Semester 3 individual project

S3-S-CB-03

Onea Bianca

Test Plan

November 3, 2021

Contents

[1. Why is testing important? 2](#_Toc86915367)

[2. What tests are going to be used? 2](#_Toc86915368)

[Unit Testing 2](#_Toc86915369)

[Integration Testing 2](#_Toc86915370)

[3. Test strategy 3](#_Toc86915371)

[1. Test risks/issues 3](#_Toc86915372)

[2. Test scope 3](#_Toc86915373)

3. [Pass/Fail criteria 5](#_Toc86915374)

4. [Test Tools 5](#_Toc86915375)

[4. User acceptance tests 6](#_Toc86915376)

# 1. Why is testing IMPORTANT?

[**Software testing**](https://rezaid.co.uk/services/software-testing/) is a process of checking software applications and products for bugs and errors in order to ensure their performance is efficient. Testing in software engineering is a fundamental process of creating reliable and usable software products. It checks for functionalities, usabilities, security, compatibility and performance of the application, so the developer(s) can aknowledge if their application works properly, has bugs or mulfunctionalities or if something is not working as expected.

# 2. WHAT TESTS ARE GOING TO BE USED?

In order to check if my web application is working properly or not, I am going to make use of the following types of tests:

## Unit Testing

Unit testing consists in testing an individual component of a software application.

The aim behind unit testing is to check for errors, mulfunctionalities or bugs for each

unit and, eventually, correcting them.

For this project, I am going to make use of unit tests in order to understand better the

base of my code and change the defects which are causing problems faster.

Furthermore, it helps me fix the problems early in the development phase, which means that

there is possibility to occur a smaller number of defects in upcoming testing levels.

## **Integration Testing**

Integration testing is the second level of the software testing process comes after unit testing. In this testing, units or individual components of the software are tested altogether. The focus of the integration testing level is to expose defects at the time of interaction between integrated components. Once all the components or modules are working independently, then you need to check the data flow between the dependent modules, which is known as **integration testing**.

For this project, I am going to start writing integration tests after I made sure that each one of the individual components are running properly and, once all the components or modules are working independently, I am going to check the data flow between the dependent modules

# 3. TEST STRATEGY

### **TEST RISKS / ISSUES**

|  |  |
| --- | --- |
| *Test Plan Risk* | *Mitigation Strategy* |
| Tight deadlines | Set **Test Priority** for each of the test activity. |
| Insufficient resources | Individual research and documentation |
| Lack of standards and techniques for testing | After the research is done, start practicing and applying the learned techniques |

### **TEST SCOPE**

Unit Testing: The scope of using unit tests for my application is to make sure that every individual component runs smoothly and works as expected.

Integration testing: The scope of using integration tests for my application is to make sure that all the individual components work together, without causing any bugs or errors.

**Items in scope**

|  |  |
| --- | --- |
| *Test item* | *Description* |
| UserDao | INTERFACE – Extends JpaRepository interface, communicates with the database and handles CRUD operations for the User entity |
| HotelDao | INTERFACE - Extends JpaRepository interface, communicates with the database and handles CRUD operations for the Hotel entity |
| OfferDao | INTERFACE - Extends JpaRepository interface, communicates with the database and handles CRUD operations for the Offer entity |
| RoomDao | INTERFACE - Extends JpaRepository interface, communicates with the database and handles CRUD operations for the Room entity |
| IUserDal | INTERFACE - Contains the methods which are going to be used in UserService class |
| IHotelDal | INTERFACE - Contains the methods which are going to be used in HotelService class |
| IOfferDal | INTERFACE - Contains the methods which are going to be used in OfferService class |
| IRoom Dal | INTERFACE - Contains the methods which are going to be used in RoomService class |
| UserDalJPA | CLASS – implements the methods in IUserDal, with the use of the UserDao interface, which directly communicates with the database |
| HotelDalJPA | CLASS - implements the methods in IHotelDal, with the use of the HotelDao interface, which directly communicates with the database |
| OfferDalJPA | CLASS - implements the methods in IOfferDal, with the use of the OfferDao interface, which directly communicates with the database |
| RoomDalJPA | CLASS - implements the methods in IRoomDal, with the use of the RoomDao interface, which directly communicates with the database |
| IUserService | INTERFACE – Contains the methods which are going to be implemented in UserService class |
| IHotelService | INTERFACE - Contains the methods which are going to be implemented in HotelService class |
| IOfferService | INTERFACE - Contains the methods which are going to be implemented in OfferService class |
| IRoomService | INTERFACE - Contains the methods which are going to be implemented in RoomService class |
| UserService | SERVICE CLASS – implements the methods in IUserService Interface, with the use of IUserDal interface |
| HotelService | SERVICE CLASS - implements the methods in IHotelService Interface, with the use of IHotelDal interface |
| OfferService | SERVICE CLASS - implements the methods in IOfferService Interface, with the use of IOfferDal interface |
| RoomService | SERVICE CLASS - implements the methods in IRoomService Interface, with the use of IRoomDal interface |
| UserController | CONTROLLER CLASS – processes all incoming REST API requests regarding the User entity, preparing a model, and returning the view to be rendered as a response |
| HotelController | CONTROLLER CLASS – processes all incoming REST API requests regarding the Hotel entity, preparing a model, and returning the view to be rendered as a response |
| OfferController | CONTROLLER CLASS – processes all incoming REST API requests regarding the Offer entity, preparing a model, and returning the view to be rendered as a response |
| RoomController | CONTROLLER CLASS – processes all incoming REST API requests regarding the Room entity, preparing a model, and returning the view to be rendered as a response |
|  |  |
|  |  |

**Items out of scope**

- WebSecurityConfiguration class

- CorsConfiguration class

- JwtAuthenticationEntryPoint class

### **PASS/FAIL CRITERIA**

The pass/fail criteria for both unit tests and integration tests is going to be determined by JUnit, the unit testing framework I am going to use for making the unit tests. After the execution of all the tests, JUnit will mark each individual unit test. The entirety of our unit testing will be considered complete once all test cases pass successfully.

### **Test Tools**

I am going to use JUnit framework in order to test my code.

# User acceptance tests

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Name | Pre-condition | Steps | Test data | Expected result |
| TC-01 | Log in as an user | Both frontend and backend are running and the user is on the Log In page | 1. Enter the username  2.Enter the password  3.Click “Log In” button | Username: bianca\_user  Password:  12345 | The user will be redirected to the Home page, the “Log In” and “Sign Up” buttons from the navbar will be replaced by the user’s username |
| TC-02 | Log in as an admin | Both frontend and backend are running and the admin is on the Log In page for the admins | 1. Enter the username  2.Enter the password  2.Click “Log In” button | Username:  bianca\_admin  Password:  parolaadmin | The admin will be redirected to the Admins’ dashboard |
| TC-03 | Try incorrect log in credentials | Both frontend and backend are running and the user is on the Log In page | 1. Enter the username  2.Enter the password  3.Click “Log In” button |  | The user will not be redirected to the Home page and the Log In page will be updated with the corresponding error messages (Invalid credentials, The fields are empty, etc) |
| TC-04 | Booking a hotel | Both frontend and backend are running and the user is on the Home page, at the Booking form | 1.Enter the destination  2.Enter the check-in date  3.Enter the check-out date  4.Enter the number of adults  (The number of children is not required) | Destination:  Bucharest  Check-in date:  2021-11-12  Check-out date:  2021-11-15  Number of adults: 1 | The user will be redirected to a page where the results of his/her search will be shown (all the hotels that meet the criteria) |
| TC-05 | See information about a certain hotel | Both frontend and backend are running and the user is either on:  1.Home page, at the Explore our hotels” section  2. The page that displays the hotels that meet the criteria of his/her search | 1.Press “Explore now” button  (If the user is on the Explore our hotels” section)  OR  1.Press “View more” button  (If the user is on the page that displays the hotels that meet the criteria of his/her search) | Destination:  Bucharest  Check-in date:  2021-11-12  Check-out date:  2021-11-15  Number of adults: 1  (If the user is on the page that displays the hotels that meet the criteria of his/her search) | The user will be redirected to a page where all the details about the hotels he/she selected are shown (pictures, address, contact information, facilities, prices, etc) |
| TC-06 | Submit the contact form | Both frontend and backend are running and the user is on the Home page, at the Contact section | 1.Enter the name  2.Enter the e-mail  3.Enter the message | Name:  Bianca Onea  E-mail:  [bia.onea@gmail.com](mailto:bia.onea@gmail.com)  Message:  “I have problems reserving a hotel, could you please give me more details about what should I do?” | The user’s details and message will be send to the admins, who will see the message and respond |
| TC - 07 | Reedem an offer | Both frontend and backend are running and the user is on the Home page, at the Offer section | 1.Press “Redeem offer” button |  | The user will be redirect to a page where more details about the offer will be shown |
| TC-08 | See all hotels | Both frontend and backend are running , the admins is logged in and is on the Dashboard page | 1.Log in  (using the steps from TC-02)  2.Go to dashboard page |  | The page will display all the hotels registered on the website |
| TC-09 | See all the registered users on the website | Both frontend and backend are running , the admins is logged in and is on the Dashboard page | 1.Log in  (using the steps from TC-02)  2.Go to dashboard page |  | The page will display all the users registered on the website |
| TC-10 | Create an account on the website | Both frontend and backend are running and the user is on the Sign Up page | 1.Enter the first name  2.Enter the last name  3.Enter the username  4.Enter the password | First name:  Dragos  Last name:  Onea  Username:  dragos\_onea  Password:  dragos123 | The user will be redirected to the Home page, the “Log In” and “Sign Up” buttons from the navbar will be replaced by the user’s username |