|  |
| --- |
| **Università degli Studi di Salerno Corso di Ingegneria del Software** |

**Hotel Campus  
Test Plan Document  
Versione 1.0**

Immagine che contiene schermata, Elementi grafici, design

Descrizione generata automaticamente

Data: 15/12/2024

**Coordinatore del progetto:**

|  |  |
| --- | --- |
| **Nome** | **Matricola** |
|  |  |
|  |  |

**Partecipanti:**

|  |  |
| --- | --- |
| **Nome** | **Matricola** |
| Luca Del Bue | 0512116173 |
| Salvatore Di Martino | 0512116932 |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Scritto da:** | Team members |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versione** | **Descrizione** | **Autore** |
| 15/12/2024 | 1.0 | Prima stesura | Team members |
|  |  |  |  |

Indice

1. Introduzione 4

2. Relazione con altri documenti 4

3. Feature da testare/da non testare 4

4. Pass/Fail Criteria 4

5. Approccio 5

6. Sospensione e ripristino 5

7. Materiale di testing 6

8. Test Cases 6

8.1. Finalizza Prenotazione 6

1. Introduzione

L’obiettivo del test è andare a verificare la correttezza e il grado di affidabilità della funzionalità di prenotazione del soggiorno. Il processo di prenotazione inizia con la ricerca delle camere disponibili per il periodo di tempo selezionato. Dopodiché il cliente seleziona la camera d’interesse e aggiunge eventualmente i servizi extra alla prenotazione. Dopo aver visualizzato un riepilogo del soggiorno da prenotare, il cliente può finalizzare la prenotazione.

1. Relazione con altri documenti

Per la corretta individuazione del test case, si fa riferimento ad altri documenti prodotti:

* Requirements Analysis Document (RAD): il testing fa riferimento al caso d’uso UC16 Finalizza Prenotazione.
* System Design Document (SDD)
* Object Design Document (ODD)

1. Feature da testare/da non testare

Si effettuerà il testing della feature:

* Finalizza Prenotazione.

1. Pass/Fail Criteria

Le attività di testing mirano a rilevare errori nel sistema per consentirne la correzione. L’esito di un test è determinato da un oracolo, che rappresenta il risultato atteso basato sui requisiti. Un test ha esito positivo (pass) se l’output ottenuto differisce da quello atteso, mentre fallisce (fail) se l’output coincide con l’atteso.

Il testing sarà considerato valido se verranno testati tutti i requisiti funzionali relativi alla funzionalità trattata. Nel caso in cui una failure venga individuata e risolta, si procede con un regression testing, rieseguendo anche i test precedenti.

1. Approccio

Il testing si compone di tre fasi: testing di unità, testing di integrazione e testing di sistema.

**Test di unità**

Per il test di unità utilizziamo un approccio black-box, ossia un testing funzionale basato sulle specifiche per partizionare lo spazio degli input in classi di equivalenza. Il test di unità verrà realizzato con JUnit e Mockito per la creazione di stubs e drivers.

**Test di integrazione**

Il test di integrazione sfrutta una strategia Sandwich, dove il target layer è rappresentato dall’application logic layer, individuato nella decomoposizione in sottosistemi (SDD).

Dunque il top layer (interface) ed il bottom layer (storage) verranno testati in modo isolato utilizzando stubs e drivers. In seguito, il test di integrazione andrà a testare il target layer sfruttando il bottom layer testato precedentemente.

Immagine che contiene testo, schermata, linea, Carattere

Descrizione generata automaticamente

Infine, per integrare i tre layer, gli stub utilizzati per il testing del top layer saranno sostituiti con il resto delle componenti già testate. In generale, il test di integrazione sarà compreso nella stessa classe di test di quelli unitari. Mockito sarà utilizzato per il test di integrazione.

Immagine che contiene testo, schermata, linea, Carattere

Descrizione generata automaticamente

**Test di sistema**

Per il test di sistema sarà eseguito il functional testing, con l’obiettivo di testare la funzionalità nel suo insieme. Per questo test verranno riutilizzati i casi di test del test di unità e di integrazione. Sarà utilizzato il tool Selenium IDE, che permette di registrare le azioni che un utente può intraprendere sul browser, in modo da poter implementare ed eseguire i test case di sistema.

1. Sospensione e ripristino

**Criteri di sospensione**

Il testing verrà sospeso in caso di rilevazione di una failure.

**Criteri di ripristino**

Il testing verrà ripreso dopo aver risolto i fault individuati.

1. Materiale di testing

L’hardware necessario per l’attività di test prevede un semplice computer.

1. Test Cases

L’approccio per la definizione dei test frame sarà il category partition. Al fine di minimizzare il numero di test case, gli input saranno partizionati in classi di equivalenza. Ogni scelta annotata con una condizione di errore verrà testata singolarmente assumendo che le altre categorie abbiano scelte valide, senza combinare errori di altre categorie. Ogni errore isolato rappresenta già un caso test valido.

## Finalizza Prenotazione

Definizione dei parametri:

* dataCheckIn (CI)
* dataCheckOut (CO)
* dataPrenotazione (DP)
* numeroOspiti (NO)
* camera (CA)
* servizi (S)

Ogetti dell’ambiente:

* cliente (CL)

Definizione delle categorie e delle scelte:

* dataCheckIn
  1. nessuna data selezionata [error]
  2. data minore della data corrente [error]
  3. data uguale alla data corrente
  4. data maggiore della data corrente
  5. data di check in maggiore o uguale alla data di checkout [error]
* dataCheckOut
  1. nessuna data selezionata [error]
  2. data minore della data corrente [error]
  3. data uguale alla data corrente [error]
  4. data maggiore della data corrente e maggiore della data di check in
  5. data di check out minore della data di check in [error]
* dataPrenotazione
  1. data uguale a null [error]
  2. data minore o uguale alla data corrente
  3. data maggiore della data corrente [error]
* numeroOspiti
  1. numero minore o uguale a 0 [error]
  2. numero maggiore di 0 e minore o uguale al numero massimo di ospiti della camera
  3. numero maggiore del numero massimo di ospiti della camera [error]
* camera
  1. camera uguale a null [error]
  2. camera disponibile
  3. camera non più disponibile
* servizi
  1. lista di servizi uguale a null [error]
  2. nessun servizio selezionato
  3. per ogni tipo di servizio selezionato il numero di servizi è minore o uguale al numero di ospiti della prenotazione
  4. per almeno un tipo di servizio selezionato il numero di servizi è maggiore del numero di ospiti della prenotazione [error]
* cliente
  1. cliente uguale a null (non autenticato) [error]
  2. cliente autenticato

|  |  |  |
| --- | --- | --- |
| Test Case ID | Test Frame | Esito |
| TC1 | CI1, CO4, DP2, NO2, CA2, S2, CL2 | Prenotazione non effettuata - nessuna data di check-in selezionata |
| TC2 | CI2, CO4, DP2, NO2, CA2, S2, CL2 | Prenotazione non effettuata - data di check-in minore della data corrente |
| TC3 | CI5, CO4, DP2, NO2, CA2, S2, CL2 | Prenotazione non effettuata - data di check-in maggiore o uguale alla data di check-out |
| TC4 | CI3, CO1, DP2, NO2, CA2, S2, CL2 | Prenotazione non effettuata - nessuna data di check-out selezionata |
| TC5 | CI3, CO2, DP2, NO2, CA2, S2, CL2 | Prenotazione non effettuata - data di check-out minore della data corrente |
| TC6 | CI3, CO3, DP2, NO2, CA2, S2, CL2 | Prenotazione non effettuata - data di check-out uguale alla data corrente |
| TC7 | CI3, CO5, DP2, NO2, CA2, S2, CL2 | Prenotazione non effettuata - data di check-out minore della data di check-in |
| TC8 | CI3, CO4, DP1, NO2, CA2, S2, CL2 | Prenotazione non effettuata - data di prenotazione non presente |
| TC9 | CI3, CO4, DP3, NO2, CA2, S2, CL2 | Prenotazione non effettuata - data di prenotazione maggiore della data corrente |
| TC10 | CI3, CO4, DP2, NO1, CA2, S2, CL2 | Prenotazione non effettuata - numero di ospiti minore o uguale a 0 |
| TC11 | CI3, CO4, DP2, NO3, CA2, S2, CL2 | Prenotazione non effettuata - numero di ospiti maggiore del massimo consentito per la camera |
| TC12 | CI3, CO4, DP2, NO2, CA1, S2, CL2 | Prenotazione non effettuata - camera non presente |
| TC13 | CI3, CO4, DP2, NO2, CA2, S1, CL2 | Prenotazione non effettuata - lista di servizi non presente |
| TC14 | CI3, CO4, DP2, NO2, CA2, S4, CL2 | Prenotazione non effettuata - numero di servizi superiore al numero di ospiti |
| TC15 | CI3, CO4, DP2, NO2, CA2, S2, CL1 | Prenotazione non effettuata - cliente non autenticato |
| TC16 | CI3, CO4, DP2, NO2, CA2, S2, CL2 | Prenotazione effettuata |
| TC17 | CI3, CO4, DP2, NO2, CA2, S3, CL2 | Prenotazione effettuata |
| TC18 | CI3, CO4, DP2, NO2, CA3, S2, CL2 | Prenotazione non effettuata – camera non più disponibile |
| TC19 | CI3, CO4, DP2, NO2, CA3, S3, CL2 | Prenotazione non effettuata – camera non più disponibile |
| TC20 | CI4, CO4, DP2, NO2, CA2, S2, CL2 | Prenotazione effettuata |
| TC21 | CI4, CO4, DP2, NO2, CA2, S3, CL2 | Prenotazione effettuata |
| TC22 | CI4, CO4, DP2, NO2, CA3, S2, CL2 | Prenotazione non effettuata – camera non più disponibile |
| TC23 | CI4, CO4, DP2, NO2, CA3, S3, CL2 | Prenotazione non effettuata – camera non più disponibile |