|  |
| --- |
|  |

|  |
| --- |
| Imam Abdulrahman Bin Faisal University |
| College of Computer Science & Information Technology  Department of Computer Science |

**CS 411 – Software Engineering**

**Term 1 – 2018/2019**

Software Test Plans

For

Railway.Manage();



Version 0.1

CS Year 4, G1

Ms.Wadha Almattar

*December 1,2018*

Software Test Plan (STP) was prepared and provided as a deliverable for Software Engineering, CS 411, Term 1, and it will be used by all developers and stakeholders.

This document is based in part on the IEEE Recommended Practice for STP Descriptions.

Team Members:

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | ID | Role |
| 1 | Muneera abdullah alhajri | 2160007230 | Leader |
| 2 | Reema Ibrahim Alyousef | 2160003133 | Member |
| 3 | Rahaf Saleh Alzahrani | 2160006662 | Member |
| 4 | Tasneem Hamdy dosoqi | 2160007430 | Member |

**Table of Content**

[Revision History 3](#_Toc531382648)

[Table of Tables 4](#_Toc531382649)

[Table of Figures 5](#_Toc531382650)

[1. Introduction 6](#_Toc531382651)

[2. Test items 8](#_Toc531382653)

[3. Features to be tested 8](#_Toc531382654)

[4. Approach 8](#_Toc531382655)

[5. Pass / fail criteria 8](#_Toc531382656)

[6. Testing process 8](#_Toc531382657)

[7. Environmental requirements 8](#_Toc531382658)

[8. Change management procedures 8](#_Toc531382659)

[9. Plan approvals 8](#_Toc531382660)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| All members | Dec 1, 2018 | Prepared initial version | 0.1 |

# Table of Tables

*Table 1 Definitions 6*

*Table 2 Acronyms* 6

*Table 3 software testing resources* 9

*Table 4 Testing Task's schedule* 9

*Table 5 Plan Approvals* 11

# Introduction

The Software Test Plan (STP) is intended to describe how the testing phase will be approached during the development of the Railway.Manage(); project. This document is designed to specify the scope, resources, and schedule of the test plan.   
  
This plan is required to identify the features to be tested, alongside specifying the types of tested to be performed and the members responsible for carrying out these tests.

# Objectives

This document is expected to specify the testing plan for the Railway.Manage(); project. This will be achieved by outlining the plan’s scope, resources and the allocated schedule for completing each task of the testing phase.

STP will also outline how the testing will approached by specifying the types of testing to be conducted, their deliverables, and allocation of the necessary resources to perform the tests.  
This plan must be accurate and complete in order for all the functions to be tested in a suitable and organized manner to assure that they work as intended by both the team and client alike.

* 1. **Testing Strategy**

Each function of this project must pass several testing stages to evaluate whether it satisfies the client’s requirements listed in the SRS document, and to compare it to the expected outputs outlined by the testing team.

For each task of the testing phase, a test plan is defined alongside the deliverables, constraints and assumptions of these tasks. It will also go through the appropriate test type in order to judge whether it fails or passes the performance criteria.

Testing will be performed at several stages in the life cycle of the project’s development. Since testing is a very dependent operation, planning and conducting tests will be a continuous process throughout the development phase of the Railway.Manage(); project. No new components or features are added unless they pass the criteria made by the testing team.

Note that any major issues discovered by the test plan must be clearly informed to the project manager, development team and the rest of the testing personnel.  
(A standard test approach is clearly defined in great detail in Section 5 of this document. Please refer to it if needed)

* 1. **Scope**

As previously stated, the testing phase requires continuous rework and updates for it to be successfully conducted. The testing team will provide upgrades to this document regularly to state whether any changes occurred during the testing stage of this project.  
  
**Any updates incorporated into this document must be informed earlier to the development team and the project manager alike.**

* 1. **Reference Material**

**Listed below are the references used in this document:**

[1] IEEE Standard for Software Project. [Online] 1998. IEEE Std 1058-1998.

* 1. **Definitions and Acronyms**

Any technical terminology mentioned in this document is specified and explained in Table 1.

Table 1 Definitions

|  |  |
| --- | --- |
| **Terminology** | **Definition** |
| **IEE Standard** | A universal standard followed by software developers. |
| **STP** | A test plan that coordinates any tests made in software development, to make sure that the project fulfils its preconditions. |
| **SRS** | A document that specifies all the requirements that needs to be completed before the completion of the project. |
| **Bugs** | Errors and issues that can be encountered during the run time. |
| **Interface** | The view provided to the user, considered the connection between the user and the system. |
| **Errors** | Deficiencies and faults during the run time of the system. |

Table 2 below lists all the acronyms used in this document.

Table 2 Acronyms

|  |  |
| --- | --- |
| **Acronym** | **Definition** |
| **STP** | Software Test Plan. |
| **SRS** | Software Requirement Specifications. |
| **IEE** | Institute of Electrical and Electronics Engineers. |
| **GUI** | Graphical User Interface. |

# Test items

As the requirements approved in Software Requirements Specification (SRS) the system will be tested as a whole to ensure it meets its requirement. And as what been approved on Software Design Specification (SDS) the system will be tested to ensure the design and the way of developed software system works matches the listed information on it. Also the interface will be tasted as will to ensure that go way with the GUI standers. The data with the database integrity will be tested also.

# Features to be tested

Railway.Manage(); application will adhere to the requirements specified in the SRS document, section 3.2; Functional Requirements. Also it will follow System Functionality identified in the SDS document, section 2.1.

# Approach

* 1. **Component Testing**
  2. **Integration Testing**
  3. **Job Stream Testing**

Job Stream Testing is to ensure that the Railway.Manage(); system operates in the production environment. It makes sure that consists of a sequence of jobs flow is running correctly together with times, priorities, and other dependencies that determine the order of processing the requirement.

* 1. **Interface Testing**

Interface testing is executed to guarantee that interactions whole system and components’ passage of data and control are working properly and errors are handled perfectly.

This section emphasizes on the connection between the application and the external component.

* 1. **Security Testing**

To guarantee that all authorization is in deeded the Security Testing will be generated , it ensure that the systems control and audit ability features of the application are functional. And to prevent unwanted access to unauthorized users to change or even view any critical information that is not allowed to. For example the passenger is not allowed to modify any information on train schedule only operators can.

**Technique:** Accessing the system with incorrect usernames and passwords should restrict the user from entering the system.

**Completion criteria**: Security testing verifies that only registered users can access their accounts.

* 1. **Recovery Testing**

The recovery testing is a non-functional testing that is necessary for the system. It is check how the software will recover after any crash or hardware failure. Recovery testing will force the system to fail in different ways then verify that the recovery is performed properly. In Railway.Manage(); the system is designed to recover automatically by itself by implementing check pointing strategy where a fault tolerance is added to the software and whenever there is failure, it will save a snapshot of the current software state then restart from that point. For data failure, there will be a backup for all the data in the database regularly.

* 1. **Performance Testing**

The performance test of the Railway.Manage(); will be conducted via measuring the response time, sturdiness under heavy workload, and consistency of the system’s various test runs.

This type of testing is especially critical to assure and validate whether the performance requirements are satisfied, therefore it is essential to conduct this test several time with different scenarios and inputs at each try to investigate how deficiencies can be generated from various events.

To consider a response time acceptable, it should not exceed 3 seconds during the retrieval of any necessary data from the database, initial load up cannot exceed 20 seconds and transitioning from one interface to another should happen within the span of 0.5 seconds. GUI should be coherent and consistent across several devices; the testing team will run the applications on various devices to check for any inconsistencies in the application’s look-and-feel.

**Technique**: Run the different functions of the system under various scenarios and test-cases. **Completion criteria:** Valid output of the functions, acceptable runs time and execution as determined by the testing team.

* 1. **Regression Testing**

When adding any new feature or modification to the system, the system must be refactored and tested several times in order to assure that the modifications are not affecting the previously added functions.

This test is critical to carry out, whether the modification seem insignificant such as a new line in the code, or adding a whole new component to the system.

**Technique**: After integrating any addition, sub-components alongside the whole system should be tested to assure that they work as expected.

**Completion criteria:** Successful integration, no new errors should be generated from the new addition.

* 1. **Beta Testing**

Beta testing is a pre-release version of Railway.Manage(); software and it is important to make sure that the software meets the required functional requirements and to detect any faults or defects that may found in the software. This test will have done by some selected customers.

This test is done in an environment that cannot be controlled by the developers. During the test, the customers will record every problem that may arise and then report these issues to the developers to modify it.

# Pass / fail criteria

Railway.Manage(); must pass this criteria in order for it to be considered successful:

Functions related to each of the primary users shall work as specified in the SRS document.

1. All the data recorded in the database must be accurate and up-to-date.
2. Queries and retrievals must be done successfully.
3. If any bugs were discovered, they must be eliminated, and the affected feature should pass all the testing cases again.
4. All invalid user inputs should be considered and dealt with appropriately to avoid run-time errors.
5. GUI should be consistent and coherent in multiple devices.
6. Inserting the booking and all related data should be recorded in the database successfully.
7. Retrieval of information should take more than 1.5 seconds.
8. Login function must be secure and impenetrable.
9. All functions should comply with the requirements set in the SRS document.

If any of these conditions are not met, they must be fixed immediately before commencing the implementation on any other function/

# Testing process

This chapter defines test deliverables, testing tasks, responsibilities, resources and schedule.

* 1. **Test Deliverables**

While working on the test stage, the software test plan document (STP document) will be produced. After finishing the testing stage, the final software project will be delivered and presented.

* 1. **Testing Tasks**

There are several testing tasks during the test stage. These tasks are listed below:

* Prepare the SRS and SDS documents to make sure the software is meeting the requirements and functioning as expected.
* Produce the STP while working during test stage.
* Prepare the hardware and software test environment.
* Perform all test activities to the software using different methods.
* Handle errors that may occur in the software during the test stage.
* Maintain the software when any change occurs.
  1. **Responsibilities**

During the test stage for the software, all the team members will be responsible for testing the software. Also, all the test tasks must be integrated, and each component specification followed. Moreover, the errors that may occur in the software during the test stage must be handled.

* 1. **Resources**

Table 3 below shows the resources that will be needed to complete the software testing phase.

*Table 3 software testing resources*

|  |  |
| --- | --- |
| Resource | Description |
| Hardware | Personal computers |
| software | * Net Beans 8.1 IDE * MySQL Workbench |
| human | Skilled team members, these skills are mentioned previously in the SPMP document, section 3.1.3, project staff training. |

* 1. **Schedule**

Table 4 Below shows the schedule of the testing tasks.

*Table 4 Testing Task's schedule*

|  |  |
| --- | --- |
| Task | Date |
| Software Requirement Specification (SRS) | October 29, 2018 |
| Software Design Specification (SDS) | November 9, 2018 |
| Develop test cases | November 20, 2018 |
| Perform all test activities using different methods. | November 26, 2018 |
| Handling errors occurred during test stage | November 29, 2018 |
| Modify the software | November 30, 2018 |
| Develop STP document | December 1, 2018 |

# Environmental requirements

This chapter define the environmental requirements: hardware, software, security, server, publications*,* and incorporated risks and assumptions.

* 1. **Hardware**

The necessary hardware needed to apply the testing tasks is the desktop computers.

* 1. **Software**

The required software needed to execute the testing tasks are:

* Net Beans 8.1 IDE
* MySQL Workbench
  1. **Security**

To produce high secure software, the system is developed using Java programming language which provides a number features that designed to improve the security of the applications. Also, the software provides a log in strategy to the system and database so doing changes in system or database is permitted only for the registered operators.

* 1. **Server**

MySQL Server will be required to employ the testing tasks.

* 1. **Publications**

There are two required documents along with STP document that produced during the testing stage. These documents are: Software Requirement Specification (SRS) document and Software Design Specification (SDS) document to make sure the software is meeting the requirements and functioning as expected.

* 1. **Risks and Assumptions**

To produce a high-quality software, the system must function as expected. Railway.Manage(); will go through the Software Development Life Cycle (SDLC) stages. The aim for SDLC is to produce a high-quality software that exceed the expectations. The rapid development of the software may cause changes in the code which leads to changes in the plan. Below is listed, the potential risks and their plans.

* + 1. **Test Item Availability**

During testing unit, an inaccessibility of an item may interrupt the testing of that unit which in turns delay the whole testing process. The contingency plan for this issue is to test other items that are ready and waiting to be tested in that unit until the incomplete item is ready and available.

* + 1. **Test Resources Availability**

During testing units, if the resources that need to be tested is unavailable, it may defer the test plan for the completed units. For example, if there was a heavy load or any technical issues on the server, it may shutdown which harm the system. The contingency plan for this issue is to have a backup server so if there is any issue in the original server, the system will shift to this backup server.

* + 1. **Time Constraints**

Any changes in the software like adding new or deleting features to the software, or any unexpected delay will cause the delivered time to be delayed to. The contingency plan for this issue is to make sure all the functions in the software are working properly as expected. Also, increase the estimated working hours to make sure estimated delivery time is met.

# Change management procedures

As stated previously in section 1, updates are expected to occur during the testing phase of this project. All of these changes must be clearly documented after acquiring the approval of the project manager and the client.

After proposing a necessary update, a meeting will be held in order to discuss and review the consequences of the proposed change.  
Note that no changes will be done unless they are reviewed and approved by the individuals stated in section 10.

# Plan approvals

Table 5 below outlines the individuals responsible for approving any modifications to the plan.

*Table 5 Plan Approvals*

|  |  |  |
| --- | --- | --- |
| Name | Signature | Date of Approval |
| Project Supervisor: Wadha Almattar |  |  |
| Project Manager: Muneera Alhajri |  |  |
| Testing Team Leader: Tasneem Dosoqi |  |  |
| Development Team Leader: Reema Alyousef |  |  |
| Research Unit Leader: Rahaf Alzahrani |  |  |