

Test Plan

Individual project

Nur Nechushtan | November 2022

Contents

[Versioning Table 3](#_Toc120269366)

[User Acceptance Tests 4](#_Toc120269367)

[Test strategy 6](#_Toc120269368)

# Versioning Table

|  |  |
| --- | --- |
| Date | Content |
| 22/11/2022 | First version of test plan |

# User Acceptance Tests

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Id | Test description | Step | Test steps | Expected result | User story |
| 1 | Verify that the app entry point is accessible to anonymous user | 1 | Anonymous user launches the app. | App has to be loaded successfully with the main content displayed. | US-1 |
| 2 | Navigation bar | The navigation bar should display Sign in and Sign up options. |
| 2 | Verify that the admin dashboard end point is NOT accessible to users without permissible role | 1 | Anonymous user launches the admin dashboard end point | App has to be loaded with the sigh in page | US-5 |
| 2 | User signs in with user account | App has to be loaded with Unauthorized page |
| 3 | Verify that find the closest airport gets the user location | 1 | Any user launches the app | App has to be loaded successfully with the main content displayed. | US-9 |
| 2 | The user presses the GPS button | App has to load the user location and display it in the location input. |
| 4 | Verify that the create operation works when a user want to make a price alert. | 1 | The user is signed in and made a valid flight search | App has to be loaded with list of flights matching the search criteria | US-4 |
| 2 | The user presses the Set price alert button | Pop up window has to be displayed with the dates of the flight and the origin and destination |
| 3 | The user presses the confirm button | Pop up window has to disappear, and a success notification is shown |
| 5 | Verify that the get operation works when a user wants to get the cheapest flight from his location | 1 | The user is signed | App has to be loaded with the main content and display the top 3 cheapest flights from the user closest airport | US-10 |
| 6 | Verify that a user can update his profile information | 1 | The user is logged in and sees the profile page | The application has successfully launched. The profile page shows, the user information and the user price alerts if any | US-6 |
| 2 | The user changes his own information and presses the update personal information button | The application saves the new information and renders the page with the user information |

# Test strategy

**Unit testing**

For the unit testing, we will test the logic layer. The reason for this is that we made our logic and it needs to be tested properly so it works as it is intended. Furthermore, the database and the data access layer will not be tested since they are made by professionals and have already been tested on their side. When testing the logic layer, we will mock the database and stub the external services to make sure that we isolate the test as much as possible, if the test fails in a later stage we could point out what break it. Finally, the pattern that we are going to use for the unit tests is the AAA pattern, which stands for Arrange, Act, and Assert.

**Integration testing**

For integration testing, we will test the integration of a flow in the application. Starting in the controller and moving on to the services calling an external API or calling the database. When making an integration test we will try to mock as little as possible, that way we can see the real behavior of the app. Because integration tests involve the database and can change the data stored in them, we will use a test profile with a dedicated test database that will be used when running the tests.