Test Case Integration Document

US UniSeat

Riferimento	
Versione	1.0
Data	13/12/2019
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

Il testing rileva la presenza di errori nel sistema e permette di verificare se il funzionamento rispetta i requisiti. L'obiettivo del testing è trovare bug nel sistema, testando il maggior numero di funzioni, per poi correggerli. Questo documento ha il compito di identificare una strategia di testing di integrazione per il sistema UniSeat, al fine di poter verificare il comportamento di un insieme di componenti integrate.

1.1 Definizioni, Acronimi e Abbreviazioni

1.1.1 Definizioni

- Branch Coverage: indica i branch testati all'interno del software;
- Testing di Integrazione: fase di test del software in cui i singoli moduli software vengono combinati e testati come gruppo;

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_TPD_V1.0.pdf – Test Plan Document

-- US_TCD_V1.1.pdf – Test Case Document

-- US_ODD_V1.2.pdf – Object Design Document

2. Test di integrazione

2.1 Approccio di Integration Testing

La strategia adottata per il testing è quella di tipo "Bottom-up". Quest'ultima prevede che venga effettuato il testing individuale sui layer dei sottosistemi di livello più basso della gerarchia. I successivi sottosistemi ad essere testati sono quelli che utilizzano servizi dei sottosistemi testati in precedenza. Si ripete quest'ultimo passo finché tutti i sottosistemi non sono stati testati.

Per il testing di integrazione "Bottom-up" vengono utilizzati i Test Driver, che simulano le componenti dei livelli più alti non ancora integrate. Per testare i layer dei dati è sufficiente l'esecuzione delle query, mentre per testare i layer di logica è sufficiente l'esecuzione dei metodi implementati. Per eseguire il testing di integrazione sarà utilizzato il tool Travis CI, un servizio utilizzato per testare progetti software.

2.2 Componenti da testare

La scelta delle componenti da testare segue la decisione di eseguire la strategia di testing Bottom-up.

Per quanto riguarda il layer Model, le componenti da testare sono:

- Utente
- Aula
- Edificio
- Prenotazione
- UtenteDAO
- AulaDAO
- EdificioDAO
- PrenotazioneDAO

Per quanto riguarda il layer Control, le componenti da testare raggruppate nel seguente modo sono:

Comuni:

- PrelevaPrenotazioniServlet
- PrelevaUtentiServlet
- PrelevaAuleServlet
- PrelevaEdificiServlet
- EliminaPrenotazioneServlet

Autenticazione:

- LoginServlet
- LogOutServlet

Studente:

- ModificaDatiProfiloServlet
- PrenotaPostoServlet
- RegistrazioneServlet

Docente:

- PrenotazioneAulaServlet

Admin:

- InserisciAulaServlet
- InscrizioneDocenteServlet
- ModificaAulaServlet
- EliminaUtenteServlet

Utili:

- PasswordEncrypter
- EmailManager
- DisponibilitaChecker

Database:

- DBConnection

Per quanto riguarda il layer View, le componenti da testare raggruppate nel seguente modo sono:

Comuni:

- IndexJsp
- AuleJSP
- LoginJSP
- PrenotazioniJSP

Studente:

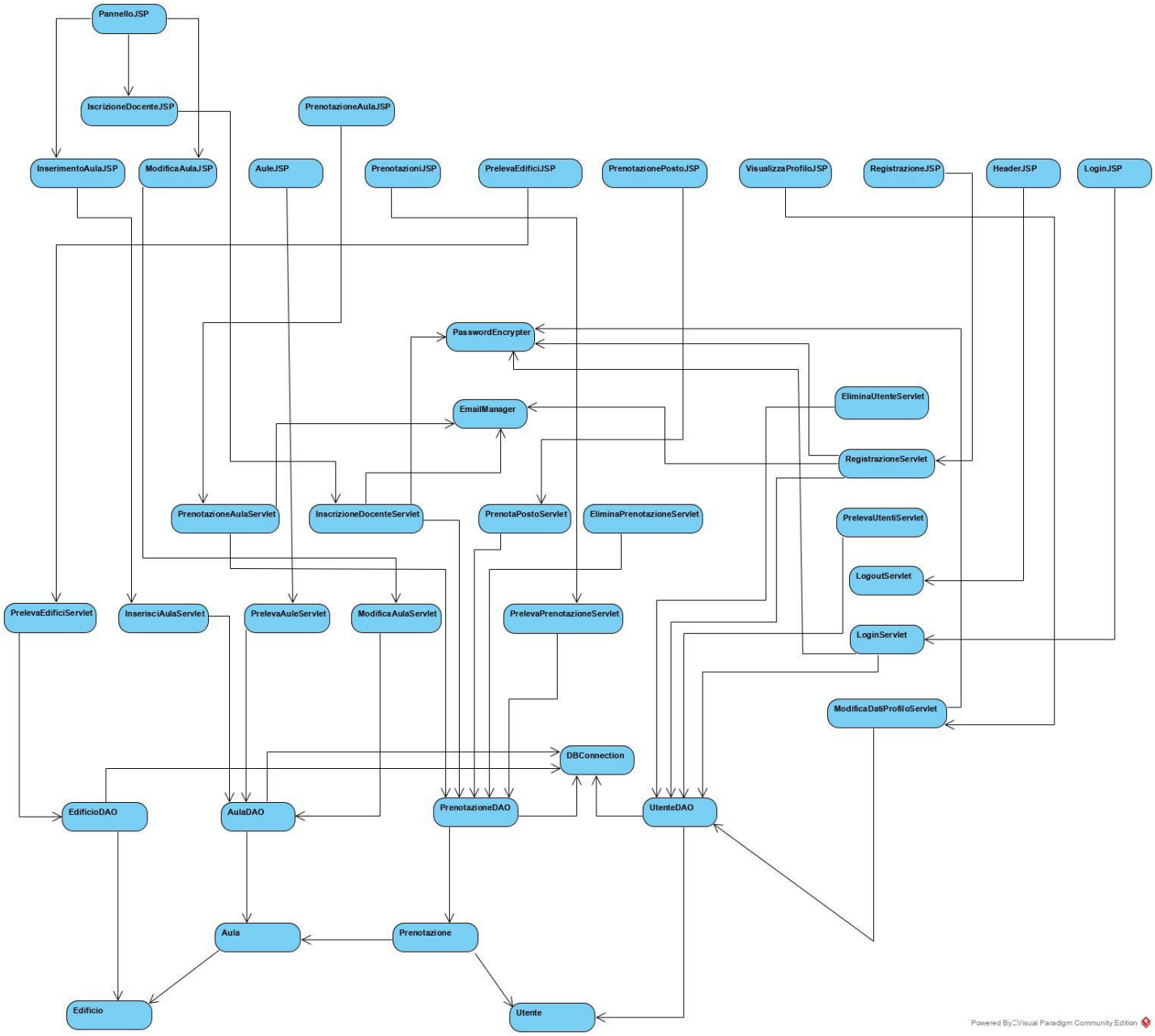
- RegistrazioneJSP
- PrenotazionePostoJSP
- VisualizzaProfiloJSP

Docente:

- PrenotazioneAulaJSP

Admin:

- IscrizioneDocenteJSP
- PannelloJSP
- ModificaAulaJSP
- InserimentoAulaJSP


PannelloJSP
IscrizioneDocenteJSP
PrenotazioneAulaJSP
InserimentoAulaJSP
ModificaAulaJSP
AuleJSP
PrenotazioniJSP
PrelevaEdificiJSP
PrenotazionePostoJSP
VisualizzaProfiloJSP
RegistrazioneJSP
HeaderJSP
LoginJSP
PasswordEncrypter
EmailManager
EliminaUtenteServlet
RegistrazioneServlet
PrenotazioneAulaServlet
InscrizioneDocenteServlet
PrenotaPostoServlet
EliminaPrenotazioneServlet
PrelevaUtentiServlet
LogoutServlet
PrelevaEdificiServlet
InserisciAulaServlet
PrelevaAuleServlet
ModificaAulaServlet
PrelevaPrenotazioneServlet
LoginServlet
ModificaDatiProfiloServlet
DBConnection
EdificioDAO
AulaDAO
PrenotazioneDAO
UtenteDAO
Aula
Prenotazione
Edificio
Utente
Powered By Visual Paradigm Community Edition

3.Pass/fail criteri

Se l'output osservato risulta essere diverso dall'output atteso, il testing ha successo.

Parleremo, quindi, di SUCCESSO se verranno individuate delle failure. In tal caso verranno analizzate e, se legate a dei fault, si procederà con le dovute correzioni. Infine, sarà iterata la fase di testing per verificare che le modifiche apportate non abbiano avuto impatto su altri componenti del sistema.

Si parlerà, invece, di FALLIMENTO se il test non riesce ad individuare un errore.