****

**[Flight Booking App]**

**TEST PLAN**

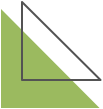
**Version [1.1]**

**Prepared By**

**[Anastasiia Yanieva]**

**[Junior QA]**

**[31.05.2023]**





**Version History**

| **Version #** | **Implemented By** | **Revision Date** | **Approved By** | **Approval Date** | **Reason** |
| --- | --- | --- | --- | --- | --- |
| [Version Number] | [Name] | [Date] | [Name] | [Date] | [Reason for the Revision] |
| Version 1.1 |  |  |  |  |  |
| Version 1.2 |  |  |  |  |  |
| Version 1.3 |  |  |  |  |  |
| Version 1.4 |  |  |  |  |  |

# 

[**1.Introduction 3**](#_tyjcwt)

[**2.Scope 3**](#_rr6kru79go56)

[**3. Features To Be Tested 4**](#_vfjri4cdlibh)

[**4. Features Not To Be Tested 5**](#_l9wtovhn7rdn)

[**5. Approach 5**](#_d6b6ej15ai3r)

[5.1 Automation testing 5](#_xnl7dashuami)

[5.2 Meetings and Organizing processes 5](#_c6isrx2lp43s)

[**6. Test strategy 6**](#_mls60khmtjw2)

[6.1 QA role in test process 6](#_7j15iboc23fb)

[6.2 Bug life cycle 7](#_1u956gvm89xr)

[6.3 Types of testing 7](#_d53ngv82tkp)

[6.4 Severity List 9](#_8mn70b9s9oja)

[6.5 Priority list 9](#_7ppf0lte26ch)

[**7. Item Pass/Fail Criteria 9**](#_76v2vcf4k1b7)

[**8. Suspension Criteria and Resumption Requirements 10**](#_c6y66zksjdfh)

[Suspension criteria 10](#_gt1xduss0g2g)

[Resumption Requirements 11](#_6tstnswvxjv3)

[**9. Test Deliverables 11**](#_rtogkpmqz7zn)

[**10. Resource and Environment needs 12**](#_o84jq7v3jkn2)

[10.1 Resource 12](#_9dq44uvnbkmb)

[10.2 Environment needs 13](#_h15ergj8213p)

[**11. Responsibilities 13**](#_m91f7g5zxo7m)

[**12. Schedule 14**](#_et7kbeeqw3d0)

[**13. Risks and Contingencies 15**](#_mbxwbcs3jb7h)

[**14. Approvals 15**](#_4m2jyp4ozu4x)

# 

# 

# 

# 1.Introduction

This test plan is designed to ensure high-quality testing of a mobile web application for booking airline tickets. Its goal is to achieve high quality, taking into account the specifics of the application.

This plan will describe the approaches and methodologies that will be used during testing. It defines the objectives, testers' responsibilities, test entry and exit criteria, scope of work, schedule of milestones, and overall approach.This test plan clearly defines the expected test results and the scope of the test scenarios. It also defines what is included in the test and what is excluded.

The main objectives of this plan are to ensure that it is understood and agreed upon by all team members and to obtain approval before testing begins. It serves as the main document for managing the test and coordinating the team's work. It also contributes to project quality assurance and identifies the need to create appropriate documents and reports that will be used during the testing process.

This test plan should be accessible and understandable to all stakeholders and should be updated and approved in accordance with changes in the test project.

The project is available [here](https://github.com/IraVolnyh/new-flight-app.git).

# 2.Scope

1. Scope of testing:

* Testing will cover all the functionality and features of our application, according to the requirements and specifications.
* We will check the main functions and possible use cases that affect the correct operation of the application.
* We will also test for compatibility with various operating systems, devices, and browsers.
* We will check the integration of our application with other systems or services.
* The appearance of the application, including compliance with the prototype design, will also be checked.

2. Exclusions from testing:

* We will not test aspects that are not part of the application's functionality or are not core requirements.
* Testing of servers, database, network, is not within the scope of this test plan.

# 3. Features To Be Tested

1. Selecting the departure city from the drop-down menu.

Make sure that the user can accurately select the desired departure city.

2. Selecting the arrival city from the drop-down menu.

Make sure that the user can select the correct arrival city from the options.

3. Selecting the departure date.

Make sure that the user can select the correct departure date from the available options.

4. Selecting the number of passengers (adults, children, elderly).

Ensure that the user can accurately select the desired number of passengers for each category using the "Steppers" function.

5. Selecting the travel class (economy, business, first class).

Ensure that the user can accurately select the desired travel class.

6. Searching for flights according to the specified parameters.

Make sure that the system correctly filters and displays available flights according to the selected criteria.

7. View information about the flight and its route.

Make sure that the user can access and view full information about the selected flight, including the route.

8. Booking a flight.

Ensure that the user can successfully book the flight by clicking the "Book Flight" button and that the booking indicator appears as expected.

9. Receive up-to-date flight information via API integration.

Ensure that the system properly retrieves and displays the latest flight information via API integration.

# 4. Features Not To Be Tested

Functions that should not be tested:

1. Payment processing: Payment processing functionality is outside the scope of this application and will not be tested.

2. User account registration: User account registration and management functions are not included in this version and will not be tested.

# 5. Approach

The testing approach for the “Flight Booking App” Web Application will involve comprehensive test case design, both manual and automated testing, test data management, test environment setup, defect tracking and management, and test reporting and metrics. The focus will be on validating the functionality, usability, performance, security, compatibility, and integration aspects of the application.

The project is using an agile approach, with weekly iterations. At the end of each week the

requirements identified for that iteration will be delivered to the team and will be tested.

The testing activities will follow a planned schedule and will be supported by appropriate resources and tools. The Test Strategy Plan will serve as a guiding document for the testing process.

## 5.1 Automation testing

Automation testing will be used for regression testing and for processes that can be tested automatically.

## 5.2 Meetings and Organizing processes

* Product Backlog: A list of tasks required for the current build.
* Sprint Planning: A team meeting in the SCRUM framework to discuss plans for the upcoming sprint.
* Sprint Backlog: A list of tasks assigned for the current sprint.
* Daily Meeting: A brief daily gathering where the team shares progress, thoughts, and outcomes of their work.
* Sprint Review: A meeting conducted at the end of a sprint where team members present and discuss the final results.
* Sprint Retrospective: A session where team members discuss challenges, identify blockers, and suggest improvements for future sprints.

# 6. Test strategy

## 6.1 QA role in test process

Test Planning:

Collaborating with stakeholders to define the scope and objectives of testing, identifying the appropriate test levels, and developing a test plan.

Test Design:

Create test scripts, test cases, and test data based on requirements and specifications in TestRail.

Preparation of the traceability matrix:

QA prepares a test matrix that maps the test cases to the relevant requirements. This will ensure that the requirements are covered.

Execution of test cases:

- Test cases will be executed by the appropriate QA based on developed scenarios, test cases and test data.

Test result (actual result, pass/fail):

QA will log defects/bugs in Jira found during the execution of the test cases.

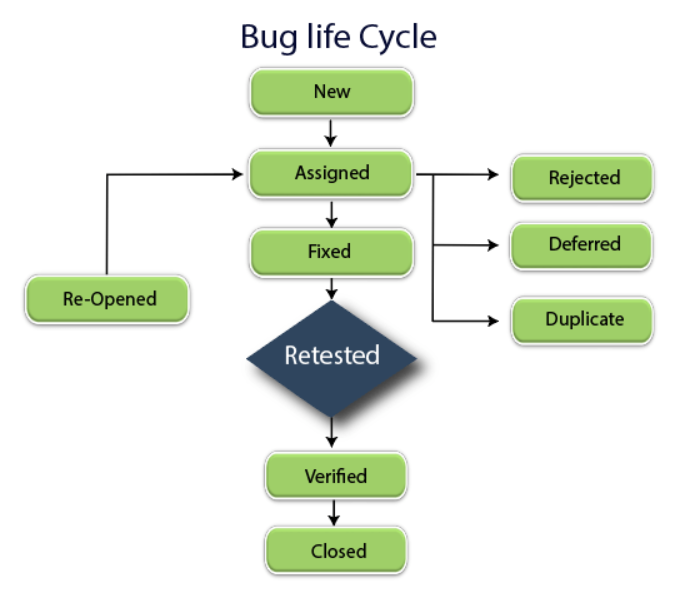
QA will then notify the relevant developer of the defect/bug.

Deployment/Delivery:

After correcting all identified defects and re-testing without identifying new defects, the report will be sent to the customer's email address.

## 6.2 Bug life cycle

All the issues found while testing will be logged into Jira.



## 6.3 Types of testing

According to the functions and capabilities of the mobile web application for booking airline tickets, the following types of testing will be applied to the project:

1. Functionality testing:

* checking the correctness of selecting the departure city and arrival city from the drop-down lists;
* checking the validity of the selected departure date;
* correctness of the number of passengers (adults, children, elderly);
* choosing a class (economy, business, first class);
* testing the successful flight search.

1. User interface testing: Checking the usability of the web application on mobile devices, the correct display of the interface on different displays and with different screen sizes.
2. API testing: Checking the connection with the external API, obtaining up-to-date flight data, checking the correctness of pulling flight information into the application.
3. Load testing: Performing tests to evaluate the performance and speed of the system when searching for flights and booking, in particular in conditions of a large number of simultaneous requests.
4. Regression testing: This type of testing will focus on retesting previously implemented functionality to ensure that any new changes or updates have not resulted in unwanted problems or errors.
5. Compatibility testing: Since it is a web application for mobile devices, compatibility testing involves checking the compatibility of the application with different mobile platforms, particularly Android and iOS, to ensure that it functions correctly and displays properly on different devices with different screen sizes and resolutions. Emulators will be used to simulate different devices during testing.
6. Acceptance testing:

The purpose of user acceptance testing is to confirm that the system is designed to meet the defined user requirements and is ready for operational use.

## 6.4 Severity List

| **Severity** | **Severity Description** |
| --- | --- |
| Critical | The module/product crashes or the bug causes non-recoverable conditions. System crashes or database or file corruption, or potential data loss, program hangs requiring reboot. |
| High | Major system component unusable due to failure or incorrect  functionality. Bugs cause serious problems such as a lack of functionality, or insufficient or unclear error messages that can have a major impact on the user, prevent other areas of the app from being tested, etc.  Bugs can have a work around, but the work around is inconvenient or difficult. |
| Medium | Incorrect functionality of component or process. There is a simple work around for the bug. |
| Minor | Documentation errors |

## 6.5 Priority list

| **Priority Level** | **Priority Description** |
| --- | --- |
| High | The damage caused by this incident increases rapidly and can result in extended downtime or greater business impact over time. The issue can grow into a higher severity issue if not handled in a timely manner. |
| Middle | The problem needs to be fixed, but there are other higher priority tasks. The priority is reduced due to other blocking factors. A workaround may or may not be necessary, depending on the customer's needs. |
| Low | No or very minimal impact on the current system. Does not require a patch or workaround. |

# 7. Item Pass/Fail Criteria

Entry criteria:

* The test environment is set up and ready to execute the tests.
* The test data is properly prepared and matches the test scenarios.

Exit criteria:

1. Requirements:

100% coverage of the requirements has been achieved.

2. Defects:

* All blocking defects are fixed, no critical defect is in an open state.
* All high priority defects are identified and fixed.
* Very few medium priority defects are open and have issues resolved.
* Very few open low priority defects that do not affect the use of the software (5% of the total).

3. Test coverage:

Test coverage of at least 95% is achieved.

4. Cost:

When the budget is spent.

5. Schedule Adherence:

When the schedule is met.

# 8. Suspension Criteria and Resumption Requirements

## Suspension criteria

1. Critical defects: defects that significantly affect the functionality, usability or security of the program.
2. Blocking issues: blocking issues or obstacles that prevent the effective progress of testing, such as the unavailability of required test environments.
3. Significant bugs: if serious and widespread defects are found that make the application unusable or pose significant risks to the business or end users.
4. Resource constraints: lack of key testing personnel, hardware or software resources required for testing.

## Resumption Requirements

1. Defect resolution: Once critical defects or bugs that stop the application from working are resolved, testing can resume. The application must be retested to ensure that the fixes have been successfully implemented and that the relevant functionality is now functioning properly.
2. Resolve locking issues: If any blocking issues or obstacles that prevented testing have been resolved, the necessary steps should be taken to remove those obstacles and allow testing to resume.
3. Availability of resources: If the reason for the suspension was a lack of resources, the necessary resources for testing should be made available before testing can resume. This may involve coordinating with relevant stakeholders, acquiring the necessary hardware or software resources, or adjusting the test schedule to accommodate the available resources.

The decision to suspend or resume testing is made by the test manager or project stakeholders based on the severity of the issues identified and their impact on the testing process.

# 9. Test Deliverables

1. Test Plan: A document that describes the overall testing strategy, goals, test objects, and the distribution of tasks among team members.
2. Test Cases: Specific test scenarios that describe the test steps and expected results for different functions and components of the application.
3. Requirement Traceability Matrix (RTM) - it ensures that all requirements are covered by the appropriate test cases and facilitates the tracking and resolution of defects.
4. Defect Reports: Reports on identified defects, including a description of the problem, priority, status, and additional details needed to fix the defect.
5. Test Summary Report: A summary report of test results, including the number of tests passed, the number of defects found, and the overall quality score of the product.
6. Test Environment Setup: Instructions and configurations required to set up the test environment, including hardware, software, and network configurations.
7. Test Completion Report: A report that summarizes the testing activities performed, highlighting the achievements, challenges, and lessons learned during the testing process.

# 10. Resource and Environment needs

## 10.1 Resource

| **Process** | **Tools** |
| --- | --- |
| Creating test cases | TestRail |
| Tracking test cases | TestRail |
| Execution of test cases | Manual testing and automation in IntelliJ IDEA |
| Test case management | TestRail |
| Defect management | Jira |
| Test reporting |  |
| Check list creating | Microsoft Excel |
| API testing | Postman |
| Android emulator | Android Studio |
| iOS emulator | Xcode's iOS Simulator |
| Load testing | Apache JMeter |

## 10.2 Environment needs

1. Devices with the Android operating system:

* Android OC (11 and 12).
* Samsung Galaxy, Google Pixel, OnePlus and HTC.
* Browsers: Google Chrome, Mozilla Firefox and Opera (latest updates).

2. Devices with the iOS operating system:

* iOS (14 and 15).
* iPhone and iPad
* Browsers: Safari, Google Chrome and Mozilla Firefox (latest updates).

3. Testing on different displays:

Use emulators to test on different screen sizes and resolutions + Google Chrome(Dev Tools)

4. Network Connectivity: Test different network conditions, including 3G, 4G, and Wi-Fi, to evaluate application performance and responsiveness.

# 11. Responsibilities

| **Role** | **Staff Member** | **Responsibilities** |
| --- | --- | --- |
| Project  Manager |  | 1. Acts as a primary contact for development and QA  team.  2. Responsible for Project schedule and the overall  success of the project. |
| QA Lead |  | 1. Participation in the project plan creation/update process.  2.Planning and organization of the test process for the release.  3.Coordinate with QA analysts/engineers on any  issues/problems encountered during testing.  4.Report progress on work assignments to the PM |
| QA |  | 1. Understand requirements  2. Writing and executing Test cases  3. Preparing RTM  4. Reviewing Test cases, RTM  5. Defect reporting and tracking  6. Retesting and regression testing  7. Bug Review meeting  8. Preparation of Test Data  9. Coordinate with QA Lead for any issues or problems  encountered during test preparation/execution/defect  handling. |

# 12. Schedule

| **Task** | **Volume of work** | **Starting date** | **Expiry date** |
| --- | --- | --- | --- |
| Test plan creation | 16 hours | 25.05.2023 | 30.05.2023 |
| Create test basis | 24 hours | 31.05.2023 | 2.06.2023 |
| Test Execution | 24 hours | 03.06.2023 | 6.06.2023 |
| Regression testing |  |  |  |
| Analysis of testing |  |  |  |
| Summing-up |  |  |  |

# 

# 13. Risks and Contingencies

Risk: Project timeline constraints may be exceeded.

Contingency: Prioritize critical tasks, streamline processes, and allocate additional resources if possible to meet the established timeline. Communicate with stakeholders and seek their input to adjust the plan accordingly.

Risk: Changes in customer requirements or scope may affect the project timeline.

Contingency: Maintain a well-defined change management process to assess and document customer-requested changes. Evaluate the impact of changes on the timeline and engage in timely discussions with the customer to manage expectations. Adjust the project plan and schedule accordingly, ensuring alignment with the customer's priorities and goals.

# 14. Approvals

|  | **Project Manager** | **QA Lead** |
| --- | --- | --- |
| Name |  |  |
| Signature |  |  |