**Test Plan**

Team P5

**Prepared by:**

Joshua Awozele

Robin Houben

10/12/2020

**TABLE OF CONTENTS**

1.0 INTRODUCTION

2.0 OBJECTIVES AND TASKS

2.1 Objectives

2.2 Tasks

3.0 SCOPE

4.0 Alpha Testing (Unit Testing)

5.0 Hardware Requirements

6.0 Environment Requirements

6.1 Main Frame

6.2 Test Schedule

7.0 Control Procedures

8.0 Features to Be Tested

9.0 Resources/Roles & Responsibilities

10.0 Schedules

11.0 Dependencies

12.0 Risks/Assumptions

13.0 Tools

14.0 Approvals

**1.0 INTRODUCTION**

The P5 team are responsible for the implementation of testing activities. Primarily, the most integral aspect of the testing will include ensuring all classes, methods, DAOs and repositories are tested to ensure a working system has been developed. Firstly, JUnit will be the key testing framework which would test core components of the overall implementation.

**2.0 OBJECTIVES AND TASKS**

**2.1 Objectives**

Some mock tests were implemented in a way that would allow a further development of testing the most important features in the system. Discord is a communication tool used within the team to plan smarter ways of working. On a weekly basis, tasks are allocated to members of the team to complete remotely and commit onto GitHub for transparency. The weekly lecture is facilitated in project-stand up method where participants are allowed to showcase each team’s progress by presenting project artefacts/deliverables.

The testing activities are divided between 2 work resources (Joshua Awozele and Robyn Houben). A test plan which would detail the activities required is important to ensure there is a baseline document. The test reports will be collated in a form of saving the test results in a video or image format with a brief description detailing the tests have passed.

**2.2** **Tasks**

The following tasks are required to begin with the basis of testing:

* PartyDAOImplSpringTest
* PartyRepositoryTest
* PartyDAOTest
* CarDAOImplSpringDeleteTest
* CarDAOImplSpringTest
* CarRepositoryTest
* ChargingRecordDAOImplSpringDeleteTest
* ChargingRecordDAOImplSpringTest
* InvoiceDAOImplSpringDeleteTest
* InvoiceDAOImplSpringTest
* RoleDAOTest
* RoleDAOImplSpringDeleteTest
* RoleDAOImplSpringTest
* RoleRepositoryTest
* UserDAOImplSpringDeleteTest
* UserDAOImplSpringTest
* UserDAOTest
* UserRepositoryTest

**3.0 SCOPE**

**General**

The aim of the testing is to ascertain that all core components of the application will function as expected/.

**Tactics**

The initial stage of carrying out testing is done by a Git pull, this would provide the basis of testing activities needed, as opposed to a general rule to test all even though they are still under implementation. The testing is done in parallel with has been implemented till date.

**4.0 Unit Testing**

**Definition:**

The core aim is to ensure components of the software system are tested. Methods, functions, classes etc are to be tested to ensure a solution has been found to mitigate against any components with bugs.

**Participants:**

Joshua Awozele

Robin Houben

**Methodology:**

Describe how unit testing will be conducted. Who will write the test scripts for the unit testing, what would be the sequence of events of Unit Testing and how will the testing activity take place?

**5.0 HARDWARE REQUIREMENTS**

Computers

Modems

**6.0 ENVIRONMENT REQUIREMENTS**

**6.1** **Main Frame**

The testing activities are designed remotely using personal computer devices which include a laptop with a capability to install standard testing tools such as IntelliJ, Maven and Tomcat. Ideally, individual electronic gadgets are passworded to protect files from unauthorised access from non-participants. GitHub has an outline of all the project-related files which locally accessible by team members.

**6.2 Test Schedule**

The test schedule has an overview of the timeline in which tests were implemented.

The schedule was created via an open-source platform (Team Gantt). The schedule entail the testing of implemented spring data and core components such as methods, classes, repositories and DAOs were also Junit tested. Primarily, Team P5 are responsible for the onboarding of new users, in turn the CRUD aspects of the implementation were tested to ensure they meet expectations.

**7.0 CONTROL PROCEDURES**

**Problem Reporting**

When incidents are encountered, a snapshot of the error must be recorded to allow the whole team to become aware of the current situation. The project is also hosted on GitHub with a separate area to gather and track all issues which may arise during the course of the project.

**Change Requests**

Activities relating to change requests will be agreed upon by the team before any implementation is carried out. The idea of making a change request can take place during weekly meetings with team members allocating who would be responsible for completing the task. Although, there would be no formal documentation on change requests, team members must ensure any changes are effectively communicated across the team to ensure there would be no decline in the overall team morale.

**8.0 FEATURES TO BE TESTED**

The following features are required to be tested:

* PartyDAOImplSpringTest
* testFindAll
* testFindByName
* testFindByUuid
* testFindByPartyRole
* testFindById
* testDelete
* PartyRepositoryTest
* testFindByName
* testFindByUuid
* testFindByPartyRole
* testFindAll
* PartyDAOTest
* deleteAllTest
* findByPartyRoleTest
* findByNameTest
* deleteByIdTest
* createPartyDaoTest
* findByIdTest
* deleteTest
* CarDAOImplSpringDeleteTest
* testDeleteAll
* testDeleteByID
* testDelete
* CarDAOImplSpringTest
* testFindByNumberPlate
* testSave
* testFindById
* CarRepositoryTest
* testFindByNumberPlate
* ChargingRecordDAOImplSpringDeleteTest
* testFindAll
* testSave
* testFindById
* InvoiceDAOImplSpringDeleteTest
* testDeleteAll
* testDeleteById
* testDelete
* InvoiceDAOImplSpringTest
* testFindAll
* testSave
* testFindById
* RoleDAOTest
* testLoadcontext
* RoleDAOImplSpringDeleteTest
* testDeleteAll
* testDeleteById
* testDelete
* RoleDAOImplSpringTest
* testFindAll
* testFindByRoleName
* RoleRepositoryTest
* findByName
* UserDAOImplSpringDeleteTest
* testDeleteAll
* testDeleteById
* testDelete
* UserDAOImplSpringTest
* testFindAll
* testSave
* testFindById
* testFindByUserName
* UserDAOTest
* testLoadcontext
* UserRepositoryTest
* findByNames
* findByUsername
* UserRepositoryTest
* test1.

**9.0 RESOURCES/ROLES & RESPONSIBILITIES**

The agreement is to have two members of the team carry out testing. This would include the documenting the process and the actual technical testing of the software system. Both participants can have an overlapping role which may see the creation of a test plan as well as implementing the test plan to cover specific activities.

Generally, the team have a planned weekly meeting which is scheduled to hold every Wednesday at 11 AM.

**10.0 SCHEDULES**

**Major Deliverables**

* Test Plan
* Test Requirements
* Test Results

**11.0 DEPENDENCIES**

Initially, the system had been tested without a need for testing services. The task regarding services has now been tested, although this should have been implemented first before proceeding on to implement tests for DAOs, repositories etc.

**12.0 RISKS/ASSUMPTIONS**

The most important aspect of the Junit testing is to ensure the system works as required. Additionally, the system may not perform up to the required standard. To mitigate against any risks, the testing activities are widely communicated about on a weekly basis and agreements are made to understand how and when the results will be made available.

**13.0 TOOLS**

The fundamental plan of implementing testing will include the use of the following tools.

* IntelliJ IDE – this is required to build the application from inception.
* Java – a high level programming language that is fit for developing dynamic and robust software systems.
* Maven – used for building the project, dependencies and allowing a seamless build process.
* GitHub – all commits and push are specifically done on GitHub.
* Discord – for weekly team meetings to discuss progress and next steps.
* Mattermost – a communication tool used by the project sponsor to distribute important information to the project teams/participants.

**14.0 APPROVALS**

Joshua Awozele

Rui Pinto

Joao Leite

Robin Houben

Aeryk Del Mundo

Afonso Cabecadas.