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

**LetsMeet  
Test Case Specification  
Versione 1.1**

****

**Docente:**

Andrea De Lucia

**Studenti:**

Vittorio Aiello

Gerardo Benevento

Raffaele Sansone

Data: 21/02/2019

|  |  |
| --- | --- |
| Progetto: LetsMeet | Versione: 1.0 |
| Documento: Test Case Specification | Data: 21/02/2019 |

**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 |
| 21/02/2019 | 1.0 | Prima stesura del TetsCaseSpecification | GB; VA; RS |
| 24/03/2019 | 1.1 | Revisione TestCaseSpecification | GB; VA; RS |

1. Descrizione del documento

Un test case è un insieme di input e di risultati attesi che servono a testare una componente per scoprirne gli errori ed i fallimenti.

Un test case è composto da 7 attributi:

1. **Nome**: univoco per distinguere i test case attuali dagli altri
2. **Percorso Test**: la path del file di test che verrà effettuato
3. **Input**: la classe di equivalenza testata
4. **Test Data**: i dati emessi al sistema
5. **Oracolo**: i dati o comportamenti attesi dal sistema
6. **Risultato Attuale**: il comportamento osservato in seguito all’esecuzione del test.
7. **Pass/Fail**: specifica se il test è fallito o meno.

Un buon modello di test dovrebbe avere il minor numero di associazioni, in quanto è possibile eseguire in parallelo test non correlati tra loro.

Seguiremo l’approccio Category Partition. I test case che presenteranno comportamenti comuni verranno raggruppati in classi di equivalenza per motivi di tempistiche e budget.

Quindi per ogni use case ci deve essere un test case e a sua volta per ogni use case dobbiamo individuare le classi di equivalenza.

2. Glossario

**Test Case Name**: la convenzione è Test\_[funzionalitàDaTestare]

**Test Case ID**: la convenzione è TC\_[N].[M], dove N è il numero del package (es. gestione recensione, gestione domande…) e M è il numero progressivo in funzione di N.

**Condizione di Entrata**: è la precondizione dello use case.

**Flusso degli Eventi**: flusso delle interazioni tra sistema e utente. Il flusso non è strettamente collegato allo use case.

**Test Data :** Bisogna specificare i dati effettivi che si usano e che vengono restituiti.

**Oracolo**: lo stato finale pronosticato della situazione.

**Risultato Attuale:** lo stato attuale della situazione.

**Pass/Fail:** stato del test.

1. Test Cases

1.1 Gestione Account

TC\_1.1 Test\_EffettuaLogin

L’utente è intenzionato ad effettuare il login sull’applicazione LetsMeet. Il sistema è in attesa della compilazione di tutti i dati che sono: e-mail e password. Se non vi sono errori, il sistema prosegue ad effettuare il login altrimenti procede con eventuali segnalazioni all’utente.

**AUTENTICAZIONE EFFETTUATA CON SUCCESSO**

|  |  |
| --- | --- |
| **Test Case Name** | Test\_EffettuaAutenticazione |
| **Test Case ID** | TC\_1.1\_01 |
| **Condizione di Entrata** | L’utente è registrato a LetsMeet. |
| **Flusso degli Eventi** | 1. L’utente accede alla pagina di login. 2. LetsMeet presenta un form in cui inserire:  * Username; * Password.  1. Clicca su login |
| **Test Data** | Username = username  Password = username45 |
| **Oracolo** | L’utente effettua autentificazione e accede alla Home Page di LetsMeet. |
| **Risultato Attuale** | L’utente effettua autentificazione e accede alla Home Page di LetsMeet. |
| **Pass/Fail** | Pass |

**AUTENTICAZIONE FALLITA E-MAIL NON PRESENTE NEL DATABASE**

|  |  |
| --- | --- |
| **Test Case Name** | Test\_EffettuaAutenticazione |
| **Test Case ID** | TC\_1.1\_02 |
| **Condizione di Entrata** | L’utente è registrato a LetsMeet. |
| **Flusso degli Eventi** | 1. L’utente accede alla pagina di login. 2. LetsMeet presenta un form in cui inserire:  * Username; * Password.  1. Clicca su login |
| **Test Data** | Username = username  Password = username45 |
| **Oracolo** | Il sistema mostra un messaggio di errore dove indica che la compilazione del campo è errata per mancato match di corrispondenza. |
| **Risultato Attuale** | Il sistema mostra un messaggio di errore dove indica che la compilazione del campo è errata per mancato match di corrispondenza. |
| **Pass/Fail** | Fail |

**CLASSE DI EQUIVALENZA AUTENTICAZIONE FALLITA CAMPO NON COMPILATO**

|  |  |
| --- | --- |
| **Test Case Name** | Test\_EffettuaAutenticazione |
| **Test Case ID** | TC\_1.1\_03 |
| **Condizione di Entrata** | L’utente è registrato a LetsMeet. |
| **Flusso degli Eventi** | 1. L’utente accede alla pagina di login. 2. LetsMeet presenta un form in cui inserire:  * Username; * Password.  1. Clicca su Login |
| **Test Data** | Username = username  Password = // (viene lasciato vuoto) |
| **Oracolo** | Il sistema mostra un messaggio di errore dove indica che la compilazione del campo è errata |
| **Risultato Attuale** | Il sistema mostra un messaggio di errore dove indica che la compilazione del campo è errata |
| **Pass/Fail** | Fail |

**AUTENTICAZIONE FALLITA PASSWORD NON CORRETTA**

|  |  |
| --- | --- |
| **Test Case Name** | Test\_EffettuaAutenticazione |
| **Test Case ID** | TC\_1.1\_04 |
| **Condizione di Entrata** | L’utente è registrato a LetsMeet. |
| **Flusso degli Eventi** | 1. L’utente accede alla pagina di login. 2. LetsMeet presenta un form in cui inserire:  * Username; * Password.  1. Clicca su Login |
| **Test Data** | Username = username  Password = ussername\_\_46 |
| **Oracolo** | Il sistema mostra un messaggio di errore dove indica che la compilazione del campo è errata in quanto non conforme alla struttura stabilita |
| **Risultato Attuale** | Il sistema mostra un messaggio di errore dove indica che la compilazione del campo è errata |
| **Pass/Fail** | Fail |

TC\_1.2 Test\_EffettuaRegistrazione

L’utente ha intenzione di registrarsi al sistema LetsMeet. Il sistema è in attesa della compilazione di tutti i campi del form e la loro sottomissione. L’utente inserisce i dati e il sistema salva i dati dell’utente.

**REGISTRAZIONE EFFETTUATA CON SUCCESSO.**

|  |  |
| --- | --- |
| **Test Case Name** | Test\_EffettuaRegistrazione |
| **Test Case ID** | TC\_1.2\_01 |
| **Condizione di Entrata** | L’utente non è registrato a LetsMeet. |
| **Flusso degli Eventi** | 1. L’utente accede alla pagina di registrazione. 2. LetsMeet presenta un form in cui inserire:  * Username; * Password; * E-mail;  1. Clicca su Registrati |
| **Test Data** | Username = username  Password = username45  E-mail = username@gmail.com |
| **Oracolo** | Il sistema mostra una schermata che la registrazione è avvenuta con successo e che per completarla è necessario che lo studente inserisca il codice di conferma inviata all’email. |
| **Risultato Attuale** | Il sistema mostra una schermata che la registrazione è avvenuta con successo e che per completarla è necessario che lo studente inserisca il codice di conferma inviata all’email. |
| **Pass/Fail** | Pass |

**CLASSE DI EQUIVALENZA REGISTRAZIONE FALLITA CAMPO MANCANTE**

|  |  |
| --- | --- |
| **Test Case Name** | Test\_EffettuaRegistrazione |
| **Test Case ID** | TC\_1.2\_02 |
| **Condizione di Entrata** | L’utente non è registrato a LetsMeet. |
| **Flusso degli Eventi** | 1. L’utente accede alla pagina di registrazione. 2. LetsMeet presenta un form in cui inserire:  * Username; * Password; * E-mail;  1. Clicca su Registrati |
| **Test Data** | Username = username  Password =// (viene lasciato vuoto)  E-mail = username@gmail.com |
| **Oracolo** | Il sistema mostra un messaggio di errore. |
| **Risultato Attuale** | Il sistema mostra un messaggio di errore indicante che almeno un campo del form non è stato compilato. |
| **Pass/Fail** | Fail |

**REGISTRAZIONE FALLITA DATI INSERITI NON CONFORMI ALL’ E-MAIL**

|  |  |
| --- | --- |
| **Test Case Name** | Test\_EffettuaRegistrazione |
| **Test Case ID** | TC\_1.2\_03 |
| **Condizione di Entrata** | L’utente non è registrato a LetsMeet. |
| **Flusso degli Eventi** | 1. L’utente accede alla pagina di registrazione. 2. LetsMeet presenta un form in cui inserire:  * Username; * Password; * E-mail;  1. Clicca su Registrati |
| **Test Data** | Username = username  Password =username45  E-mail = username@.com |
| **Oracolo** | Il sistema mostra un messaggio di errore. |
| **Risultato Attuale** | Il sistema mostra un messaggio di errore indicante che il campo e-mail non è inserito correttamente |
| **Pass/Fail** | Fail |

**REGISTRAZIONE FALLITA USERNAME/E-MAIL GIA’ IN USO**

|  |  |
| --- | --- |
| **Test Case Name** | Test\_EffettuaRegistrazione |
| **Test Case ID** | TC\_1.2\_03 |
| **Condizione di Entrata** | L’utente non è registrato a LetsMeet. |
| **Flusso degli Eventi** | 1. L’utente accede alla pagina di registrazione. 2. LetsMeet presenta un form in cui inserire:  * Username; * Password; * E-mail;  1. Clicca su Registrati |
| **Test Data** | Username = username  Password =username45  E-mail = username@alice.com |
| **Oracolo** | Il sistema mostra un messaggio di errore. |
| **Risultato Attuale** | Il sistema mostra un messaggio di errore indicante che il campo username / email è già utilizzato |
| **Pass/Fail** | Fail |

TC1.3 Test\_RegistrazioneModeratore

L’utente è interessato a registrare un Moderatore. Il sistema è in attesa che l’utente scriva un username. L’utente scrive un username e clicca su invio. L’utente inserisce i dati e il sistema salva i dati del Moderatore.

**REGISTRAZIONE MODERATORE EFFETTUATA CON SUCCESSO.**

|  |  |
| --- | --- |
| **Test Case Name** | Test\_RegistrazioneModeratore |
| **Test Case ID** | TC\_1.3\_01 |
| **Condizione di Entrata** | L’utente è un Super-Admin e ha fatto l’accesso nella sezione dedicata. |
| **Flusso degli Eventi** | 1. L’utente accede alla pagina di registrazione Moderatore dedicata. 2. LetsMeet presenta un form in cui inserire:  * Username;  1. Clicca su Registrata Moderatore |
| **Test Data** | Username = username |
| **Oracolo** | Il sistema mostra una schermata che la registrazione è avvenuta con successo. |
| **Risultato Attuale** | Il sistema mostra una schermata che la registrazione è avvenuta con successo. |
| **Pass/Fail** | Pass |

**CLASSE DI EQUIVALENZA REGISTRAZIONE FALLITA CAMPO MANCANTE**

|  |  |
| --- | --- |
| **Test Case Name** | Test\_RegistrazioneModeratore |
| **Test Case ID** | TC\_1.3\_02 |
| **Condizione di Entrata** | L’utente è un Super-Admin e ha fatto l’accesso nella sezione dedicata. |
| **Flusso degli Eventi** | 1. L’utente accede alla pagina di registrazione Moderatore dedicata. 2. LetsMeet presenta un form in cui inserire:  * Username;  1. Clicca su Registrata Moderatore |
| **Test Data** | Username = //(vuoto) |
| **Oracolo** | Il sistema mostra una schermata che la registrazione non è avvenuta con successo. |
| **Risultato Attuale** | Il sistema mostra una schermata che la registrazione non è avvenuta con successo comunicando la mancanza del riempimento del form. |
| **Pass/Fail** | Fail |

**REGISTRAZIONE FALLITA USERNAME NON PRESENTE NEL DATABASE**

|  |  |
| --- | --- |
| **Test Case Name** | Test\_RegistrazioneModeratore |
| **Test Case ID** | TC\_1.3\_03 |
| **Condizione di Entrata** | L’utente è un Super-Admin e ha fatto l’accesso nella sezione dedicata. |
| **Flusso degli Eventi** | 1. L’utente accede alla pagina di registrazione Moderatore dedicata. 2. LetsMeet presenta un form in cui inserire:  * Username;  1. Clicca su Registrata Moderatore |
| **Test Data** | Username = usernameee |
| **Oracolo** | Il sistema mostra una schermata che la registrazione non è avvenuta con successo. |
| **Risultato Attuale** | Il sistema mostra una schermata che la registrazione non è avvenuta con successo per mancato match. |
| **Pass/Fail** | Fail |

TC1.4 Test\_RimozioneModeratore

L’utente è interessato a rimuovere un Moderatore. Il sistema è in attesa che l’utente scriva un username. L’utente scrive un username e clicca su invio. L’utente inserisce i dati e il sistema rimuove i dati del Moderatore.

**RIMOZIONE MODERATORE EFFETTUATA CON SUCCESSO.**

|  |  |
| --- | --- |
| **Test Case Name** | Test\_RegistrazioneModeratore |
| **Test Case ID** | TC\_1.4\_01 |
| **Condizione di Entrata** | L’utente è un Super-Admin e ha fatto l’accesso nella sezione dedicata. |
| **Flusso degli Eventi** | 1. L’utente accede alla pagina di registrazione Moderatore dedicata. 2. LetsMeet presenta un form in cui inserire:  * Username;  1. Clicca su Registrata Moderatore |
| **Test Data** | Username = username |
| **Oracolo** | Il sistema mostra una schermata che la rimozione è avvenuta con successo. |
| **Risultato Attuale** | Il sistema mostra una schermata che la rimozione è avvenuta con successo. |
| **Pass/Fail** | Pass |

**CLASSE DI EQUIVALENZA RIMOZIONE FALLITA CAMPO MANCANTE**

|  |  |
| --- | --- |
| **Test Case Name** | Test\_RegistrazioneModeratore |
| **Test Case ID** | TC\_1.4\_02 |
| **Condizione di Entrata** | L’utente è un Super-Admin e ha fatto l’accesso nella sezione dedicata. |
| **Flusso degli Eventi** | 1. L’utente accede alla pagina di registrazione Moderatore dedicata. 2. LetsMeet presenta un form in cui inserire:  * Username;  1. Clicca su Registrata Moderatore |
| **Test Data** | Username = //(vuoto) |
| **Oracolo** | Il sistema mostra una schermata che la rimozione non è avvenuta con successo. |
| **Risultato Attuale** | Il sistema mostra una schermata che la rimozione non è avvenuta con successo comunicando la mancanza del riempimento del form. |
| **Pass/Fail** | Fail |

**RIMOZIONE FALLITA USERNAME NON PRESENTE NEL DATABASE**

|  |  |
| --- | --- |
| **Test Case Name** | Test\_RegistrazioneModeratore |
| **Test Case ID** | TC\_1.4\_03 |
| **Condizione di Entrata** | L’utente è un Super-Admin e ha fatto l’accesso nella sezione dedicata. |
| **Flusso degli Eventi** | 1. L’utente accede alla pagina di registrazione Moderatore dedicata. 2. LetsMeet presenta un form in cui inserire:  * Username;  1. Clicca su Registrata Moderatore |
| **Test Data** | Username = usernameee |
| **Oracolo** | Il sistema mostra una schermata che la rimozionenon è avvenuta con successo. |
| **Risultato Attuale** | Il sistema mostra una schermata che la rimozione non è avvenuta con successo per mancato match. |
| **Pass/Fail** | Fail |

TC1.5 Test\_RicercaUtente

L’utente è interessato a cercare un utente della piattaforma. Il sistema è in attesa che l’utente scriva un username. L’utente scrive un username e clicca su invio. L’utente visualizza le informazioni fornitogli dal sistema.

**RICERCA EFFETTUATA CON SUCCESSO.**

|  |  |
| --- | --- |
| **Test Case Name** | Test\_RicercaUtente |
| **Test Case ID** | TC\_1.5\_01 |
| **Condizione di Entrata** | L’utente ha fatto l’accesso alla piattaforma. |
| **Flusso degli Eventi** | 1. L’utente seleziona la barra di ricerca visibile nella pagina. 2. LetsMeet presenta un form in cui inserire:  * Username;  1. Clicca su Ricerca |
| **Test Data** | Username = username |
| **Oracolo** | Il sistema mostra una schermata con l’utente con l’username specificato. |
| **Risultato Attuale** | Il sistema mostra una schermata con l’utente con l’username specificato. |
| **Pass/Fail** | Pass |

**CLASSE DI EQUIVALENZA RICERCA FALLITA CAMPO MANCANTE**

|  |  |
| --- | --- |
| **Test Case Name** | Test\_RicercaUtente |
| **Test Case ID** | TC\_1.5\_02 |
| **Condizione di Entrata** | L’utente ha fatto l’accesso alla piattaforma. |
| **Flusso degli Eventi** | 1. L’utente seleziona la barra di ricerca visibile nella pagina. 2. LetsMeet presenta un form in cui inserire:  * Username;  1. Clicca su Ricerca |
| **Test Data** | Username =//(Vuoto) |
| **Oracolo** | Il sistema mostra una schermata di errore. |
| **Risultato Attuale** | Il sistema mostra una schermata vuota. |
| **Pass/Fail** | Fail |

**RICERCA FALLITA USERNAME NON PRESENTE NEL DATABASE**

|  |  |
| --- | --- |
| **Test Case Name** | Test\_RicercaUtente |
| **Test Case ID** | TC\_1.5\_03 |
| **Condizione di Entrata** | L’utente ha fatto l’accesso alla piattaforma. |
| **Flusso degli Eventi** | 1. L’utente seleziona la barra di ricerca visibile nella pagina. 2. LetsMeet presenta un form in cui inserire:  * Username;  1. Clicca su Ricerca |
| **Test Data** | Username =usernameee |
| **Oracolo** | Il sistema mostra una schermata di errore. |
| **Risultato Attuale** | Il sistema mostra una schermata vuota. |
| **Pass/Fail** | Fail |

TC2.1 Test\_CreazioneEvento

L’utente è interessato a creare un utente della piattaforma. Il sistema è in attesa che l’utente scriva: nome dell’evento, categoria, data e ora e descrizione. L’utente riempie i campi e clicca su invio. L’utente crea l’evento e lo visualizza sulla mappa.

**CREAZIONE DELL’EVENTO AVVENUTA CON SUCCESSO**

|  |  |
| --- | --- |
| **Test Case Name** | Test\_CreazioneEvento |
| **Test Case ID** | TC\_2.1\_01 |
| **Condizione di Entrata** | L’utente ha fatto l’accesso alla piattaforma. |
| **Flusso degli Eventi** | 1. L’utente seleziona un punto sulla mappa. 2. LetsMeet presenta i form in cui inserire:  * Nome dell’evento; * Posizione; * Tipologia; * Data e ora; * Descrizione;  1. Clicca su Crea |
| **Test Data** | Nome dell’evento: Calcetto a 5;  Posizione = “Via Astori 9, Firenze”  Tipologia = Sport  Data e ora = 23/01/2019 10:53  Descrizione = partita di calcio a 5; |
| **Oracolo** | Il sistema crea il nuovo evento. |
| **Risultato Attuale** | Il sistema mostra un pop-up di creazione dell’evento avvenuta, posiziona il marker sulla mappa nel luogo specificato nel campo “Posizione” e salva l’evento. |
| **Pass/Fail** | Pass |

**CLASSE DI EQUIVALENZA CREAZIONE EVENTO FALLITA CAMPO MANCANTE**

|  |  |
| --- | --- |
| **Test Case Name** | Test\_CreazioneEvento |
| **Test Case ID** | TC\_2.1\_02 |
| **Condizione di Entrata** | L’utente ha fatto l’accesso alla piattaforma. |
| **Flusso degli Eventi** | 1. L’utente seleziona un punto sulla mappa. 2. LetsMeet presenta i form in cui inserire:  * Nome dell’evento; * Posizione; * Tipologia; * Data e ora; * Descrizione;  1. Clicca su Crea |
| **Test Data** | Nome dell’evento: Calcetto a 5;  Posizione = “Via Astori 9, Firenze”  Tipologia = Sport  Data e ora = //(vuoto)  Descrizione = partita di calcio a 5; |
| **Oracolo** | Il sistema non crea l’evento. |
| **Risultato Attuale** | Il sistema mostra un pop-up di errore della creazione dell’evento. |
| **Pass/Fail** | Fail |

TC2.2 Test\_InvioCommento

L’utente è interessato a inviare un commento di un evento della piattaforma. Il sistema è in attesa che l’utente scriva un stringa . L’utente riempie il campo e clicca su invio. L’utente invia il commento.

**INVIO DI UN COMMENTO CON SUCCESSO**

|  |  |
| --- | --- |
| **Test Case Name** | Test\_InvioCommento |
| **Test Case ID** | TC\_2.2\_01 |
| **Condizione di Entrata** | L’utente ha fatto l’accesso alla piattaforma è entrato nella pagina di un evento. |
| **Flusso degli Eventi** | 1. LetsMeet presenta iL form in cui inserire:  * Il testo del commento.  1. Clicca su Invia, |
| **Test Data** | Testo del commento = “Evento davvero molto bello!” |
| **Oracolo** | Il sistema crea il commento |
| **Risultato Attuale** | Il sistema mostra il commento nella zona messaggi dell’evento. |
| **Pass/Fail** | Pass |

**CLASSE DI EQUIVALENZA INVIO COMMENTO FALLITOO CAMPO MANCANTE**

|  |  |
| --- | --- |
| **Test Case Name** | Test\_InvioCommento |
| **Test Case ID** | TC\_2.2\_02 |
| **Condizione di Entrata** | L’utente ha fatto l’accesso alla piattaforma è entrato nella pagina di un evento. |
| **Flusso degli Eventi** | 1. LetsMeet presenta iL form in cui inserire:  * Il testo del commento.  1. Clicca su Invia, |
| **Test Data** | Testo del commento = //(vuoto) |
| **Oracolo** | Il sistema non crea il commento. |
| **Risultato Attuale** | Il sistema non mostra il commento nella zona messaggi dell’evento. |
| **Pass/Fail** | Fail |

**//Completare per la ricerca evebto non so ancora bene i campi**