

INFO 6068 Capstone
Test Strategy
[Zero to Hero]

Amendment History:

Version	Date	Amendment History
1.0	2024.01.29	First version completed
2.0	2024.01.30	Revised section 1.2, 3.2, 3.7
2.1	2024.01.31	Revised section 3.10, 4
2.2	2024.02.01	Revised section 5, 6, and added references
2.3	2024.02.02	Revised section 1.2, 3.2, 4, 5

Reviewers:

This document must be reviewed by the following:

Name	Signature	Title / Responsibility	Date	Version
Susmita Haldar		Project Sponsor		
Suvidha Desale	Suvidha Desale	Project manager	2024.02.03	2.3
Rupalben Patel	Rupalben Patel	Scribe	2024.02.03	2.3
Swapna Tambe	Swapna Tambe	Business Analyst	2024.02.03	2.3
Fan Yang	Fan Yang	QA lead	2024.02.03	2.3
Bidushi Kabir	Bidushi Kabir	Tester	2024.02.03	2.3

Approvals:

This document must be approved by the following:

Name	Signature	Title / Responsibility	Date	Version
Susmita Haldar		Project Sponsor		

Related Documents:

These documents will provide additional information.

Ref no	Doc Reference Number	Title	Version
1	1	Travel-and-Tourism-Management-System	1.0

Glossary of Terms:

List any terms used in this document.

Term	Acronym	Definition	
STLC	Software Test Life Cycle	A series of defined steps carried out during the testing process to ensure the achievement of software quality objectives.	
BVA	Boundary Value Analysis	A testing approach that employs boundary values to design test cases is known as the boundary value analysis black-box test design technique.	

Table of Contents

1. Introduction	5
1.1. Objectives	5
1.2. Scope	5
2. Roles and Responsibilities	6
3. Testing Overview	7
3.1. Test LifeCycle	7
3.2. Test Approach	9
3.3. Standards	11
3.4. Test Stages	12
3.5. Reviews and Inspections	13
3.5.1. Reviews	13
3.5.2. Inspections and Walkthroughs	13
3.6. Test Documentation	14
3.7. Test Execution	14
3.7.1. Recording Actual Results versus Expected Results	14
3.7.2. Escalation of Issues for resolution	15
3.7.3. Test Execution Roles	15
3.8. Entry & Exit Criteria	16
3.8.1. Table of Entry and Exit Criteria	16
3.9. Test Results Capture	16
3.10. Progress Reporting	17
3.10.1. progress Report	17
3.10.2. Test Reporting	17
4. Test Data	18
5. Testing Environments	18
5.1. Specification	18
5.1.1. Identification of the physical components, the communications, the system and middleware necessary	18
5.1.2. Other software or supplies needed to support testing	18
5.1.3. Security and access requirements to the test area and equipment	18
5.1.4. Test tools and utilities required	19
5.1.5. Any other testing needs	19
6. Testing Tools	19
6.1. Test Management Tools	20
6.2. Test Automation Tools	20
References	21

1. Introduction

TravelApp is a system that automates the activities of a travel and tourism office. The current system is inconvenient and time-consuming for customers. The proposed web application aims to address these issues by using recent technologies like Java and MySQL etc. And maintains a central information hub to make journey planning easier. The website that represents the planned solution seeks to synchronise different travel firms and expedite users' ticket purchase procedure. The Administrator Module, Travels Module, Routes Module, Reservations Module, and Feedback Module are the five components that make up the application structure. Two different login types are available in the TravelApp system: an admin login for administrative functions and a user login for everyday users.

The test strategy document is a detailed manual that outlines how we intend to test the TravelApp application. It describes the methodical techniques and methods that will be used all through the testing process. In order to verify the TravelApp application's functionality and dependability, the test strategy document clarifies the scheduled testing procedures and activities.

1.1. Objectives

The main objective of this testing procedure is to make sure that the TravelApp web-based application aligns with both users and the business expectations by providing a reliable, secure and user-friendly platform. This testing strategy document serves as an outline for the team, assisting in the planning, development and execution of various stages of STLC. The main responsibility of the team is to test both functional and nonfunctional aspects of the application.

The project test team will use this document as a guide to ensure that testing processes are followed consistently throughout the project. It will be updated as necessary to reflect the development of the project.

1. Team members' roles must be well defined to plan and carry out the testing procedure.
2. Describing the procedures and styles of testing.
3. Choosing the level of coverage and intensity.
4. Determining the outputs for every stage of testing.
5. Summarising the procedure used to execute the tests.
6. Establishing requirements for admission and departure.
7. Talking about the testing environment, tools, and test data.

1.2. Scope

The TravelApp Application Requirement document's specifications serve as the exclusive limits of the scope. After evaluating the requirements, the team will perform both functional and non-functional testing. The scope for this project is described as follow

In-Scope:

Test Plan Implementation: This involves keeping track of and carrying out the test plan during the project. It includes specifying the goals, schedule, and deliverables of the test. In the test case development phase, creating test scenarios, development of test cases and creation of

test scripts will be in the scope.

Functional testing: It involves testing that all the TravelApp modules like Administrator, Travels, Routes, Reservations, and Feedback follow the guidelines. To test the functionality of each module, these testing techniques will be followed by the team.

- Equivalence partitioning,
- BVA
- Combinatorial
- Use case testing.

Non-Functional testing – To verify the non-functional aspects of this web application such as performance and usability following testing types will be followed by team

- Load/Stress testing
- Security testing
- Usability testing
- Installation testing

Out of Scope: To ensure a practical and efficient testing procedure, the testing focus stays on meeting the established requirements within the given scope. So certain testing techniques will be out of scope for this project as follows.

- Unit Testing
- Integration Testing
- Cross-compatibility testing – As we are testing this application only for desktop devices the testing of mobile devices and other multiple browser testing is out of scope for this project.
- Localization Testing – As we are not testing this application for any specific location or language this testing technique is out of scope for this project.

2. Roles and Responsibilities

Name	Role	Responsibility
Susmita Haldar	Project Sponsor	Set the requirements and budget for software application.
Suvidha Desale	Project manager	Planning and Getting approvals from business stakeholders. Setting the goals for project development and supervising the team. Manage and develop a budget.

Rupalben Patel	Scribe	Document the meeting discussions and meeting minutes. Record the issue log. Proofread and present the test strategy, test plan and test cases.
Swapna Tambe	Business Analyst	Analysing the project requirements and preparing requirement documents. Facilitate the business requirements and communication to the project team.
Fan Yang	QA lead	Lead and guide the testing team for creating test plans and test reports. Responsible for the quality assurance of application.
Bidushi Kabir	Tester	Develop and execute the test cases based on RTM. Develop the test scripts and test cases and report executed results to QA lead.
UX team	Tester, Business Analyst	To test the usability of the software application

3. Testing Overview

3.1. Test LifeCycle

For this project, the team is testing a web-based application “TravelApp” that automates the activities of travel and tourism offices. The team will verify the functional and nonfunctional aspects of this application and determine the thorough requirements for the STLC. Our team is going to use agile methodology and work on this methodology team has determined 7 key phases for testing this application as follows:

1. Requirement analysis

This is the first and important step of the testing lifecycle. Requirement analysis

includes gathering all requirements from stakeholders and determining and analysing this data for the test planning phase.

Documentation:

- Team will review the business plan and determine requirements.
- Prepare RTM

2. Test planning

In this phase the team will plan the steps for testing including test tool selection, determining roles and responsibilities, schedules and test stages.

Documentation:

- Prepare Test strategy document.
- Prepare a Test plan.

3. Environment setup activities

This phase includes the installation of software, hardware, and testing tools. This phase ensures the AUT runs smoothly and ready for testing.

Main activities include:

- Setup test environment
- Smoke testing
- Test data

4. Development of test cases

During this phase, test cases are created for the Application Under Test (AUT) to verify the specified requirements of the application.

This phase provides.

- Test cases
- Test scripts

5. Test execution phase

This is the main phase of the testing lifecycle as it executes the test cases and validates the results to the specified requirements for testing the functionality of this application.

Documentation:

- RTM - Team updates the executed test cases results and status in RTM

6. Defect management

This phase includes the tracking of defects during test case execution, and the team documents and identifies issues across test stages. These defects will be reported to the developers for resolution.

Documentation:

- Defect Report

7. Test closure procedures

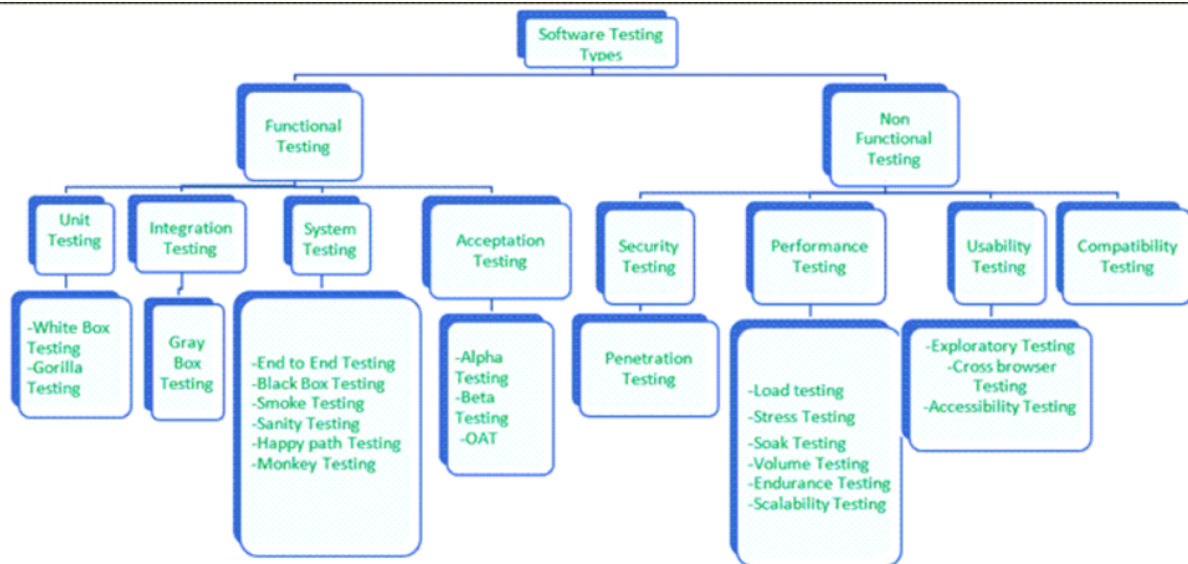
This is the final phase of the testing lifecycle, and it evaluates the software testing processes, activities and results and documents in the test closure report.

Documentation

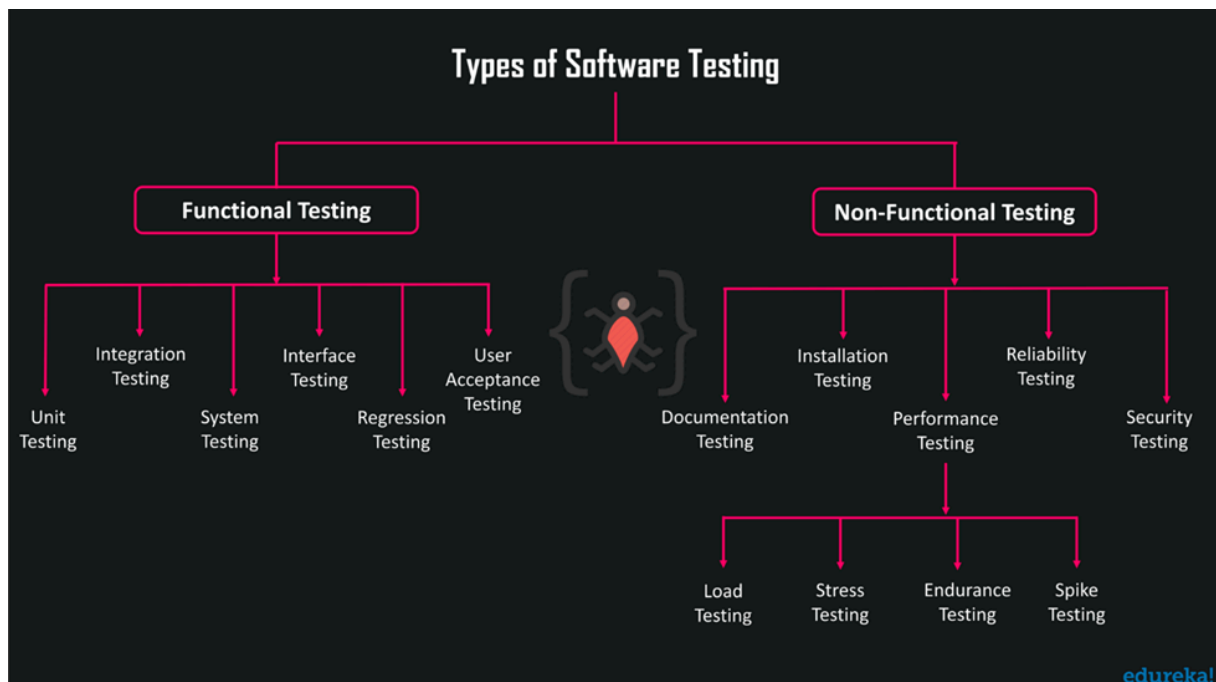
- Test closure report – Summary of all Test processes, test results, risks and issues

3.2. Test Approach

The Test Strategy document serves primarily as a roadmap for the TravelApp's testing methodology. Ensuring a thorough assessment of the system's functionality, usability, and dependability, it offers a well-defined path for both manual and automated testing.



(Figure 1)



(Figure 2)

The test approach will clarify details such as the project's implementation methodology. The TravelApp web application will be tested by our team using an Agile methodology. Our team will conduct testing regularly as it goes through multiple iterations by employing this incremental method. Additionally, increasing the quality of the product would require ongoing testing. Our team will perform manual testing as well as automation testing.

The project has a thorough testing methodology that aims to complete all tests in a limited

period. The following test practices will be implemented:

- **Manual Testing:**
In a simple word we can say that software can be tested manually, that is, without the use of any automated tools or scripts. To validate the TravelApp's general usability, functionality, and user experience, manual testing will be used. To verify that our TravelApp application satisfies user expectations, we will do acceptance testing, exploratory testing, and user interface testing. We will use Equivalence Partitioning and Boundary Value Analysis as techniques during the manual testing. We also will use State Transition Testing techniques that are often applied to both manual and automated testing.
- **Automated Testing:**
Automation testing includes writing and running automation test scripts. The team will be encouraged to do automation testing; this will help the team to execute more tests in less time. To improve coverage and efficiency, automated testing will be used. Automated testing will concentrate on scenarios involving repetition and regression testing. To speed up the testing process and guarantee consistency in test execution, automation scripts will be created utilising the proper technologies, such as Selenium WebDriver.

Data-driven testing:

It is crucial for our project to connect to an external data source in order to conduct data-driven testing. An XML, CSV, or Excel file, or a fully functional database such as MySQL, could be this source.

The following types of testing will be covered:

- **Functional Testing:**
All of the TravelApp's modules—the Administrator, Travels, Routes, Reservations, and Feedback modules—are included in the testing scope. All modules will be put through functional testing to make sure they work correctly and adhere to the criteria.
- **System Testing:**
System testing, which includes end-to-end testing, entails assessing the complete system in comparison to predetermined requirements. It's make sure to confirm complete operation during our Travel app application test that means the team will examine the travel application's overall functionality. For the travel App application to operate robustly and dependably, system testing is essential for spotting any possible problems and verifying the integrated functionalities.
- **Acceptance Testing:**
To make sure the TravelApp application meets user expectations. Customer, business, or client testing the software in real-time business scenarios is known as acceptance testing. Only when every feature and functionality functions as intended is software considered acceptable. Finally, software is tested before it is put into production; this stage is also known as User Acceptance Testing. The goal is to confirm that end users' expectations are met by the Travel App application. Acceptance testing seeks to ensure that the user experience is seamless and satisfying by concentrating on user expectations.
- **Regression Testing:**
To guarantee that updated modifications don't adversely affect current features when we do our testing. In order to verify that bug patches, new feature additions, deletions, or upgrades have no effect on the functioning application, regression testing makes sure that the application's unaltered features are tested. That means our team will perform regression testing whenever a new change occurs.

- **Non-Functional Testing:**
Testing for performance, security, regression, and usability is classified as non-functional testing. The user experience will be evaluated by usability testing, system responsiveness will be assessed by performance testing, code stability will be guaranteed by regression testing, and potential vulnerabilities will be addressed by security testing. The following non-functional testing will be covered:
 1. **Performance Testing:**
Through this testing, we can make sure to evaluate the system's response to different loads during our testing time (TravelApp application). With the aid of programs like Loader. IO, JMeter, or LoadRunner, performance testing evaluates the responsiveness and stability of an application under load.
 2. **Load Testing:**
A load equal to or less than the intended user base is applied during load testing to assess an application's responsiveness and stability. This testing also makes sure that to assess how the system behaves under harsh circumstances, performance testing will also incorporate stress testing. TravelApp's responsiveness and scalability will be evaluated through performance testing, guaranteeing that it operates at peak efficiency under various load scenarios.
 3. **Usability Testing:**
Usability testing evaluates a programme by looking at how it looks, feels, and is easy to use from the user's point of view. This testing is to make sure that we assess the user interface and the entire experience during our TravelApp application test.
 4. **Security Testing:**
Testing software, apps, or websites for vulnerabilities against both internal and external threats is known as security testing. In our project, the goal of security testing is to find and fix holes in the system, especially in the Administrator and Reservations modules where private information is kept up to date.
 5. **Installation Testing:**
The TravelApp application will be installed and configured correctly on various environments through installation testing. This guarantees a flawless and error-free installation procedure.

3.3. Standards

The following severity levels are proposed for issues arising throughout the Test Lifecycle:

Severity Level	Impact	Description
01	Low	The problems are minor or negligible.

02	Medium	The problem does require attention and needs to be fixed in the near future.
03	High	Issues need to be fixed in top-most priority as it has a major impact on the system.
04	Very critical	An error occurs on the basic functionality of the application and will not allow the users to use the system.

3.4. 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	Module testing	Integration testing	Function testing	Performance testing	Acceptance testing	Installation testing
Functional			X		X	
Non-Functional				X	X	
Business Processes						
Volume				X	X	
Performance				X	X	
Security (including Penetration Testing)				X	X	
Data Protection						
Usability				X	X	
Interface Tests						
Installation & Configuration						X
Systems & Service Management & Service Level Reporting						

Test Areas/ Test Type	Module testing	Integration testing	Function testing	Performance testing	Acceptance testing	Installation testing
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						

3.5. Reviews and Inspections

3.5.1. Reviews

A review will be conducted on documents to ensure correctness and completeness.

Below documents will be reviewed by peers.

1. Requirement Reviews
2. Design Reviews
3. Code Reviews
4. Test Plan Reviews
5. Test cases Reviews.

3.5.2. Inspections and Walkthroughs

Inspection is a most formal form of review. In which at least 3-8 people will sit together in the meeting (Moderator, Observer, Producer, reviewer, recorder, etc.).

Inspection will have a proper schedule which will be arranged by email.

Walkthroughs are informal reviews. In this process team members will give their opinions and suggestions. It's not pre planned and can be done whenever required. This will help to improve the system.

3.6. 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 planning and control
Test Plans	Test analysis and design
Test Specifications	Test analysis and design
Test Scripts	Test implementation and execution
Test Log	Test implementation and execution
Test Report	Evaluating exit criteria and reporting

3.7. Test Execution

Test execution comes under the STLC (software testing life cycle). It's the process of executing test cases developed by testers to compare the actual test result versus expected test result along with business requirements. Both functional and non-functional factors are tested during the execution process.

The test execution process will be performed at a Fanshawe college downtown campus under the guidance of the project Manager and sponsor.

3.7.1. Recording Actual Results versus Expected Results

The Test Reports will be kept up to date with the results of the tests carried out. Any supporting evidence required or provided and an indication as to whether the test step is considered to have "Passed" or "Failed". This will be collected in an Excel spreadsheet.

Test Execution Report Template

Test ID	Test cases and description	Test Data	Pre-conditions	Post-conditions	Expected Result	Actual Result	Test Criteria (Pass/Fail)	Severity (High, Medium, Low)
-	-	-	-	-	-	-	-	-

Test case Result Template

Test case Id#:
 Test case description:
 Executed by (Tester Name):
 Execution date:
 Test environment:

Test Steps	Expected Result	Actual Result	Pass/Fail	comments
1				
2				
3				
4				
5				

3.7.2. Escalation of Issues for resolution

Test escalation is the process where the issues found during the test execution phase are prioritised based on its severity level (High, Medium, Low). These issues will be forwarded to the Project Manager. The tester validates the severity of the defect, suggests the changes as per the requirements, raises a ticket using Jira and assigns the ticket to the development team. After making the necessary changes, the regression testing or retesting will be performed to make sure that all the issues are fixed. Issue log reports are reviewed and maintained by Test lead weekly.

3.7.3. Test Execution Roles

Role	Responsibility	Sign-off	Comments (Y/N)
Project Manager	Manages the project completely.	-	-
Test Lead	Deals with all testing procedures and makes sure that team has all the necessary resources to execute testing activities.	-	-

Testers	Deals with executing test cases and prioritize automation test cases for execution	-	-
Scribe	Note down the defects/issues that are found while executing the test cases and make a report to review.	-	-

3.8. Entry & Exit Criteria

3.8.1. Table of Entry and Exit Criteria

Phases	Entry Criteria	Exit Criteria
Requirement Analysis	Document Requirements for both Functional / Non Functional Testing.	RTM signed off, Released of Test Automation Report by the Client
Test Planning	Requirement of documents, Defined Scope, RTM	Test strategy document and test plan approved
Test case Development	Document Requirements, approved test plan	Approved test cases/scripts, Test Data reviewed, and Signed.
Test Environment Set Up	Environment plan set up, availability of architecture document and test design	Test data setup completion, environment ready for testing, Smoke test Successful.
Test Execution	RTM baseline, Environment set up ready, test plan/ test data available. Availability of integration testing	Test planned execution completed, defects fixed
Test Cycle Closure	Defects Fixed, Test execution completed and Review of Metrics and test results.	Approved Test closure report by the client

3.9. Test Results Capture

During the Requirements analysis stage, the team reviews the requirements to ensure they are clear and testable. Once the requirements are reviewed testers start working on Test Plan/ strategy deliverables and define the testing scope. After completing the planning stage review testers design the test cases, setting up the environment for test execution. After designing, the stage team executes the test cases and fixes the defects. In the test closure process the release of the product and the team performs the smoke testing to ensure the released product should be defect free.

3.10. Progress Reporting

3.10.1. progress Report

In which we have tested the application properly. We will perform several types of testing such as Functional/non functional, performance, usability, integration, and regression testing. Encountered the issue will be logged into defect reporting tools such as JIRA. Entry and exit criteria will be reported for each phase of testing along with test execution status. In this, test lead will share the daily progress stating on how many test cases are executed vs non-executed. In progress report, defect report will be included along with respective severity and priority with respective defect status.

3.10.2. Test Reporting

In which we have tested the application properly. We performed several types of testing such as Functional/non functional, performance, usability, integration, and regression testing. Encountered the issues, used different tools and techniques to detect the defects. Entry and exit criteria done successfully, test execution done properly, and product is ready to release.

Test Report

Executed	Passed		-
	Failed		-
	Total no Executed		-
In Progress			
Pending			
Blocked			
Test Planned			

Functions	Description	No of % Tc Executed	No of %TC Passed	No of Pending	Priority	No of Defect	Defect Fixed

4. Test Data

The following test data will be used throughout the test lifecycle.

Test type: Functional and Non-Functional	Source of test data
1. Regression Testing	Using our current test data
2. Performance Testing	By using Ad hoc Data
3. Security Testing	By using login information
4. Usability Testing	Performed according to client requirement
5. Positive Testing	Using valid data to ensure that the system works as expected
6. Negative Testing	Using invalid data to ensure that the system behaves as expected and throws an error

5. Testing Environments

5.1. Specification

5.1.1. Identification of the physical components, the communications, the system and middleware necessary

- Desktops with different browsers such as the latest versions of Chrome, Firefox, and Edge etc. installed for testing.
- Devices such as mouse and keyboard.
- Operating system – Windows 11

5.1.2. Other software or supplies needed to support testing

- Apache Tomcat Version 10 for running the TravelApp based on the technology of Java and J2ee.
- MySQL Version 8.0.35.0 for hosting databases.
- Lightspark – the Flash replacement to implement Flash functionality
- Microsoft software – Microsoft Word for creating testing documentation such as test strategy, Microsoft Excel for creating testing documentation such as test case, Microsoft Project for managing testing (more details in section 6.1 Test Management Tools), Microsoft PowerPoint for presenting to the business.
- Eclipse IDE for building test scripts in Java Version 21.0.1(jdk-21).
- JIRA for managing testing and tracking issues (more details in section 6.1 Test Management Tools)

5.1.3. Security and access requirements to the test area and equipment

- Conducting testing in a secure environment with access control using keys for protecting the test area.
- Implementing password security measures for protecting test equipment and devices against unauthorised access.
- Implementing encryption for protecting test data and test documentation security.

5.1.4. Test tools and utilities required

- Selenium WebDriver, JUnit or TestNG, Maven for implementing automated tests (more details in section 6.2 Test Automation Tools).
- JMeter for implementing load test or stress test of performance testing stage.
- Jenkins for monitoring the test execution process, collecting test results and supporting the testing process for TravelApp.

5.1.5. Any other testing needs

If issues are discovered or new testing requirements arise, any other testing software or tools may be required during the testing process. For example, when testing certain functions, a data generator tool may be required to provide large amounts of data to support data-driven testing.

6. Testing Tools

6.1. Test Management Tools

In this project we are going to use test management tools for planning and managing the project activities throughout the software life cycle. These tools help in organising and scheduling the tasks, tracking the test results, and generating the reports which helps in analysing the testing progress from start to finish. Following tools we are going to use for this project.

- Microsoft Project (MS Project): The MS project is a project management tool, and, in this project, we are going to use this tool for developing project plans, creating tasks, scheduling, assigning resources and to generate Gantt charts and reports.
- JIRA – This tool is a bug tracking tool and, in this project, we are going to use this tool to organise the tasks, track the issues and plan and manage teamwork.
- Google Meet – Google meet allows the team to connect virtually with audio and video options. It will be utilised for this project, to connect with team members for team meetings, collaborate on documents by sharing desktop screens and present the documentations. Google meet will allow our team to facilitate better project communication.
- Microsoft Excel – Excel helps users to organise, calculate and format the data. In our project, we are going to use excel for managing the test cases for manual and automation testing. This tool will help us update and track the test cases efficiently
- Microsoft PowerPoint – It is a widely used tool for communicating ideas and information to audiences. Here, in this project we will use this tool for presenting project information to the team members and project sponsors in an organised manner.
- Microsoft Word- With the help of this tool we will document all our project work in an efficient way and we will create test strategy , test plan and test reports for team and business.

6.2. Test Automation Tools

We need to use some automation tools for our travelApp testing. Please see the below,

- Selenium
An open-source tool for automating web application testing across several platforms and browsers is called Selenium. It facilitates the development of automated test cases and supports a variety of computer languages. It makes it easier to run tests more quickly, providing feedback on the functionality of the application more quickly. To automate the testing of our TravelApp's login functionality, a Selenium script may be created that opens a browser, opens the login page, inputs valid credentials, clicks the login button, and checks to see if the user has successfully logged in.
- Junit or TestNG
JUnit is an open-source framework for unit testing Java programmes. This framework is used by Java developers to write and run automated tests. Every time a new piece of code is added to Java, some test cases need to be run again. This is done to make sure that there are no broken codes in the code.
In this project we can simply determine how many test cases are passed, failed, and skipped during our test period by using TestNG to provide an appropriate report. The unsuccessful test cases can be run independently. Java programmes are functionally, integrated, and unit tested using JUnit and TestNG.

- Maven
The Apache Software Foundation created the automation and management tool Maven. It is written in Java to construct C#, Ruby, and other language projects. With the use of plugins and the Project Object Model, developers may establish projects, dependencies, and documentation.
As part of the build process, Maven can be set up to run tests from JUnit or TestNG in our project. As an example, a Maven project may contain a "test" phase that, upon project build, automatically executes each unit test.
- JMeter
Website load testing can be done with JMeter and Selenium in combination. Load testing is carried out with Selenium, and its performance is measured with JMeter. The speed at which HTML pages load can be measured, as well as the functionality of CSS and JavaScript. Example: We will simulate numerous people visiting our travel App application at the same time using JMeter, and we will test the throughput and response time of the application under various load situations.
- Jenkins
Software projects can be delivered and integrated continuously (CI/CD) with the help of Jenkins, an open-source automation server. Jenkins testing automation can assist in lowering the time and effort needed to conduct tests by hand by automating the procedure. Our team uses Jenkins for automation testing since we want to make sure that our code is reliable and of high quality.
- Eclipse
We will use Eclipse to write our automation test script in this project. An environment for creating and executing automated tests for our travel App application is effectively and efficiently created by the combination of Selenium, Maven, TestNG, JUnit, Eclipse, and Java. When it comes to managing dependencies, creating test cases, developing test scripts, and running tests inside the Eclipse IDE, the tools work well together.

References

Figure 1:

Vijay. (2023, June 24). *Types of software testing: Different testing types with details*. Software Testing Help. <https://www.softwaretestinghelp.com/types-of-software-testing/>

Figure 2:

<https://d1jnx9ba8s6j9r.cloudfront.net/blog/wp-content/uploads/2019/02/types.png>