



# Test plan

---

---

## Version table

Author	Florin Deleanu 3784711	Revision Date
Version1	First draft	April 2021
Version1	Completed tests	May 2021

## Contents

Version table .....	1
Test cases .....	2
Unit tests .....	3
Integration Tests .....	3
Back end.....	3
Front end.....	3
Conclusion.....	3

## Test cases

The following test cases will be based on the user stories from the project plan and their acceptance criteria's. Only the ones included in this spring will be written

Test case	Name	Input test data	Expected result
1	Search for a specific flight	Origin: a Destination: b Date: 2021-01-01	Being redirected to another page where the flights are shown based on inputted data
2	Search for flights with origin	Origin: a	Being redirected to another page where all flights with origin "a" are shown
3	Search for flights with origin and destination	Origin: a Destination: b	Being redirected to another page where all flights with origin "a" and destination "b" are shown
4	Ordering flights after searching	Clicking on any column header Ex:price	Clicking on a column header will sort all flights that were displayed based on the column values Ex:price low-high or high-low
5	Seeing all planes owned by the company	Clicking on "our planes" button	Seeing an overview of all planes
5.1	Seeing detailed information about a plane	In the "our planes" page click on the show more details button for a plane	A pop up window displaying more specific info about the chosen plane
6	Starting booking a ticket	Clicking "book" button while in the Flights page while logged in	Redirected to the ticket booking page
6.1	Starting booking at ticket(not logged in)	Clicking "book" button while in the Flights page while not logged in	Redirected to login page
7	Actually booking a ticket	While on the ticket booking page click "next"	User in the second part of booking tickets
7.1	Selecting a free seat	Inputting Row:5 Seat:A	Text shows in green "Seat available". User can go to next step
7.2	Selecting an occupied seat	Inputting Row:1 Seat:A	Text shows in red "Seat not available". User can't go to the next step
7.3	Paying for the flight	User at the last step of the booking process clicks "Pay"	The booking will be added in the user's profile and the cost will be deducted from their budget
7.4	Cancelling Flight	User at the last step of the booking process clicks "Cancel"	User will be redirected to the flights page
8	Logging in	Email: <a href="mailto:a@a.com">a@a.com</a>	Redirected to profile page for test user

		Password: a	"a"
9	Register	Email: <a href="mailto:florind@gmail.com">florind@gmail.com</a> Password: asdfghjk123 First name: Florin Last name: Deleanu Birthdate: 1999-12-30	User with the mentioned details created in the database and now able to log in

## Unit tests

For the unit tests the functionality is split into 2: one with the fake data test and the other tests the actual service which uses a database. Because testing the actual database with unit tests is not a proper way to do it, mocks will be used to stub methods database for testing.

I test the service layer as that layer is the most important one because it contains the logic in the application. In addition, while testing the services, models will also be tested if they work as intended.

Most unit tests are performed on CRUD functions to determine whether they work properly.

## Integration Tests

### Back end

Some of the unit tests for the services include multiple linking repositories ( for example checking flights also requires checking the airplanes which fly them). Because of this fact the simple unit tests mentioned above turn into integration ones which check whether the logic functions as a whole process.

### Front end

Front end tests minimize the need to check manually after every code update if the interface keeps performing as expected. Because of this reason it is an important part of any web application.

For front end testing I am using Cypress to automatically check if certain elements exist or not and whether the application behaves as if being tested by a user. Examples of front end tests include checking whether the user can type in input boxes and whether they can log in or search for flights.

## Conclusion

Based on the aforementioned reasons, my application contains automatic tests for the front and back end. It also included manual acceptance tests for users to verify if everything performs according to expectations.