Test Plan Document

US UniSeat

Riferimento	
Versione	1.0
Data	18/12/2019
Destinatario	Prof.ssa Ferrucci Filomena
Presentato da	De Caro Antonio, De Santis Marco, Spinelli Gianluca, Capozzoli Lorenzo, Rocco Simone Pasquale
Approvato da	

Revision History

Data	Versione	Cambiamenti	Autori
13/12/2019	0.1	Prima stesura documento	Rocco Simone Pasquale, Spinelli Gianluca, De Santis Marco
15/12/2019	0.2	Introduzione capitoli 2 e 3	Spinelli Gianluca, De Santis Marco
16/12/2019	1.0	Revisione documento	De Caro Antonio, Spinelli Gianluca, De Santis Marco

Sommario

1.Introduzione

In questo documento vengo definite le tipologie di testing e i criteri riguardanti la piattaforma UniSeat. Saranno identificati gli elementi da testare e non, le tecniche e gli eventuali tool da utilizzare.

1.1 Definizioni, Acronimi e Abbreviazioni

1.1.1 Definizioni

- Branch Coverage: indica i branch testati all'interno del software;
- Fault: causa di un failure che può essere di tipo meccanico o algoritmico, nel caso algoritmico dovuto a errori nella fase di implementazione;
- Failure: il sistema si comporta in modo diverso da quello specificato;
- Model View Control (MVC): Model-view-controller, è un pattern architetturale molto diffuso nello sviluppo di sistemi software, in particolare nell'ambito della programmazione orientata agli oggetti, in grado di separare la logica di presentazione dei dati dalla logica di business;

1.2 Riferimenti

Libro:

-- Object-Oriented Software Engineering (Using UML, Patterns, and Java) Third Edition

Autori:

-- Bernd Bruegge

-- Allen H. Dutoit

Documenti:

-- US_RAD_V1.5.pdf – Requirements Analysis Document

-- US_SDD_V1.2.pdf – System Design Document

-- US_ODD_V1.2.pdf – Object Design Document

2.Possibili rischi

ID	Rischio	Risvolto	Probabilità	Impatto
R1	Organizzazione del testing non adeguata	Negativo	Bassa	Alto
R2	Utilizzo di tools appena conosciuti	Negativo/Positivo	Alta	Medio

3.Funzionalità da testare

3.1. Componenti testate

Verranno testare le funzionalità a priorità alta e a priorità media.

Quelli a priorità alta sono:

- Login;
- LogOut;
- VisualizzazionePosti;
- VisualizzaAule;
- VisualizzaEdifici;
- Registrazione;
- PrenotaazionePosto;
- EliminazionePrenotazione;
- VisualizzaPrenotazionePosto;
- PrenotazioneAula;
- DisdettaPrenotazioneAula;
- VisualizzaPrenotazioniAule;
- InserimentoAula;

Quelli a priorità media sono:

- ModificaDatiPersonali;
- VisualizzaListaPrenotazioni;
- VisualizzazioneUtentiRegistrati;
- ModificaAule;
- RimozioneUtente;
- InvioAutomaticoEmailConferma;
- NotificaAutomaticaPrenotazione;

4.Funzionalità da non testare

Le funzionalità a priorità bassa non verranno testate.

5.Approccio

L'obiettivo è quello di trovare i bug all'interno del sistema e correggerli. L'attività di testing avrà un esito positivo, se esiste almeno un input per un determinato test case per il quale l'output risultante non corrisponderà con l'oracolo. L'attività di testing può essere divisa in tre fasi:

- **Testing di unità:** rappresenta l'attività con cui è possibile testare una singola unità di sistema, si testa ogni metodo direttamente. La strategia usata per il testing è di tipo Black-Box, che si concentra sul comportamento input/output ignorando la struttura interna della componente. Per minimizzare il numero di test case, i possibili input verranno partizionati secondo il metodo del category partition. Il tool utilizzato per il testing è JUnit, il criterio di accettazione scelto per considerare il testing 'superato con successo' è di 75% di coverage.
- **Testing di integrazione**: ha lo scopo di integrare tutte le componenti di una funzionalità per poter verificare il corretto funzionamento. La strategia di testing utilizzata è il Bottom-up, che consiste nel testare prima i livelli più bassi, creando test driver e consente di ridurre le dipendenze fra le funzionalità dipendenze e facilitare la ricerca di errori nelle interfacce di comunicazione tra sottosistemi. Il tool utilizzato è Travis CI.
- **Testing di sistema**: il sistema deve soddisfare tutti i requisiti richiesti. Il tool utilizzato per il testing di sistema è Katalon. Verranno testate:
    - Funzionalità;