

# Test plan

## 1. Introduction:

The main goal of this test plan document is to give a comprehensive test strategy and execution plan for our upcoming tourism application. The document aims to provide team members with a thorough grasp of the test's goal, scope, and expected outcomes. It is a significant resource for the development team, test team, and relevant stakeholders, providing insightful advice and reference material.

## 2. Test Objectives:

The test objectives of this plan are established to ensure comprehensive validation of the tourism application, ensuring its adherence to high-quality standards. The primary goals and expected outcomes are as follows:

### 2.1 Functionality Validation:

Thoroughly verify that all features and functionalities of the application align with the intended design and functional requirements.

Validate the seamless functioning of critical user interactions, including registration, login, and reviews.

Conduct rigorous assessments to ensure the accuracy, consistency, and reliability of data presented by the application.

### 2.2 Performance Testing:

Measure and analyze the application's response time for various operations, such as page loading and transaction processing, to ascertain optimal performance.

Evaluate the application's capacity to handle concurrent user requests and assess its scalability under different load scenarios.

### 2.3 Compatibility Testing:

Validate the compatibility of the application across diverse operating systems, web browsers, and mobile devices.

Verify that the application exhibits consistent functionality and appearance across multiple platforms to ensure a seamless user experience.

### 2.4 Reliability and Stability Assessment:

Identify and rectify any software defects, including crashes, freezes, or unexpected behavior, to enhance the application's stability and reliability.

Conduct comprehensive stress testing to evaluate the application's resilience and ascertain its performance limits under heavy user loads.

### 2.5 Security Verification:

Assess the effectiveness of the application's security measures, encompassing aspects such as authentication, data encryption, and protection against potential vulnerabilities.

Perform thorough penetration testing to identify and address any security loopholes or weaknesses, providing actionable recommendations for improvement.

## 2.6 Usability and User Experience Evaluation:

Gather valuable feedback through user testing and surveys to evaluate the application's usability and user experience.

Identify areas for improvement in terms of user interface design, navigation, and overall user satisfaction, ensuring an intuitive and engaging user experience.

By attaining these test objectives, we will gain invaluable insights into the functionality, performance, dependability, compatibility, security, and user experience of the tourism application, thus guaranteeing its adherence to rigorous quality benchmarks.

## 3. Test scope:

The test plan encompasses the following scope of testing activities, including system modules, functionalities, platforms, environments, and user roles:

### 3.1 System Modules and Functionalities

Identify all system modules and functionalities that require testing to ensure comprehensive coverage of critical and core components. This may include, but is not limited to, the following aspects:

User registration and login

Travel destination search and information retrieval

User reviews and ratings

Map and navigation functionalities

### 3.2 Platforms and Environments

Operating systems: iOS, Android, Windows

Browser: Chrome

Devices: mobile phone, desktop computers

### 3.3 User Roles

We need to identify the various user roles that should be included in the testing process to assess different user needs and functionality permissions. This may involve roles like regular users, registered users, administrators, and more.

By clearly defining the scope of our testing, we can make sure that we take into account and cover all essential system modules, functionalities, testing platforms, environments, and user roles. This approach will help us achieve a comprehensive and effective test coverage.

## 4. Test Strategy:

### Test Strategy

The test strategy document provides a framework for testing methodologies and approaches, including the selection of test types, their prioritization, test coverage,

## Team 10 -test plan

allocation of testing resources, utilization of testing tools, and establishment of the testing environment.

### Test Types:

Our testing approach encompasses three primary test types: API testing, frontend functionality checklist, and stress testing. These tests collectively ensure a comprehensive validation of the tourism application.

### Test Prioritization:

In order to prioritize our testing efforts effectively, we will consider the criticality and impact on the user experience. Our focus will initially be on high-priority areas, starting with API functionality, followed by the frontend functionality checklist, and concluding with stress testing.

### Test Coverage:

Our objective is to achieve extensive test coverage by thoroughly examining the critical functionalities, system modules, and user scenarios. This includes validating user interactions, data inputs, handling edge cases, and addressing error conditions across both the API and frontend components.

### Testing Resources:

To ensure the efficient execution of these tests, we will allocate dedicated resources for test planning, design, execution, and defect tracking. Our testing team will be responsible for conducting the tests and ensuring their accuracy.

### Testing Tools:

To streamline the testing process, improve efficiency, and enhance test accuracy, we will employ a combination of manual testing techniques and appropriate automated testing tools.

### Testing Environment:

We will establish a dedicated testing environment that closely replicates the production environment. This entails configuring the necessary hardware, software, networks, and settings to simulate real-world conditions during the testing phase.

By developing a well-rounded test strategy that includes API testing, a frontend functionality checklist, and stress testing, we can establish a structured and effective testing procedure that comprehensively addresses all vital components of the tourism application.

## 5. Test Schedule:

Test Start Date: March 23, 2023

Test End Date:

Phase 1: Test Planning and Preparation

Expected Duration: March 25 – April 3, 2023

Milestone: Completion of test plan, test strategy, and test cases

Phase 2: Functional Testing

Expected Duration: April 20 – April 30, 2023.

Milestone: Completion of functional testing, identification of defects

Phase 3: Performance Testing

Expected Duration: May 1<sup>st</sup> - May 6, 2023

Milestone: Completion of performance testing, analysis of results

Phase 4: Compatibility Testing

Expected Duration: May 1<sup>st</sup> - May 7, 2023

Milestone: Completion of compatibility testing, validation of platform compatibility

Phase 5: Reliability and Stability Testing

Expected Duration: May 1<sup>st</sup> – May 7, 2023

Milestone: Completion of reliability and stability testing, resolution of critical issues

Phase 6: Usability and User Experience Evaluation

Expected Duration: May 1<sup>st</sup> – May 7, 2023

Milestone: Completion of usability testing, gathering user feedback

Phase 7: Final Testing and Bug Fixing

Expected Duration: May 5 – May 10, 2023

Milestone: Completion of final testing, resolution of remaining defects

Test Completion Date: May 10, 2023

## 6. Test Tasks

### Test Planning and Preparation

Description: Develop a test plan outlining the testing approach, objectives, and scope. Prepare test environments and test data.

Owner: Testing team

Expected Completion Date: April 3, 2023

### Functional Testing

Description: Execute test cases to validate the functional requirements of the application. Verify the correctness of user interactions, data inputs, and system responses.

Owner: Testing Team

Expected Completion Date: April 30, 2023

### Performance Testing

## Team 10 -test plan

Description: Conduct performance tests to assess the application's response time, scalability, and resource usage under different load conditions. Identify and address performance bottlenecks.

Owner: Back-end Team

Expected Completion Date: May 6, 2023

### Compatibility Testing

Description: Verify the compatibility of the application across different platforms, browsers, and devices. Ensure consistent functionality and appearance.

Owner: Back-end Team & Front-end Team

Expected Completion Date: May 7, 2023

### Usability Testing

Description: Assess the application's user-friendliness, ease of navigation, and overall user experience. Gather user feedback and identify areas for improvement.

Owner: Testing team

Expected Completion Date: May 7, 2023

### Bug Tracking and Reporting

Description: Document and track identified defects, prioritize them based on severity, and communicate them to the development team for resolution.

Owner: Testing Team

Expected Completion Date: May 5 – May 10, 2023

## 7. Resource Requirements

To successfully conduct the testing activities, the following resources are required:

### 7.1 Hardware.

mobile phone, desktop computers

### 7.2 Software.

Operating systems: Windows/ macOS/ Linux

API testing: Postman/ IntelliJ

Load testing: Apache/ JMeter

Functional Testing: Word

### 7.3 Human resources

## 8. Potential Risks:

In our project, it is necessary to identify potential risks and concerns that may effect our testing plan. We are committed to proactively anticipating and addressing these risks through a thorough risk assessment, even though we are aware that we do not have the same level of resources and knowledge as a professional testing team. The

following are some of the potential risks we have identified, along with corresponding mitigation strategies:

**Limited testing resources:** As student testers, we can face barriers to accessing testing environments, tools, and equipment. We will give priority to necessary testing tasks that can be carried out with the resources at hand in order to reduce this risk. We will also look into complementary approaches and open-source tools that can aid our testing efforts.

**Time constraints:** As students, we may experience time limitations and conflicting priorities. We'll develop a practical testing schedule and set aside specific time for testing activities to handle this risk. To increase productivity, we'll place a strong emphasis on time management and encourage open communication among the members of our testing team.

**Lack of knowledge:** Due to our lack of professional experience, there may be obstacles we must overcome in certain domains that call for specialised knowledge and abilities. We will use online resources, look for academic advice, and contact subject matter experts as necessary to address this risk. In order to improve our testing abilities, we are also committed to making time for self-learning and training.

By proactively identifying these potential risks and implementing appropriate mitigation strategies, we aim to minimize their impact on our testing plan.

## 9. Execution Plan:

In our project, we have formulated the following execution plan:

### 9.1 Test Sequence:

The test cases will be ranked in order of importance and influence on the application. The sequential execution of tests will guarantee a logical flow and reduce dependencies.

### 9.2 Test Data:

In order to cover a range of scenarios and inputs, we will generate a wide variety of test data.

This could entail producing bespoke datasets for certain test situations, using pre-existing datasets, or developing sample data.

Creating a test environment

The test environment will be set up to closely mirror the production environment. This include configuring the necessary gear, software, network settings, and pertinent test data.

### 9.3 Test Execution:

Each test case will be painstakingly carried out in accordance with the prescribed test sequence.

The test results, including observed behaviour, any found flaws, and variations from anticipated results, will be recorded.

Reports of Tests:

To record the outcomes, we shall produce thorough test reports following each test execution.

The reports will contain data like the test cases that were actually run, observed behaviour, flaws, and accompanying documentation like screenshots or logs.

We make an effort to guarantee a methodical and meticulously recorded test execution procedure by following this execution strategy. This strategy makes it possible for team members to work together effectively, makes it simple to monitor the testing process, and helps our testing efforts succeed as a whole.

## 10. Evaluation Criteria

### 10.1 Test Pass Criteria:

A test case is deemed successful if the behaviour is consistent with the anticipated results that we specified in our test specs.

There shouldn't be any serious bugs or requirements violations in the programme, and it should function properly.

### 10.2 Test Failure Criteria:

If the observed behaviour and the predicted results diverge, the test case is labelled as failed.

Functional concerns, performance problems, compatibility problems, security flaws, and other substantial departures from the requirements can all result in failures.

### 10.3 Error Categorization:

Based on how serious they are and how they affect how well the programme works and how the user interacts with it, we will classify any issues that have been found.

There are a number of different types of faults that are frequently found, each with a different level of severity and need for a remedy, such as critical errors, significant errors, minor errors, and cosmetic issues.

## 11. Deliverables

As part of our team's commitment, we will provide a set of deliverables as outlined in our testing plan. These deliverables include:

11.1 Checklist: A prepared checklist that encompasses the essential testing tasks and criteria. This checklist will serve as a comprehensive guide for executing and documenting the testing process with utmost precision.

11.2 Loaded Test Report: A crafted report that presents a thorough analysis of the system's performance and stability under varying load conditions. This report will provide in-depth insights into the system's behavior, identifying any potential bottlenecks or performance issues.

In order to effectively document and communicate the results of our testing, we are dedicated to completing the delivery of these crucial documents as soon as possible.