Info 6068 Capstone

**Topic**

Test Strategy

Group 3 [Test Tribe]

### Kuzi Onyibe 1050553

### Opemipo A. Quadri 1171053

### Olorunsuyi Bright 1168301

### Magandy Opara Uzoma 1157223

### Obinna Charles Iheanacho 1147930

**Software and Information Systems Testing**

**01st February 2023**

### Amendment History:

|  |  |  |
| --- | --- | --- |
| Version | Date | Amendment History |
| V 1.0 | 23/1/2024 | First draft |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

### Reviewers:

This document must be reviewed by the following:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Signature | Title / Responsibility | Date | Version |
| Opemipo A. Quadri |  | QA | 25/1/24 |  |
| Olorunsuyi Bright |  | Scribe | 25/1/24 |  |
| Kuzi Onyibe |  | Project Manager | 25/1/24 |  |
| Magandy Opara Uzoma |  | QA | 25/1/24 |  |
| Obinna Charles Iheanacho |  | QA | 25/1/24 |  |

### Approvals:

This document must be approved by the following:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Signature | Title / Responsibility | Date | Version |
| Susmita Haldar |  | Project Sponsor | 2/2/24 |  |

### Distribution:

### Related Documents:

These documents will provide additional information.

|  |  |  |  |
| --- | --- | --- | --- |
| Ref no | Doc Reference Number | Title | Version |
| 001 |  | Travel and Tourism Management System | 1.0 |

### Glossary of Terms:

List any terms used in this document.

|  |  |  |
| --- | --- | --- |
| Term | Acronym | Definition |
| Software Development Life cycle | SDLC | The various stages of a software in the development process |

# Table of Contents

[Table of Contents 3](#_Toc450572426)

[1. Introduction 4](#_Toc450572427)

[1.1. Objectives 5](#_Toc450572428)

[1.2. Scope 5](#_Toc450572429)

[2. Roles and Responsibilities 5](#_Toc450572430)

[3. Testing Overview 6](#_Toc450572431)

[3.1. Test Lifecycle 6](#_Toc450572432)

[3.2. Test Approach 6](#_Toc450572433)

[3.3. Standards 7](#_Toc450572434)

[3.4. Test Stages 8](#_Toc450572435)

[3.5. Reviews and Inspections 9](#_Toc450572436)

[3.6. Test Documentation 9](#_Toc450572437)

[3.7. Test Execution 9](#_Toc450572438)

[3.8. Entry & Exit Criteria 11](#_Toc450572439)

[3.9. Test Results Capture 12](#_Toc450572440)

[3.10. Progress Reporting 12](#_Toc450572441)

[4. Test Data 13](#_Toc450572442)

[5. Testing Environments 13](#_Toc450572443)

[5.1. Specification 13](#_Toc450572444)

[6. Testing Tools 14](#_Toc450572445)

[6.1. Test Management Tools 14](#_Toc450572446)

[6.2. Test Automation Tools 14](#_Toc450572447)

# Introduction

Travel Application is the name of the proposed software system, which automates the functions of a real travel agency. This web-based application will function as a significant repository for all relevant travel and tourism-related information, which will be accessible to users upon request. The system will consist of five distinct modules, each with a unique set of features. Since the system will basically be an integration of all these modules, it is essential that each one undergoes thorough testing until the following conditions are met:

ADMINISTRATOR MODULE:

This module provides administrator-related functionality. The administrator manages all information and has access rights to add, delete, edit, and view the data related to places, travels, routes, bookings, etc.

TRAVELS MODULE:

This module provides the details of various travel agencies. A user can select the appropriate agency depending on convenience and accessibility.

ROUTES MODULE:

This module provides information related to various routes connecting sources and destinations. For each route, information such as source, destination, fare, reservation details, pick-up points, etc are provided. Only the administrator can add, delete, edit, and manage the data. Users can only view the information.

RESERVATIONS MODULE:

This module provides functionalities that allow a user to book tickets or cancel previously booked tickets. The module maintains the details of all reservations made so far and allows the administrator to either confirm or reject the bookings.

FEEDBACK MODULE:

Users of this application can post their opinions, complaints, and suggestions regarding this portal and services to the administrator. Accordingly, the administrator can take various steps to act on the complaints and suggestions.

Since the testing is meant to be iterative and start concurrently with the software development process, it will typically follow an agile methodology. This implies that ongoing stakeholder collaboration and continuous improvement at every level of the development process would be required for the tests. As a result, the requirements are flexible and can be altered as the project moves forward with the specific consent of the sponsor and other relevant parties. Until the exit criteria are satisfied, this test will be conducted in our setting. The non-functional aspects of the test, such as usability, performance, reliability, etc., would be the last stages. Before the software is released to the production environment, all of these non-functional aspects must be considered satisfactory.

## Objectives

The objective of this document is to provide a detailed description of every method required to carry out a comprehensive test of the travel application software. After the project sponsor gives their approval, the testing team will employ the plan that has been drafted to satisfy the business requirement and is included in the following sections. Once all test plans have been thoroughly reviewed and approved by all stakeholders, the test execution will commence. Until all requirements provided by the customer are satisfied and the exit criteria, as specified in this document have been met, the test will be repeated.

## Scope

This Test Strategy will cover the following:

* Individual modules of the travel application would be tested using unit/component testing.
* The modules of the travel application will be tested using Integration testing to understand their interaction with one another.
* Once the modules are integrated, systems testing will be performed to test the entire system as a whole.
* Then User acceptance testing will be performed on the travel application.
* Also, Non-functional testing will be performed on the travel application which includes performance testing, stress testing, reliability testing, load testing, etc.

# Roles and Responsibilities

The Roles and Responsibilities are as follows:

The Tester:

* In charge of creating the test cases
* In charge of carrying out tests and making sure the regression and confirmation tests are all completed satisfactorily.
* Making sure the software is built in compliance with the business requirements.
* Continual testing until the exit criteria are satisfied.

Project Sponsor:

* In charge of project financing
* approves ideas and modifications that impact the project.
* Functions as a key decision-maker for the project

Project Manager:

* In charge of creating schedules, writing budgets, and organizing and planning the tests.
* Monitor every test execution, identify potential hazards, and take appropriate action.
* Keep the project sponsor informed and update the other team members on a regular basis.
* Assign tasks and set deadlines.

Scribe (or recorder): documents all the issues, problems and open points that were identified during meeting.

# Testing Overview

## Test Lifecycle

Software applications are tested systematically to make sure they meet requirements and are error-free using the Software Testing Life Cycle (STLC) methodology. Every stage of the process has distinct goals and deliverables, and it is organized into a sequence of steps or phases. The software must be of the highest caliber, dependable, and satisfy end users' requirements; this is ensured by the STLC.

For the STLC we have about six major phases listed below are in the actual order of activities.

The stages of the STLC include.

* Test Planning
* Test Analysis,
* Test Design,
* Test Environment Setup,
* Test Execution,
* Test Closure

## Test Approach

Test approach is the implementation of the test strategy in a software project that defines how testers will carry out [software testing](https://www.lambdatest.com/learning-hub/software-testing), along with throwing light on strategy and execution to carry out different tasks.

For this Travel Application the test approach considered are:

* Water fall methodology
* Agile methodology

## Standards

The following severity levels are proposed for defects that are found during the Test Execution:

|  |  |  |
| --- | --- | --- |
| Severity Level | Priority Level | Description |
| Critical | P4 | A defect that completely blocks the testing of the product/feature is a critical defect. If the application crashes or becomes unusable/not able to proceed further, the defect could be classified under critical severity. |
| Major | P3 | Any Major feature implemented that does not meet the customer’s requirements/use case(s) and behaves differently than expected, can be classified under Major Severity. |
| Minor | P2 | A medium defect appears when the product or application doesn’t meet certain criteria or still exhibits some strange behaviors, however, the functionality as a whole is not impacted and can be classified under Minor Severity. |
| Low | P1 | Any corrective defects including spelling mistakes alignment issues or font casing can be classified under Low Severity. |

## Test Stages

Each test stage is a discrete form of testing with its own objectives, methods and requirements coverage and therefore a set of its own test scripts.

A coverage matrix of all the Test Stages / Test Areas to be covered in each Test Release is appended below.

| Test Areas/ Test Type | Unit testing | Integrated testing | Systems testing | Acceptance testing |
| --- | --- | --- | --- | --- |
| Functional |  |  |  |  |
| Non-Functional |  |  |  |  |
| Business Processes |  |  |  |  |
| Volume |  |  |  |  |
| Performance |  |  |  |  |
| Security (including Penetration Testing) |  |  |  |  |
| Data Protection |  |  |  |  |
| Usability |  |  |  |  |
| Interface Tests |  |  |  |  |
| Installation & Configuration |  |  |  |  |
| Systems & Service Management & Service Level Reporting |  |  |  |  |
| Network Worthiness |  |  |  |  |
| Disaster Recovery |  |  |  |  |
| Helpdesk Tools & Processes |  |  |  |  |
| Management Information Reporting |  |  |  |  |
| Audit |  |  |  |  |
| Resilience |  |  |  |  |
| Capacity Planning |  |  |  |  |
| Data Migration |  |  |  |  |
| Training Processes, Contents & Effectiveness |  |  |  |  |
| Cutover & Fallback & Go-Live Simulation |  |  |  |  |
| Back-up, recovery, journaling |  |  |  |  |
| Operations Support Processes |  |  |  |  |
| Commissioning |  |  |  |  |

## Reviews and Inspections

### Reviews

Each Test Stage will be run according to the Test Plan and Test Specification applicable to that stage. Each document will be reviewed internally and submitted to the business (professor) for review and approval.

The review types are:

* Inspection
* Walkthrough

### Inspections and Walkthroughs

**Inspections**

It is a formal type of review where a person has to go through a defined set of instructions in order to find defect/defects. There can be a number of reviewers involved in this type of reviewing.

**Walkthroughs.**

It is an informal method of reviewing issues in a software program, taking less time to complete. Users walk through the application and review any defects or deviations.

## Test Documentation

Identify all the Test documentation that will be delivered during each of the Test Phases and test cycles.

As an example, this is set out as the table below: add any additional relevant documents

| Document | Phase and cycle |
| --- | --- |
| Test Strategy | Test initial phase (test planning) |
| Test Plans | Test initial phase (planning phase)  Create Project Plan |
| Test Cases | Test Design |
| Test Specifications | Test design |
| Test Scripts | Test execution  Monitor the progress. |
| Test Report | Test execution  Record process execution and findings.  Document and review the data. |
| Report Defects | Defect Report Management |
| Test summary | Test closure. |

.

## Test Execution

Test execution is the process of running test cases on an application to determine its functional and non-functional parameters in relation to the requirements. Test planning and analysis frequently come before test execution.

The QA environment will be used to conduct this test. For this test, we would be keeping an environment that is specifically designated for it. During the planning and initiation phase of the project, there will be a smaller number of QA team members than the anticipated team size. However, because more QA will be required to run the tests, the team's capacity will likely be reached during the test execution phase.

The primary goals of this cycle are to find the critical defects that are blocking progress, fix the coding scripts, and get the complete project results.

### Recording Actual Results versus Expected Results

Any deviations that exist between the actual and expected outcomes must be noted recorded and stored.

Tests are considered successful when the actual result aligns with the expected result. Anything else It is considered a failure of the test and the deviations from expectations are duly recorded; in case the actual results do not match the expected result.

The result would be in a table with seven columns as follows:

* Test case ID
* Test case information
* Assumptions and Pre conditions
* Test data
* Steps of test executed
* Expected result and actual results
* Pass or fail.

### Escalation of Issues for resolution

In the escalation process there are various steps can be taken.

• Find the root of the problem and document it. Give it to the manager or a higher-ranking individual.

• If a software tester discovers a problem that needs to be reported to a higher authority, the tester will attempt to resolve it on their end first. If this is feasible, there won't be a need for an escalation; if the problem persists, it will be forwarded to a higher authority.

• The software tester sends the issue in a documented format after requiring some evidence filled with information before forwarding it to the other technical person.

• A software tester could identify who the issue needs to be reported to and why during the documentation process.

• Following document delivery, the tester will get in touch with the recipient to clarify the problem.

• After explaining, the recipient will approach a higher authority to find a solution.

• Once the methods are identified, the coder will work on the issue and attempt to resolve it.

### Test Execution Roles

Throughout the project's design, implementation, and execution phases, software testers will test the updated software.

Throughout the software development process, software testers develop various test designs, test cases, test procedures, and test data. In accordance with the selective phase, it also conducts the testing procedures. scan all of the data, compile all of the reports, and record the testing strategy.

### 

## Entry & Exit Criteria

The core high-level requirements for entry and exit criteria are the environment and the conditions that must be met before the application test can begin or be closed.

**Entry Criteria – Conditions for test commencement**

* Application Build Availability
* Test Environment Setup
* Test Data Availability
* Test Documentation Readiness
* Adequately Trained Test Team
* Requirements Review.

**Exit Criteria – Conditions for test closure**

* Test Execution Completion
* Defect Resolution.
* Functional and Non-functional Requirements Validation
* Regression Testing
* UAT (User Acceptance Testing) Completion
* Documentation Update
* Metrics Analysis
* Approval for Application Release

### Table of Entry and Exit Criteria

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STLC stage | Entry criteria | Activity | Exit criteria | Deliverables |
| Requirement Analysis | - Ensure functional and nonfunctional documents are available.  - Acceptance criteria defined. | - Acknowledge all the transactions in module.  - Look for the type of tests to be performed.  - Prepare requirement traceability matrix. | Test automation report log off by client. | Automation feasibility report. |
| Test planning | -Requirement traceability matrix.  - Documents required. | - Various testing approaches analyzing.  - test tool selection  - select the best approach for test | Approved test plan document. | Test plan and strategy document. |
| Test case development | -Test plan  - Automation analysis report. | -Create test cases and test designs.  - create test data. | Reviewed and signed test cases. | -Test cases.  -Test data. |
| Test environment setup | Environment set up plan | -hardware and software preparation.  -connectivity requirements finalization.  -Prepare environment setup checklist. | The environment setup is as per the plan and checklist.  -The test data set is complete. | The final environment setup is done. |
| Test execution | Test cases, test plan and scripts are set up. | -Perform the tests as per plan.  -update test plan and test cases. | All test cases were executed. | Test cases updates with actual results. |
| Test cycle closure | -Testing has been done.  -test outputs are available. | -Prepare test metrics on the test outputs.  -test final report.  -analysis of the test reports. | -test closing report reviewed and signed out by the client. | Test closure report. |

## Test Results Capture

Every stage of the test cases has test results that are recorded and saved by taking screen photos or using snipping tools.

An investigation column has been added to the Excel spreadsheet with the current test scenario, while the test difference between expected and actual results is being examined. According to work completion, it updates for subsequent processes.

## Progress Reporting

It is customary in the software testing process to create a set of test cases for every deliverable. The acceptance criteria for each module are represented by the test cases. Each test set includes the actions or steps that were taken to achieve the desired outcomes.

We store the relevant test cases and additional data for later use for each deliverable task in a separate Excel spreadsheet. The spreadsheet was loaded with a lot of data.

* Test case ID
* The screen ID
* The screen Area or Zone
* What action is to be performed
* Expected results.
* Obtained results.
* Result: passed, failed, not tested
* Description of any other scenarios
* Related defect tracking issue

For team members to assess, the final progress report can compile data on the active test cases in the form of graphs and numbers.

### Test Report

**Test Report** is a document which contains a summary of all test activities and final test results of a testing project. Test report is an assessment of how well the[Testing](https://www.guru99.com/software-testing.html)is performed. Based on the test report, stakeholders can evaluate the quality of the tested product and decide on the software release.

The test report contains many criteria which are mentioned below.

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Information** | **Test objective** | **Test summary** | **Defect** |
| Project Name | Test type | Test result | Description |
| Description | Test Purpose | Test failed | Priority |
|  |  | Test blocked | Status |

It also includes...

* Count of test cases performed.
* Count of the passed test cases.
* Count of the failed test cases.
* Pass percentage.
* Fail percentage.
* Other comments for more information.

Fore more details here is the link below.

<https://www.guru99.com/how-test-reports-predict-the-success-of-your-testing-project.html>

# Test Data

|  |  |
| --- | --- |
| **Test Type** | **Source of Test Data** |
| Functional Testing | Business Requirement Specifications, User stories, Use cases, Functional Specifications |
| Non-Functional Testing | Software Requirement Specifications |
| Structural Testing | Code, architecture, workflows and/or data flows |
| Change-related Testing | Software Requirement Specifications |

# Testing Environments

## Specification

### Identification of the physical components, the communications, the system and the middleware necessary

* System – PCs, Laptops
* Platforms - Windows, Mac OS
* Browsers - Google Chrome, Mozilla Firefox, Microsoft Edge, Safari
* Technology – Java 21
* Web Technologies – HTML, JavaScript, CSS
* Web Server – Tomcat 10.1.18
* Database – MySQL8.0

### Other software or supplies needed to support testing

* Eclipse license for test environment
* Apache subversion (SVN) for version control

### Security and access requirements to the test area and equipment

* To access server rooms and the test area, an employee card is required.
* It takes valid credentials to access the application that is being tested.

### Test tools and utilities required

The following tools are available for use

* Redmine for managing bugs;
* Jira for managing tests;
* Browser Stack for cross-browser testing.for manual testing:
* Selenium Webdriver
* Owasp;
* Apache Jmeter tool

### Any other testing needs

.

# Testing Tools

## Test Management Tools

* Jira

## Test Automation Tools

* Selenium Webdriver
* Apache Jmeter tool
* Selenium
* Owasp tool