Università degli Studi di Salerno
Corso di Ingegneria del Software

PlanetVideo
Test Plan Document
Versione 1.0

Data: 06/04/2020

Coordinatore del progetto:

Nome	Matricola

Partecipanti:

Nome	Matricola
Contente Antonio	0512103870
Mecca Vincenzo	0512103528

Scritto da:	Contente Antonio Mecca Vincenzo

Revision History

Data	Versione	Descrizione	Autore
06/04/2020	1.0	Stesura documento	Contente Antonio Mecca Vincenzo

Indice

1. INTRODUZIONE

In questo documento verranno definiti gli approcci e le attività di testing riguardanti la piattaforma web PlanetVideo. Verranno identificati gli elementi da testare e da non testare, gli approcci e gli strumenti usati per l'attività di testing.
Lo scopo del testing è quello di verificare il corretto funzionamento del nostro sistema in diversi casi, studiati per mettere alla prova le varie funzionalità del software. Effettuando tali test saremo in grado di rivelare errori, bug o incongruenze tra il comportamento desiderato e quello effettivo.
Questa attività ci consentirà di scoprire degli errori prima della messa in esercizio del sistema; così facendo, potremo porre rimedio a questi errori e fare in modo che non si rivelino durante la messa in esercizio del sistema.
Le funzionalità che andremo ad analizzare sono le seguenti:

- Gestione autenticazione
- Gestione catalogo
- Gestione recensioni
- Gestione account

2. DOCUMENTI CORRELATI

Il presente documento è in stretta relazione con i documenti che son stati prodotti finora, in cui il sistema è stato pianificato e delineato.
Nella fase di testing, si verificherà che le aspettative descritte nei documenti precedentemente prodotti, siano in larga parte rispettate.

I test case sono basati sulle funzionalità individuate nel documento di raccolta ed analisi dei requisiti.

2.1. Relazione con il documento di raccolta e analisi dei requisiti (RAD)

La relazione tra Test Plan e RAD riguarda in particolare i requisiti funzionali e non funzionali del sistema: i test verranno eseguiti su quelle funzionalità, tenendo conto delle specifiche espresse nel precedente documento.
In particolare, nel RAD viene descritto lo scopo, l'ambito e gli obiettivi del sistema, mostrando una panoramica di requisiti funzionali, requisiti non funzionali, scenari, casi d'uso, diagrammi e mockup del sistema.

2.2. Relazione con il System Design Document (SDD)

Nell' SDD è presente l'architettura del sistema (MVC), la struttura dei dati e i servizi dei sottosistemi.

2.3.Relazione con l'Object Design Document (ODD)

Nell'ODD sono identificati i package e le class interfaces del sistema che andranno prese in considerazione durante la fase di testing.

3. PANORAMICA DEL SISTEMA

Come definito nel System Design Document, la struttura del nostro sistema segue l'architettura MVC (Model View Controller).

La componente fondamentale di questa architettura è il controller, che si occuperà della logica di business della specifica funzionalità.

Nel model verranno mappate le entità persistenti sul DB come oggetti.

La view si occuperà di mostrare le interfacce utente.

Sono stati individuati i seguenti sottosistemi:

- Gestione autenticazione
- Gestione catalogo
- Gestione recensioni
- Gestione account

4. FUNZIONALITÀ DA TESTARE

Di seguito, sono elencati i requisiti da testare per ogni gestione:

- Gestione autenticazione
    - Login utente
    - Login amministratore
    - Registrazione
- Gestione catalogo
    - Visualizzazione catalogo
    - Visualizzazione dettagli
    - Aggiunta alla lista preferiti
    - Rimozione dalla lista preferiti
    - Aggiunta contenuto al catalogo
    - Rimozione contenuto dal catalogo
- Gestione recensioni
    - Aggiunta recensioni
- Gestione account
    - Visualizzazione dati account
    - Visualizzazione dati abbonamento
    - Visualizzazione fatture
    - Rinnovo abbonamento

5. APPROCCIO

5.1. Testing di Unità

In questa fase andremo a testare ogni singola funzione degli oggetti creati. Questa rappresenterà la nostra unità.

Verrà utilizzato un approccio black-box, ovvero non sarà basato sulla conoscenza dell'architettura e del funzionamento interno di un componente ma sulle sue funzionalità esternamente esposte.

Per tale fase utilizzeremo JUnit, framework di unit testing per il linguaggio Java, al quale abbineremo Mockito.

5.2. Testing di Integrazione

In questa fase le singole unità vengono combinate e testate come gruppo.

Per poter effettuare l'integration test è stata scelta la strategia bottom-up, in quanto consente di poter iniziare l'attività di testing non appena il primo modulo è stato specificato.

La riusabilità del codice è uno dei principali benefici dell'approccio bottom-up. Nonostante questa strategia di testing di integrazione abbia alcune limitazioni, risulta essere la più semplice e naturale forma con cui eseguire questo tipo di testing.

5.3. Testing di Sistema

Prima della messa in esercizio del sistema, verrà effettuata una fase di testing di sistema per dimostrare che i requisiti richiesti siano soddisfatti.
In questo tipo di testing andremo a testare le funzionalità più importanti e usate maggiormente dall'utente.

Trattandosi di una applicazione web, verrà utilizzato il tool Selenium, che si occuperà di simulare l'interazione con il sistema dal punto di vista dell'utente.

6. CRITERI DI PASS/FAIL

Una volta individuati i vari dati di input del test, questi verranno raggruppati in base a caratteristiche comuni in insiemi. In questo modo ci sarà possibile diminuire ed ottimizzare il numero di test.

La fase di test avrà successo se individuerà una failure, cioè se l'output osservato sarà diverso da quello atteso. Ogni qual volta verrà individuata una failure, questa verrà analizzata e, se legata ad un fault, si procederà alla sua correzione.
Una volta completata la correzione, si procederà, in modo iterativo, ad una nuova fase di test per

verificare che la modifica non ha impattato su altri componenti del sistema.

Al contrario, il testing fallirà se l'output osservato sarà uguale all'oracolo.

7. CRITERI DI SOSPENSIONE E RIPRESA

7.1. Criteri di sospensione

La fase di testing verrà interrotta quando verranno raggiunti i risultati attesi in accordo con in tempi e i costi allocati alle attività di testing.

7.2. Criteri di ripresa

Le attività di testing riprenderanno in seguito a delle modifiche che potrebbero introdurre nuovi errori all'interno del sistema.

8. MATERIALE PER IL TESTING

Gli strumenti necessari per l'attività di test sono un computer con una connessione ad Internet, su cui è installato un browser, dato che il nostro progetto è interamente web based.
Inoltre è necessario un IDE per Java EE e l'utilizzo di Selenium.