Università degli studi di Salerno

***Corso di Laurea in Informatica***

******

***PROGETTO DI INGEGNERIA DEL SOFTWARE – Test Plan***

***“BUY & SEE”***

**Docente:**

Andrea De Lucia

# Studenti:

##### Nome Matricola

## Fabio Curci 0512103296

## Francesca Di Mauro 05121

## Francesco Ciampa 05121

## Anna Santoro 05121

**Indice**

1. Introduzione
2. Documenti correlati

2.1 Relazioni con il documento di analisi dei requisiti (RAD)

2.2 Relazioni con il System Design Document (SDD)

2.3 Relazioni con l’Object Design Document (ODD)

1. Panoramica del sistema
2. Funzionalità da testare
3. Criteri Pass\Failed
4. Approccio

6.1 Testing di unità

6.2 Testing di integrazione

6.3 Testing di sistema

7. Sospensione e ripresa

7.1 Criteri di sospensione

7.2 Criteri di ripresa

1. Materiale per il testing
2. Test Cases

9.1 Gestione Account

9.2 Gestione Registrazione

9.3 Gestione Prodotti

9.4 Gestione Store

10. Glossario

**1****. Introduzione**

Lo scopo di questo documento è quello di analizzare e gestire lo sviluppo e le attività di Testing riguardanti il software Buy&See. Questa sessione di lavoro deve verificare il corretto funzionamento di Buy&See 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 Autenticazione
* Gestione Catalogo
* Gestione Account
* Gestione Visione

**2.** **Documenti correlati**

Il test plan è strettamente collegato ai documenti prodotti fino ad ora, poiché prima di passare alla fase di testing abbiamo bisogno di avere una gran parte delle funzioni previste già implementate, queste sono state definite nei precedenti documenti. Di seguito verranno descritte le relazioni che ci sono fra il test plan e gli altri documenti

**2.1 Relazioni con il documento di analisi dei requisiti (RAD)**

La relazioni tra test plan RAD riguarda in particolare i requisiti funzionali e non funzionali del sistema poiché i test che saranno eseguiti su ogni funzionalità Terranno conto delle specifiche espresse nel RAD

* 1. **Relazioni con il System Design Document (SDD)**

L’SDD (System Design Document) rappresenta l’architettura del sistema suddiviso in tre layer: Model, View e Controller. 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 sistema**

Come descritto nel System Design Document (SDD) il nostro sistema è costruito da una architettura MVC suddivisa in tre componenti, al fine di garantire basso accoppiamento e alta coesione fra le classi. Le tre componenti della architettura MVC sono **Model, View e Controller**. Le tre componenti contengono rispettivamente lo strato di visualizzazione, lo strato di businness e lo strato di logica applicativa. Il sistema Buy&See è stato diviso ulteriormente in sottosistemi più piccoli, in particolare è stato diviso per gestioni. Le gestioni che abbiamo individuato prevedono principalmente metodi che effettuano operazioni CRUD(Create, Read, Update e Delete) e saranno queste le funzionalità che andranno testate.

**4. Funzionalità da testare/non testare**

Gestione Autenticazione

* Registrazione
* Login

Gestione Catalogo

* Rimuovi film dal catalogo
* Aggiorna sezione novità

Gestione Account

* Modifica dati personali
* Invia Segnalazione

Gestione visione contenuti

* Acquisto film
* Visione film

Le restanti funzionalità non verranno testate.

**5. Criteri Pass/Field**

I dati di input del test saranno suddivisi in classi di equivalenza, ovvero saranno raggruppati in insiemi con caratteristiche comuni, in modo tale da poter testare solo uno degli elementi della classe. L’input avrà superato il test se l’output risultante è quello atteso, il risultato atteso sarà quello specificato dal membro del team che si occuperà del testing su tale test case.

**6. Approccio**

L’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 unità**

Durante questa fase verranno ricercate le condizioni di fallimento, isolando i componenti. La strategia usata per il testing è la tecnica Black-Box, che si concentra sul comportamento Input/Output ignorando la struttura interna della componente. Per minimizzare il numero di test cases i possibili input verranno partizionati in classi di equivalenza e per ogni classe verrà usato un test case. Gli errori scovati in questa fase, e anche in tutte le altre, devono essere comunicati agli sviluppatori in modo tale che questi possano correggere l’errore e ripristinare la fase di testing al più presto. I fallimenti identificati durante la fase di testing verranno specificati nel test incidenti report.

**6.2 Testing di integrazione**

Questa fase ha lo scopo di integrare tutte le componenti di una funzionalità al fine di testarle nel complesso utilizzando una **strategia Bottom-up**. L’approccio Big Bang mira a testare prima le componenti individualmente e poi tutte insieme, come un unico sistema.

**6.3 Testing di sistema**

In questa ultima fase vogliamo dimostrare che il sistema soddisfi tutti i requisiti richiesti. Si cercherà di testare le funzionalità più utili per l’utente e quelle che hanno una maggiore probabilità di fallimento.

**8. Test Case**

**Gestione Autenticazione**

* Registrazione

Category Partition

|  |  |
| --- | --- |
| Parametro: Nome  Formato: ^[a-zA-Z]{3,20}$ | |
| Lunghezza |  |
| Formato |  |
|  |  |

|  |  |
| --- | --- |
| Parametro: Cognome  Formato: ^[a-zA-Z]{3,20}$ | |
| Lunghezza |  |
| Formato |  |
|  |  |

|  |  |
| --- | --- |
| Parametro: Username  Formato: ^[A-Za-z0-9.\_]{4,10}$ | |
| Lunghezza |  |
| Formato |  |
|  |  |

|  |  |
| --- | --- |
| Parametro: Email  Formato: [^@\s]+@[^\.\s]+\.[a-zA-Z]+$ | |
| Lunghezza |  |
| Formato |  |
|  |  |

|  |  |
| --- | --- |
| Parametro: Password  Formato: ^(\S){5,15}$ | |
| Lunghezza |  |
| Formato |  |
|  |  |

**Gestione Autenticazione**

* Login

Category Partition

|  |  |
| --- | --- |
| Parametro: Username  Formato: ^[A-Za-z0-9.\_]{4,10}$ | |
| Lunghezza |  |
| Formato |  |
|  |  |

|  |  |
| --- | --- |
| Parametro: Password  Formato: ^(\S){5,15}$ | |
| Lunghezza |  |
| Formato |  |
|  |  |

**Gestione Visione Contenuti**

* Acquisto film

Category Partition

|  |  |
| --- | --- |
| Parametro: Nome intestatario  Formato: | |
| Lunghezza |  |
| Formato |  |
|  |  |

|  |  |
| --- | --- |
| Parametro: Cognome intestatario  Formato: | |
| Lunghezza |  |
| Formato |  |
|  |  |

|  |  |
| --- | --- |
| Parametro: Numero carta  Formato: | |
| Lunghezza |  |
| Formato |  |
|  |  |

|  |  |
| --- | --- |
| Parametro: Codice di sicurezza  Formato: | |
| Lunghezza |  |
| Formato |  |
|  |  |