati appositamente per mettere alla prova ogni singola funzionalità e caratteristica del sistema,  
al fine di verificare se esistono incongruenze tra il comportamento atteso e il comportamento  
osservato. Andremo quindi a rilevare gli eventuali errori prodotti all’interno del codice, per evitare  
che essi si presentino nel momento in cui il sistema verrà utilizzato dall’utente finale.  
Le attività di test sono state pianificate per le seguenti gestioni:  
● Gestione Account;  
● Gestione Eventi;  
I risultati di questi test saranno utilizzati per capire dove bisognerà intervenire, e quindi correggere eventuali errori o apportare modifiche per il miglioramento dei vari sottosistemi.

2. Documenti correlatiIl test plan ha una stretta relazione con il resto dei documenti che sono stati prodotti finora poiché il  
sistema è stato pianificato nelle precedenti documentazioni e implementato in parte. Nella fase di  
testing si verificheranno le eventuali somiglianze tra il sistema desiderato e quello proposto. Di  
seguito verranno indicate le relazioni con i precedenti documenti.

2.1 Relazioni con il documento di analisi deirequisiti (RAD)La relazione tra test plan e RAD (Requirement Analysis Document) riguarda in particolare i  
requisiti funzionali e non funzionali del sistema poiché i test verranno eseguiti su quelle  
funzionalità tenendo conto delle specifiche espresse nel precedente documento.  
In particolare il RAD contiene lo scopo del sistema, l’ambito del sistema e gli obiettivi,  
mostrando una panoramica di requisiti funzionali, requisiti non funzionali, scenari, casi  
d’uso, diagrammi e mockup del sistema.  
2.2 Relazioni con il System Design Document (SDD)L’SDD (System Design Document) rappresenta l’architettura del sistema suddiviso in tre  
layer: Presentation Layer, Application Layer e Storage Layer. Il test deve tenere conto di  
queste suddivisioni. In particolare l’SDD contiene l’architettura del software corrente e  
proposto e i servizi dei sottosistemi.  
2.3 Relazioni con l’Object Design Document (ODD)Il test si baserà sulle class Interfaces definite nell’ODD (Object Design Document). In  
particolare l’ODD contiene i package e i class interface del sistema.  
  
3. Panoramica del sistemaCome analizzato e deciso nel System Design Document la struttura del nostro sistema è divisa  
secondo un’architettura “Three Tiers” cioè a tre livelli: Presentation Layer, Application Layer,  
Storage Layer. Il livello più alto interagisce con il livello applicativo che a sua volta si occuperà di  
eseguire le operazioni nel database di LetsMeet, cercando di garantire il più possibile basso  
accoppiamento e alta coesione tra le varie classi. Il sistema inoltre è stato suddiviso in sottosistemi  
più piccoli, in particolare è stato diviso per gestioni. Abbiamo individuato i seguenti sottosistemi:  
● Gestione Account;  
● Gestione Eventi  
Quasi ognuna delle precedenti gestioni prevede principalmente operazioni di inserimento,  
modifica, rimozione, visualizzazione e ricerca che saranno testate nel corso della fase di testing  
del sistema.  
  
4. Funzionalità da testareDi seguito saranno elencate le funzionalità introdotte nel sistema dopo il greenfieldengeneering che saranno sottoposte a test divise per ogni gestione del sistema.  
● Gestione Account:

- Login

- Logout

- Registrazione

- Registrazione Moderatore

- Ricerca Utenti

- Visualizza Profilo

● Gestione Eventi:

- Visualizzazione Eventi

- Creazione Evento

- Rating Evento

- Partecipazione Evento

- Verifica Partecipazione Evento

- Ricerca Evento

- Visualizzazione Info Evento

- Scrivere un commento

- Visualizza Commenti

- Cancellazione Evento

5. Criteri Pass/FailedI dati di input del test saranno raggruppati in insiemi dalle caratteristiche comuni in modo da  
effettuare un test su di un unico elemento rappresentativo.  
Il testing ha successo se l’output osservato è diverso dall’output atteso: ciò significa che la fase di  
testing avrà successo se individuerà una failure. In tal caso questa verrà analizzata e, se legata ad  
un fault, si procederà alla sua correzione. Sarà infine iterata la fase di testing per verificare che la  
modifica non abbia impattato su altri componenti del sistema.  
La failure quindi è uno stato di condizione nel quale non si trova l’output desiderato, si può dire che  
è il contrario di un successo.  
6. ApproccioL’approccio alla fase di testing si compone di 3 fasi, la prima servirà a testare le componenti una  
ad una, poi si passerà a testare le funzionalità delle integrazioni dei vari sottosistemi, infine si  
testerà l’intero sistema assemblato per verificare soprattutto che esso soddisfi le richieste del  
cliente.  
6.1 Testing di unitàPer realizzare il testing di ogni singola componente verrà utilizzata la tecnica “Black-Box  
testing”.  
Così facendo andremo ad esaminare le funzionalità dell’applicazione ed il comportamento  
input/output delle singole componenti senza tener conto della loro struttura interna.  
🡪Essendo quasi impossibile generare tutti i possibili input, verranno create classi  
d’equivalenza scegliendo per ognuna un test case per ridurre la ridondanza e rendere il test  
più efficiente. I risultati del testing verranno analizzati e usati per correggere gli errori che  
causano il fallimento del sistema  
6.2 Testing di integrazioneDopo aver sottoposto ogni componente al testing di unità, ed aver corretto gli eventuali  
errori scaturiti dal test, essi verranno integrati in sottosistemi più grandi per sottoporli ad un  
test di integrazione. Il testing verrà effettuato seguendo la strategia 🡪“Sandwich Testing” che  
prevede la divisione del sistema in 3 layer: sopra al target, target, sotto al target. Questa  
suddivisione consente di combinare ed effettuare in parallelo il testing top-down e il testing  
bottom-up con lo scopo di integrare il target, più specificamente i test partono in parallelo  
da top e da bottom fino a convergere verso il target layer.  
6.3 Testing di sistemaPrima di essere pronto all’uso, il sistema affronterà l’ultima fase di testing, quello di  
sistema, per dimostrare che siano soddisfatti tutti i requisiti richiesti. Lo scopo di questa  
fase è testare le funzionalità più importanti, usate maggiormente e con maggior probabilità  
di fallimento. Trattandosi di un sistema web-based verrà utilizzato, per effettuare il testing di  
sistema, il tool Selenium. Selenium si occupa di simulare l’interazione con il sistema dal  
punto di vista dell’utente.  
7. Sospensione e ripresa7.1 Criteri di sospensioneLa fase di testing del sistema verrà sospesa quando si otterranno i risultati attesi in accordo  
con i tempi di sviluppo previsti, tenendo sempre conto dei costi dell’attività di testing.  
Questo processo verrà quindi portato avanti quanto più possibile nel tempo senza però  
rischiare di ritardare la consegna finale del progetto.  
7.2 Criteri di ripresaLa fase di testing potrà riprendere in seguito a modifiche o correzioni che generano errori o  
fallimenti, i test case verranno, quindi, sottoposti nuovamente al sistema assicurandosi così  
di aver risolto effettivamente il problema.  
8. Materiale per il testingGli strumenti necessari per l’attività di test sono un computer, su cui è installato un browser,  
con una connessione ad internet visto che il database del sistema è stato caricato in un  
dominio online.

9. Test Cases

9.1 Gestione Account

9.1.1 Login

|  |
| --- |
| **Parametro:** Username **Formato:** [A-Z a-z] |
| **Lunghezza[LUS]** | ● <2 and >80 [error] ● >=2 and <= 80 [property lunghezzaLNOK] |
| **Formato[FUS]** | ● Rispetta il formato [iflunghezzaLNOK] [propertyformatoFNOK, rispetta il formato [A-Z a-z]] ● Non rispetta il formato [iflunghezzaLNOK] [error] |

|  |
| --- |
| **Parametro:** Password **Formato:** [A-Z a-z 0-9] |
| **Lunghezza[LPS]** | ● <2 and >256 [error] ● >=2 and <= 256 [property lunghezzaLNOK] |
| **Formato[FPS]** | ● Rispetta il formato [iflunghezzaLNOK] [propertyformatoFNOK, rispetta il formato [A-Z a-z]] ● Non rispetta il formato [iflunghezzaLNOK] [error] |

**9.1.2 Registrazione**

|  |
| --- |
| **Parametro:** Username **Formato:** [A-Z a-z] |
| **Lunghezza[LUS]** | ● <2 and >80 [error] ● >=2 and <= 80 [property lunghezzaLNOK] |
| **Formato[FUS]** | ● Rispetta il formato [iflunghezzaLNOK] [propertyformatoFNOK, rispetta il formato [A-Z a-z]] ● Non rispetta il formato [iflunghezzaLNOK] [error] |

|  |
| --- |
| **Parametro:** Password **Formato:** [A-Z a-z 0-9] |
| **Lunghezza[LPS]** | ● <2 and >256 [error] ● >=2 and <= 256 [property lunghezzaLNOK] |
| **Formato[FPS]** | ● Rispetta il formato [iflunghezzaLNOK] [propertyformatoFNOK, rispetta il formato [A-Z a-z]] ● Non rispetta il formato [iflunghezzaLNOK] [error] |

|  |
| --- |
| **Parametro:** E-mail **Formato:** [A-Za-z0-9. \_%+-] @ [A-Za-z0-9. -]. [A-Z] |
| **Lunghezza[LE]** | ● <5 and >256 [error] ● >=5 and <= 256 [property lunghezzaLEOK] |
| **Formato[FE]** | ● Rispetta il formato [iflunghezzaLEOK] [propertyformatoFEOK, rispetta il formato [A-Za-z0-9. \_%+- ] @ [A-Za-z0-9. -]. [A-Z] |

**9.1.3 Registrazione Moderatore**

|  |
| --- |
| **Parametro:** Username **Formato:** [A-Z a-z] |
| **Lunghezza[LUS]** | ● <2 and >80 [error] ● >=2 and <= 80 [property lunghezzaLNOK] |
| **Formato[FUS]** | ● Rispetta il formato [iflunghezzaLNOK] [propertyformatoFNOK, rispetta il formato [A-Z a-z]] ● Non rispetta il formato [iflunghezzaLNOK] [error] |

**9.3.4 Ricerca Utenti**

|  |
| --- |
| **Parametro:** Username **Formato:** [A-Z a-z] |
| **Lunghezza[LUS]** | ● <2 and >80 [error] ● >=2 and <= 80 [property lunghezzaLNUOK] |
| **Formato[FUS]** | ● Rispetta il formato [iflunghezzaLNUOK] [propertyformatoFNUOK, rispetta il formato [A-Z a-z]] ● Non rispetta il formato [iflunghezzaLNUOK] [error] |

**9.2 Gestione Eventi**

**9.2.1 Creazione Evento**

|  |
| --- |
| **Parametro:** Nome Evento **Formato:** [A-Z a-z 0-9 ] |
| **Lunghezza[LNE]** | ● <2 and >100 [error] ● >=2 and <= 100 [property lunghezzaLNUOK] |
| **Formato[FNE]** | ● Rispetta il formato [iflunghezzaLNUOK] [propertyformatoFNUOK, rispetta il formato [A-Z a-z]] ● Non rispetta il formato [iflunghezzaLNUOK] [error] |

|  |
| --- |
| **Parametro:** Orario di inizio **Formato:** [dd/mm/yyyy] |
| **Lunghezza[LOI]** | ● dd<1 and dd>31, mm<1 and mm>12,yyyy<1900 and yyyy>9999 ● 1<=dd<=31, 1<=mm<=12, 1<=yyyy<=9999 [property lunghezza LDOK] |
| **Formato[FOI]** | ● Rispetta il formato [iflunghezzaLNOK] [propertyformatoFNOK, rispetta il formato [dd/mm/yyyy]] ● Non rispetta il formato [iflunghezzaLNOK] [error] |

|  |
| --- |
| **Parametro:** Orario di fine **Formato:** [dd/mm/yyyy] |
| **Lunghezza[LOF]** | ● dd<1 and dd>31, mm<1 and mm>12,yyyy<1900 and yyyy>9999 ● 1<=dd<=31, 1<=mm<=12, 1<=yyyy<=9999 [property lunghezza LDOK] |
| **Formato[FOF]** | ● Rispetta il formato [iflunghezzaLNOK] [propertyformatoFNOK, rispetta il formato [dd/mm/yyyy]] ● Non rispetta il formato [iflunghezzaLNOK] [error] |

|  |
| --- |
| **Parametro:** Descrizione Evento **Formato:** [A-Z a-z] |
| **Lunghezza[LDE]** | ● <2 and >256 [error] ● >=2 and <= 256 [property lunghezzaLNUOK] |
| **Formato[FDE]** | ● Rispetta il formato [iflunghezzaLNUOK] [propertyformatoFNUOK, rispetta il formato [A-Z a-z]] ● Non rispetta il formato [iflunghezzaLNUOK] [error] |

**9.2.2 Ricerca Evento**

|  |
| --- |
| **Parametro:** Nome Evento **Formato:** [A-Z a-z] |
| **Lunghezza[LUV]** | ● <2 and >256 [error] ● >=2 and <= 256 [property lunghezzaLNUOK] |
| **Formato[FNV]** | ● Rispetta il formato [iflunghezzaLNUOK] [propertyformatoFNUOK, rispetta il formato [A-Z a-z]] ● Non rispetta il formato [iflunghezzaLNUOK] [error] |

**9.2.3 Scrivere un commento**

|  |
| --- |
| **Parametro:** Contenuto Commento **Formato:** [A-Z a-z !?] |
| **Lunghezza[LSC]** | ● <2 and >256 [error] ● >=2 and <= 256 [property lunghezzaLNUOK] |
| **Formato[FSC]** | ● Rispetta il formato [iflunghezzaLNUOK] [propertyformatoFNUOK, rispetta il formato [A-Z a-z]] ● Non rispetta il formato [iflunghezzaLNUOK] [error] |

**10. Glossario**

* **LUS:** Lunghezza Username
* **FUS:** Formato Username
* **LPS:** Lunghezza Password
* **FPS:** Formato Password
* **LE:** Lunghezza Email
* **FE:** Lunghezza Email
* **LNE:** Nome Evento
* **FNE:** Formato Nome Evento
* **LOI:** Lunghezza Orario Inizio
* **FOI:** Formato Orario Inizio
* **LOF:** Lunghezza Orario Fine
* **FOF:** Formato Orario Fine
* **LDE:** Lunghezza Descrizione Evento
* **FDE:** Formato Descrizione Evento
* **LSC:** Lunghezza Commento
* **FSC:** Formato Commento