Test Plan
for

Booking platform

-Semester 3 Individual Project-

Plamen Peev

Student number: 4179080

Contents

1.Scope

- Target audience
  - This document will be reviewed by the following teachers:
    - Geurts,Jaap J.
    - Beks, Mark M.L.
  - This document's target audience is:
    - Any person who is interested in the test strategy of the project
    - Any tester who wants to execute the tests and verify that the application is ready for deployment, based on the given criteria in this document
- Testing activities carried out during the creation of the project
  - Requirements Analysis
  - Test Planning
  - Test Environment Setup
  - Test Execution
- Tested components - iTrips API
  - Controllers
  - Services
    - Mocking tests
    - Faking tests

2.Test Strategy

- Testing levels
    - Unit testing
        - Mocking tests
        - Faking tests
    - Integration testing
        - API endpoint requests
    - System testing
        - Manual testing
        - Acceptance testing
        - Security testing
- Testing Environment
    - Testing tools
        - Junit (Unit testing)
        - SonarQube (Code Quality)
        - Mockito (Mock unit tests)
        - H2 – in-memory-database (Faking tests)
        - Postman – (Manual testing)
    - Automated Tests using CI/CD
        - Unit tests
        - Integration tests

3.Test Deliverables

3.1. Introduction

Before explaining what and how to test, here is a short description of the flow of the tests

We have 3 main components we are going to test, with 3 subsequent types of tests that will be executed for each of the components. The 3 components are:

-Bookings

-Hotels

-Users

Each of them has 3 types of tests:

-**Mocking Tests**: Used to test if the correct methods are called inside the corresponding method

-**Faking Tests**: Used to test if the correct result is being asserted after executing the verified actions form the mocking tests

-**Integration tests**: Used to test if the request to the API returns the expected response after completing the actions, which were verified by the mock and faking tests

3.2. What To Test

- Bookings
    - Service Mocking Test
        - Save a booking
        - Delete a booking
    - Service Faking Test
        - Save a booking
        - Check if a booking is possible for certain room and date
        - Get available offers based on several parameters
        - Get all bookings by a user
        - Cancel a booking
    - Controller Integration Test

      -All API endpoints which are calling the above listed methods form the service class

- Hotels
    - Service Mocking Test
        - CRUD methods for hotels
        - CRUD methods for rooms
        - Get hotel by manager Id
    - Service Faking Test

      -All methods from the Mock Test
    - Controller Integration Test

      -All API endpoints which are calling the above listed methods form the service class

- Users
    - Service Mocking Test
        - CRUD operations for user
        - Exception handling on failed login/registration
        - Successful login/registration
    - Service Faking Test

      -All methods from the Mock Test
    - Controller Integration Test

      -All API endpoints which are calling the above listed methods form the service class

3.3. Tests Execution

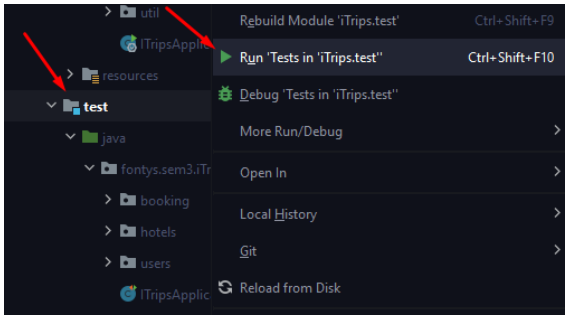
3.3.1. Unit & Integration Tests

- Command – Line

To run the tests without opening the project with IntelliJ, simply open any command-line tool inside the folder 'backend/iTrips' and type : ./gradlew test

- IntelliJ

Open the folder 'backend/iTrips' in IntelliJ. Then from the Project tab, right-click on the folder 'iTrips/src/test' and then click 'Run tests'



3.3.2 Acceptance Tests

To execute the Acceptance tests, follow the cases from 4.Test Cases

3.4. Test Completeness

In order to define the above executed testing as 'complete', for Point 3.3.1 you should see 'BUILD SUCCESSFULL'.

For Point 3.3.2, The corresponding filed from 'Expected result' should be met.

4.Test Cases

ID	Action	Pre-condition	Test data	Expected result
TC-01	Register as a Booker	Opened the website	First Name: Anthony Last Name: Garfield Username: anthony Password: anthony1234	System adds the user to the database and user is redirected to login page
TC-02	Login as a Booker	Opened the website	Username: Anthony Password: anthony1234	System check if username and password are correct and redirects person to booker home page with saved JWT in localhost
TC-03	Update user profile	Logged in with any account	Name: Anthony Joshua Country: Netherlands City: Eindhoven Street: Marconilaan 65	System saves the new data in the database and User sees his profile updated
TC-04	Update use password	Logged in with any account	Current password: (the account's password) New Password: updated123 Repeat Password: updated123	System checks if current password is correct and user can log in with his new password
TC-05	Logout	Logged in with any account	Click on the user avatar at the top right corner, then select Logout	System deletes user information from localhost and user is redirected to login page
TC-06	Login as a Hotel Manager	Opened the website	Username: gregor Password: gregor1234	System checks if username and password are correct and redirect to hotel manager overview page
TC-07	Create a Hotel	Logged in with a hm (hotel manager) account	Hotel Name: Royal Spa Helmond Caption: Test Caption Description: Test Description Stars: 4, Country: Netherlands City: Eindhoven, Street: Teststraat 123	System adds the hotel to the database and redirects to the 'Add new room' page
TC-08	Create a Room	Logged in with hm account, completed TC-05 step	Room type: Double Deluxe Sleeps: 2 Total rooms: 10 Price: 35.00	System adds the room to the database and adds it to the hotel overview details list in the page
TC-09	Update a Hotel	Logged in with hm account, Clicked on an edit icon form the hotels' list	Hotel Name: Royal Spa Helmond Caption: Test Caption UPDATED Description: Test Description UPDATED Stars: 5, Country: Netherlands City: Eindhoven, Street: Teststraat 123	System saves the new data in the database and HM sees the hotel updated
TC-10	Update a Room	Logged in with hm account, Clicked on an edit icon form the hotels' list Clicked on 'Next' button Clicked on an existing room	Room type: Double Deluxe Updated Sleeps: 2 Total rooms: 100 Price: 55.00	System saves the new data in the database and HM sees the updated Room in the list

TC-11	Search for a booking	Logged in with booker account, Completed TC-06 step	Location: Eindhoven, Netherlands Check-in: 01/02/2022 Check-out: 06/02/2022 Guests: 4	System check the database for available rooms with the given parameters and retrieves one result
TC-12	Book a room	Logged in with booker account, Completed TC-09 step	Select on the retrieved hotel, then select on any room from the list of rooms in the booking offer page	System adds the booking to the database and user is redirected to the bookings list
TC-13	Delete a booking	Logged in with booker account, Completed TC-10 step	Click on delete button in the bookings list, after this click confirm on the pop-up window	System deletes the booking from the database and user doesn't see the booking in the list anymore