Testing Plan

Indice

A cura di:

- **Cauteruccio Luisa**
- **Pascucci Valentina**
- **Tortora Carlo**

1. Introduzione

L'obiettivo che si vuole raggiungere all'interno del documento di testing è quello di trovare le differenze fra il comportamento atteso e il comportamento osservato durante l'implementazione del sistema. Lo scopo dell'attività di testing è infatti quello di verificare che i requisiti raccolti durante la fase di analisi corrispondano alle aspettative del sistema dopo averlo implementato. Pertanto, bisogna ricercare le failures e i faults del sistema, correggerli, così da evitare che si ripresentino quando il sistema viene rilasciato. Il sistema, durante le fasi primarie dello sviluppo, è stato suddiviso in varie gestioni, non tutte sono state implementate e pertanto saranno testate le gestioni implementate.

2. Documenti correlati

Il documento di testing fa riferimento all'ultima fase dello sviluppo del software, pertanto i documenti precedenti sono considerati un "punto chiave" e saranno presi come riferimento per la stesura del documento del testing.

2.1 Relazione fra documento di testing e RAD

All'interno del "Requirement Analysis Document" il sistema viene suddiviso in gestioni: Gestione Autenticazione, Gestione Registrazione, Gestione Account, Gestione Annunci, Gestione Eventi, Amministrazione Piattaforma. Successivamente alla stesura del Rad, la gestione degli annunci non è stata implementata in quanto ad essa è stata associata una media priorità.

2.2 Relazione fra documento di testing e SDD

All'interno del documento del System Design, il sistema è stato suddiviso secondo un'architettura "Three-layer" nei seguenti livelli: Presentation Layer, Application Layer e Storage Layer. Il Presentation layer, dedicato alle interfacce utente, e l'Application layer, riguardante le funzionalità e le gestioni del sistema, sono stati implementati all'interno della piattaforma AndroidStudio e verranno testate nella medesima piattaforma. Lo sviluppo del server è stato implementato tramite la piattaforma Eclipse. All'interno di tale piattaforma è presente lo Storage layer, il livello che si interfaccia con il database tramite query che richiedono la restituzione, l'aggiornamento, l'inserimento e l'eliminazione dei dati. È stata implementata lato server la gestione amministrativa della piattaforma che include le visualizzazioni di utenti, eventi, proposte di evento da parte dell'amministratore che potrà procedere all'eliminazione di questi. Il formato scelto per il trasferimento di dati client-server è JSON, che permette la conversione dei risultati delle query in un formato accedibile lato client.

2.2 Relazione fra documento di testing e ODD

Nell'ODD sono state definite le interfacce delle classi, i sottosistemi e i moduli che verranno testati nell'integration testing rispetteranno tali interfacce.

3. Panoramica del sistema

Come già detto, il sistema è stato suddiviso in gestioni: Gestione Autenticazione, Gestione Registrazione, Gestione Account, Gestione Annunci, Gestione Eventi, Amministrazione Piattaforma. Nello specifico saranno descritte tali gestioni: Gestione Autenticazione gestisce il login che gli utenti, registrati sulla piattaforma, effettuano per utilizzarla; Gestione Registrazione gestisce la registrazione degli utenti; Gestione Account: comprende tutte le funzionalità relative all'account di un utente; Gestione Annunci: gestisce la pubblicazione degli annunci da parte di band e artisti; Gestione Eventi: comprende la creazione, la pubblicazione e la ricerca di proposte evento ed eventi. Di seguito verranno riportate le funzionalità che verranno testate, suddivise per gestioni.

4. Funzionalità testate e funzionalità non testate

Lato client:

- **Gestione Autenticazione:**

Login: metodo che permette ad utenti, nello specifico band, artisti e promotori, registrati sulla piattaforma, di effettuare l'accesso.

Logout: metodo che permette agli utenti di effettuare il logout dalla piattaforma.

- **Gestione Account:**

Visualizza Mio Profilo: tale metodo permette ad un utente di visualizzare il proprio profilo. Questo metodo è stato realizzato attraverso 3 metodi: **Visualizza Profilo Band** che permette ad una band di visualizzare il proprio profilo; **Visualizza Profilo Artista:** che permette ad un artista di visualizzare il proprio profilo**; Visualizza Profilo Promotore:** che permette ad un promotore di visualizzare il proprio profilo

Cerca: tale metodo permette ad un utente registrato e autenticato di visualizzare informazioni relative ad altri utenti iscritti alla piattaforma. Questo metodo è stato realizzato attraverso 3 metodi differenti: **Cerca Artista** che permette ad un utente di ricercare un'artista, **Cerca Band** che permette ad un utente di ricercare una band, **Cerca Promotore** che permette a un' artista e/o una band di ricercare un promotore.

- **Gestione Eventi:**

Cerca: tale metodo permette ad un utente registrato e autenticato di ricercare eventi e proposte di evento. Questo metodo è stato realizzato attraverso due metodi differenti: **Cerca Eventi** che permette a un utente di ricercare eventi pubblicati sulla piattaforma e **Cerca Proposte Evento** che permette ad artisti e/o band ricercare proposte di evento pubblicate dai promotori all'interno della piattaforma.

Lato server:

- **ClientListener:** la classe che permette di catalogare il tipo di risposta da inviare al Client in base alla richiesta elaborata dalla classe Connessione.

  Trova Metodo: metodo che permette di stabilire la tipologia di risposta che sarà inviata al Client

- **Connessione:** è la classe che permette di stabilire la connessione tra Client e Server. Si occupa delle richieste del Client, comunicando con il ClientListener soddisfa tali richieste.

5. Criteri Pass/Fail

Un test avrà "successo" se l'output osservato è diverso da quello atteso, in quanto significa che è stata riscontrata una failure, la quale verrà analizzata, se tale failure è legata ad un fault si provvederà alla correzione. Il test verrà considerato "fallito" se l'output atteso corrisponderà a quello osservato.

6. Approccio

Il testing procederà nel modo seguente: verrà eseguito il testing di unità che si focalizza sulle singole componenti del sistema, in seguito il testing di integrazione che si focalizza su insiemi di componenti del sistema al fine di verificare la correttezza delle interfacce, infine sarà eseguito il testing di sistema all'interno del quale verrà testato l'intero sistema implementato al fine di verificare che tutti i requisiti siano stati rispettati.

6.1 Testing di unità

Come definito in precedenza, il testing di unità prevede il test delle singole componenti del sistema al fine di verificare la corretta codifica delle componenti e l'esecuzione delle funzionalità intese.
Per eseguire il testing di unità lato server, sarà utilizzata la strategia Black-Box, in quanto saranno affrontati l'input e l'output di ciascun test, senza considerare particolari aspetti interni delle componenti.
Per eseguire il testing di unità lato client, saranno combinate le strategie Black-Box e White-Box insieme, in quanto con la tecnica Black-Box sono testate le funzionalità delle componenti, mentre con la tecnica White-Box saranno considerati gli aspetti strutturali e dinamici di tali componenti. Utilizzando infatti l'ambiente AndroidStudio, gli unit test saranno affrontati sotto forma di "Instrumented Test", poiché, oltre a testare delle singole funzionalità (come ad esempio la visualizzazione di un profilo utente), ci sarà bisogno di utilizzare l'API di AndroidStudio, in particolar modo le cosiddette View e le operazioni ad esse legate che permettono di legare una funzionalità al layout in cui è integrata. In particolare, per il testing lato client, si utilizza la strategia dell'Equivalence testing, grazie alla quale, per ciascuna classe di equivalenza, saranno selezionati sia un input valido che un input invalido per osservare il comportamento finale di ciascun test.

6.2 Testing di integrazione

Nel testing di integrazione vengono testati insiemi di componenti del sistema al fine di controllare le interfacce fra i vari sottosistemi.
Per i test di integrazione, è stata scelta la strategia di testing di integrazione verticale, tramite la quale le componenti saranno integrate in base alle differenti funzionalità offerte dal sistema implementato. Infatti, per un dato caso d'uso, le parti di cui avrà bisogno la componente per essere testata (come ad esempio l'interfaccia utente, la logica di bussiness, la parte relativa allo storage layer), saranno identificate e sviluppate in parallelo nel test di integrazione.

6.3 Testing di sistema

Per effettuare il test sull'intero sistema sarà incluso il Functional Testing, che troverà le differenze tra i requisiti funzionali e il sistema implementato. Ciascun test case sarà infatti derivato dal modello dei casi d'uso definito nel RAD, considerando i casi comuni e quelli relativi ad input errati. Le funzionalità del sistema sono definite nella sezione 2 del RAD che fa riferimento ai requisiti funzionali e nella sezione 3 realtiva ai modelli di sistema, quali scenari e casi d'uso.

Saranno inoltre inclusi vari aspetti relativi al Performance Testing, quali Stress Tests e Volume Tests.

La parte relativa al Performance Testing troverà le differenze tra i design goals e i requisiti non funzionali definiti nel SDD e nel RAD nei confronti del sistema implementato. Per quanto riguarda invece la fase di Volume Testing saranno effettuati test relativi allo stream di dati scambiati tra client e server.

7 Criteri di sospensione e ripresa

7.1 Criteri di sospensione

La fase di testing sarà sospesa quando si raggiungerà il giusto compromesso tra qualità del prodotto e i costi dell'attività.

7.2 Criteri di ripresa

Dopo aver eseguito un test, e aver corretto i relativi fault e failure, sarà effettuato un **Test di regressione** al fine di verificare che le correzioni non hanno generato ulteriori errori.

8 Materiale per il testing

L'hardware necessario per il testing richiede almeno due pc che durante l'utilizzo dovranno disporre della connessione ad Internet: uno per gestire il lato client, l'altro per gestire il lato server. I software necessari saranno AndroidStudio, Eclipse, MySQL. Durante la fase di testing con la piattaforma AndroidStudio sarà utilizzato un emulatore che simulerà l'interazione con l'applicazione sviluppata. Inoltre, per eseguire il testing lato client è stato utilizzato Espresso.

9 Test Case

T C 1. Login

TC 1.1 Category Partition: e-mail

Parametro	e-mail
Formato	/^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$/;
Categorie	
Lunghezza[LE]	**1.** LE <5 **OR** LE>50 **[error]** **2.** LE>=5 **OR** LE<50 **[property LE_OK]**
Formato[FE]	**1.** Non rispetta il formato **[if LE_OK]** **2.** Rispetta il formato **[if LE_OK] [property FE_OK]**
Esiste [EE]	**1.** Nessuna corrispondenza nel DB **[if LE_OK and FE_OK] [errore]** **2.** Corrispondenza nel DB **[if LE_OK and FE_OK] [propertyEE_OK]**

TC 1.2 Category Partition: password

Parametro	Password
Formato	[a-z, A-Z,0-9,._%-]{5,20}
Categorie	
Lunghezza[LE]	**1.** LE<5 **[error]** **2.** LE>=5 and LE<32 **[property LE_OK]** **3.** LE>20 **[error]**
Formato[FE]	**1.** Non rispetta il formato **[if LE_OK]** **2.** Rispetta il formato **[if LE_OK][property FE_OK]**

Esiste[EE]	1. Nessuna corrispondenza nel DB **[if LE_OK and FE_OK] [errore]** 2. Corrispondenza nel DB **[if LE_OK and FE_OK] [propertyEE_OK]**

Test case: login

Codice	Combinazione	Esito
TC 1.1_1	**LE1**	**Errato**
TC 1.1_2	**LE2, FE1**	**Errato**
TC 1.1_3	**LE2, FE2, EE1**	**Errato**
TC 1.1_4	**LE2, FE2, EE2**	**Corretto**
TC 1.2_1	**LE1**	**Errato**
TC 1.2_2	**LE2, FE1,**	**Errato**
TC 1.2_3	**LE2, FE2, EE1**	**Errato**
TC 1.2_4	**LE2, FE2, EE2**	**Corretto**
TC 1.2_5	**LE3**	**Errato**

In definitiva, il login viene effettuato correttamente in TC 1.1_4 e TC 1.2_4

T C 2: Visualizza utenti

Codice	Combinazione	Esito

T C 2_1		**Corretto**

T C 3 : Visualizza lista eventi

Codice	Combinazione	Esito
T C 3_1		**Corretto**

T C 4: Visualizza lista proposte evento

Codice	Combinazione	Esito
T C 4_1		**Corretto**

T C 5: Visualizza mie proposte evento

Codice	Combinazione	Esito
T C 5_1		**Corretto**

T C 6: Visualizza miei eventi

Codice	Combinazione	Esito
T C 6_1		**Corretto**

T C 7: Cerca artisti

Codice	Combinazione	Esito
T C 7_1		**Corretto**

T C 8: Cerca band

Codice	Combinazione	Esito
T C 8_1		**Corretto**

T C 9: Cerca Promotore

Codice	Combinazione	Esito
T C 9_1		**Corretto**

T C 10: Cerca Eventi

Codice	Combinazione	Esito
T C 10_1		**Corretto**

T C 11: Cerca Proposte Evento

Codice	Combinazione	Esito
T C 11_1		**Corretto**

T C 12: Logout

Codice	Combinazione	Esito
T C 12_1		**Corretto**

Test case specification: questa sezione è dedicata ad ogni singolo test, vengono definiti per ogni test gli input e gli output attesi.

Test case specification: Login

Test Case ID	T C 1.1_1
Pre-Condition	L'utente visualizza la schermata contenente il form per effettuare il login
Flow of event	L'utente inserisce i dati richiesti **Input** | **Valore** Email | ds L'utente preme sul pulsante "Accedi"
Oracle	La mail inserita non rispetta la lunghezza minima pertanto, l'utente non può effettuare il login.

Test Case ID	T C 1.1_2
Pre-Condition	L'utente visualizza la schermata contenente il form per effettuare il login
Flow of event	L'utente inserisce i dati richiesti

	Input | **Valore** Email | .,.!!! L'utente preme sul pulsante "Accedi"
Oracle	La mail inserita rispetta la lunghezza prevista ma il formato non è corretto pertanto, l'utente non può effettuare il login.
Test Case ID	T C 1.1_3
Pre-Condition	L'utente visualizza la schermata contenente il form per effettuare il login
Flow of event	L'utente inserisce i dati richiesti **Input** | **Valore** Email | Luilids@dsdsa.com L'utente preme sul pulsante "Accedi"
Oracle	La mail inserita rispetta la lunghezza prevista ma il formato è corretto, ma la mail non risulta presente nel database pertanto, l'utente non può effettuare il login.

Test Case ID	T C 1.1_4
Pre-Condition	L'utente visualizza la schermata contenente il form per effettuare il login
Flow of event	L'utente inserisce i dati richiesti **Input** | **Valore** Email | Luilids@dsdsa.com L'utente preme sul pulsante "Accedi"
Oracle	La mail inserita rispetta la lunghezza prevista ma il formato è corretto e la mail è presente nel database.

Test Case ID	T C 1.2_1
Pre-Condition	L'utente visualizza la schermata contenente il

<table>
<tr><td></td><td>form per effettuare il login</td></tr>
<tr><td>Flow of event</td><td>L’utente inserisce i dati richiesti

<table><tr><th>Input</th><th>Valore</th></tr><tr><td>Email</td><td></td></tr><tr><td>Password</td><td>Y</td></tr></table>
L’utente preme sul pulsante “Accedi”</td></tr>
<tr><td>Oracle</td><td>La password inserita è troppo corta rispetto alla lunghezza minima prevista, pertanto l’utente non può effettuare il login .</td></tr>
</table>

<table>
<tr><td>Test Case ID</td><td>T C 1.2_2</td></tr>
<tr><td>Pre-Condition</td><td>L’utente visualizza la schermata contenente il form per effettuare il login</td></tr>
<tr><td>Flow of event</td><td>L’utente inserisce i dati richiesti

<table><tr><th>Input</th><th>Valore</th></tr><tr><td>Email</td><td></td></tr><tr><td>Password</td><td>Yyyyyyy</td></tr></table>
L’utente preme sul pulsante “Accedi”</td></tr>
<tr><td>Oracle</td><td>La password inserita rispetta la lunghezza prevista, ma il formato è errato, pertanto l’utente non può effettuare il login.</td></tr>
</table>

<table>
<tr><td>Test Case ID</td><td>T C 1.2_3</td></tr>
<tr><td>Pre-Condition</td><td>L’utente visualizza la schermata contenente il form per effettuare il login</td></tr>
<tr><td>Flow of event</td><td>L’utente inserisce i dati richiesti

<table><tr><th>Input</th><th>Valore</th></tr><tr><td>Email</td><td></td></tr><tr><td>Password</td><td></td></tr></table>
L’utente preme sul pulsante “Accedi”</td></tr>
<tr><td>Oracle</td><td>La password inserita rispetta la lunghezza prevista, il formato è corretto, ma la password</td></tr>
</table>

	non è presente nel database, pertanto l'utente non può effettuare il login

Test Case ID	T C 1.2_4
Pre-Condition	L'utente visualizza la schermata contenente il form per effettuare il login
Flow of event	L'utente inserisce i dati richiesti **Input** / **Valore** Email / Giovanni123@gmail.com Password / Cang_uro13 L'utente preme sul pulsante "Accedi"
Oracle	La password inserita rispetta la lunghezza prevista, il formato è corretto, e la password risulta presente nel database. L'utente effettua il login

Test Case ID	T C 1.2_5
Pre-Condition	L'utente visualizza la schermata contenente il form per effettuare il login
Flow of event	L'utente inserisce i dati richiesti **Input** / **Valore** Email / Password / Gfgjfdgfjdghfdghfdgjkfdhgfdhgfdkj L'utente preme sul pulsante "Accedi"
Oracle	La password inserita supera la lunghezza prevista. L'utente non può effettuare il login