Università degli studi di Salerno

***Corso di Laurea in Informatica***

******

***INGEGNERIA DEL SOFTWARE***

***Test Summary Report***

***“UNI-AirLines”***

# Studenti:

##### Nome Matricola

Santoro Mario 0512104850

Marino Raffaele 0512104508

Pastore Matteo 0512104724

Fortunato Angelo 0512104532

*Anno Accademico: 2018/19*

***SOMMARIO***

**1. Introduction…………………………………………………………2**

**2. Riepologo Testing…………………………………………………2**

2.1 Testing d’unità…………...………………………………………………………….2

2.2 Testing d’integrazione…..………………………………………………………..2

2.3 Testing di sistema………………………..………………………………………..2

**3. Risultati Testing…………………………………………………..5**

**4. Risoluzioni………………………………………………………….7**

1. **Introduction**

Il presente documento fornisce un riepilogo dei risultati di tutti i processi di testing effettuati sul sistema Uni-AirLines dopo il completamento della fase di implementazione. In particolare, vengono analizzate le attività di testing pianificate prima della realizzazione del sistema, confrontandole con quelle effettivamente eseguite e con i risultati ottenuti.

1. **Riepilogo Testing**

Così come descritto dal documento Test Plan – Uni-AirLines, le attività di testing che sono state pianificate durante la fase di progettazione del sistema si dividono in tre diverse categorie: test d’unità, test d’integrazione e test di sistema. Ognuna di queste, focalizzandosi su diversi aspetti del sistema e servendosi di approcci differenti, ha l’obiettivo di verificare che il comportamento del sistema rispetti quello atteso e pianificato durante lo sviluppo. Combinando tra loro questi diversi tipi di testing, si è tentato di garantire un rilevamento degli errori che risultasse quanto più completo possibile.

**2.1 Testing d’unità**

La fase di testing di unità prevedeva di testare le diverse componenti del sistema realizzato. Le componenti del layer Model sono state tutte testate così come pianificato, dove è stato analizzato il comportamento di tutti i metodi rilevanti delle classi Bean e DAO utilizzando, quando necessario, degli stub per sostituire le componenti del livello sovrastante. Lo stesso discorso non vale per i due layer successivi, ovvero Controller e View. Entrambi i livelli, infatti, sono composti da classi Servlet e Jsp, le quali dipendono fortemente dal server Tomcat sul quale fanno affidamento. Per questo motivo, non è stato possibile testare i metodi delle suddette classi.

Per implementare i casi di test realizzati, ci si è serviti delle librerie Java fornite da JUnit.

**2.2 Testing d’integrazione**

Per quanto riguarda il testing d’integrazione, anche questa attività ha subito gli stessi problemi che hanno riguardato il testing d’unità delle classi Servlet e Jsp, infatti, non è stato sufficiente servirsi dei driver nel primo caso e degli stub nel secondo, poiché la natura stessa del sito web implementato tramite server Tomcat non ha permesso di isolare alcune di queste componenti dalle altre. Per testare ciò che non si è potuto provare qui, ci si è affidati alla fase d testing di sistema.

**2.3 Testing di sistema**

Il testing di sistema è stato realizzato tenendo presente le diverse funzionalità del sistema che si è deciso di testare, così come specificato dal Test Plan. Questa attività di testing è riuscita in larga parte a coprire le lacune evidenziate in precedenza sia dal testing d’unità che dal testing d’integrazione, in quanto, andando a testare direttamente le funzionalità disponibili, sono stati verificati anche tutti i comportamenti che si desiderava testare durante le altre fasi di testing.

I vari casi di test sono stati codificati tramite le librerie Java messe a disposizione da Selenium, che hanno permesso di verificare la correttezza dell’esecuzione di tutte le funzionalità offerte dal sistema.

**3.Risultati testing**

|  |  |  |  |
| --- | --- | --- | --- |
| Classe | Metodo | Esito | Note |
| AereoDAO | doSave | **Negativo** |  |
| AereoDAO | getAvaibleAir | **Negativo** |  |
| AereoDAO | getCodPosti | **Negativo** |  |
| AereoDAO | maxAereo | **Negativo** |  |
| AeroportoDAO | doSave | **Negativo** |  |
| AeroportoDAO | getAeroporto | **Negativo** |  |
| PrenotazioneDAO | showAllReservation | **Negativo** |  |
| UtenteDAO | doRetriveByKey | **Negativo** |  |
| UtenteDAO | doSave | **Negativo** |  |
| VoloDAO | doSave | **Negativo** |  |
| VoloDAO | showAll | **Negativo** |  |
| VoloDAO | doRettriveByAeroportiEDataSolaAndata | **Negativo** |  |
| VoloDAO | doRetriveByCod | **Negativo** |  |
| VoloDAO | isLowCost | **Negativo** |  |
| VoloDAO | showAllLowCost | **Negativo** |  |
| VoloDAO | lowCostByCodVolo | **Negativo** |  |

**4.Risoluzioni**

L’esecuzione dei test di sistema non ha rivelato alcun errore pertanto non è stato necessario andare ad agire sul codice del progetto.