SEM5640 Group Project

Test Plan

|  |  |
| --- | --- |
| Author: | Dominic Parr |
| Config Ref: | SEM5640.2019.tp |
| Date: | 29th October 2019 |
| Version: | 0.1 |
| Status: | Draft |

Department of Computer Science

Aberystwyth University

Aberystwyth

Ceredigion

SY23 3DB

Copyright © Aberystwyth University 2015

CONTENTS

CONTENTS 2

1. Introduction 3

1.1 Purpose of this Document 3

1.2 Scope 3

1.3 Objectives 3

2. TESTING 3

REFERENCES 3

DOCUMENT HISTORY 4

# Introduction

## Purpose of this Document

The purpose of this document is to describe the groups’ plan for testing the functionality elements of the system and to ensure a certain standard is maintained across all systems.

## Scope

This test plan will apply to all systems within the application, some sections may be more relevant than others.

## Objectives

The objectives of this document are:

* To identify the types of testing that will be explored.
* To state a conformity that all tests will follow.

# Testing

There are two categories of testing that will be applied; this is manual testing – the practice of a user or users exploring a system to identify bugs by chance via exploratory testing or conforming to a set test table to ensure functionality of the system is as intended. In most cases, both are applied. There is also automated testing – the practice of creating tests using tools such as MSTest (Asp.Net), JUnit (Java EE), Moq.   
In this project there will be a higher focus to automated testing, this is primarily because the group size of the project is too small to pour large quantities of time into manual testing of a large system on a regular occasion. Manual testing will be performed at certain milestones within the project. The four types of testing that will be implemented are:

* Unit testing – These tests will enforce consistent functionality across all levels within the project, utilising Moq testing for system behaviour, and other tests to enforce functionality.
* Integration Testing – These tests will involve a mix of manual, and automated testing. The manual testing will consist of a graphical check to ensure that the design remains consistent. The unit tests that had been compiled from all components will be refactored (if necessary) and ran to confirm proper functionality.
* Functional Testing – This will primarily be covered by unit tests that will be written as the code is developed, it will test a range of things with a high focus on system dependencies, such as the databases. At this stage we would also confirm that the test table could be satisfied fully, and any discrepancies would be reported.
* Exploratory Testing – This is a testing exercise in which testers are assigned a loosely defined task to achieve using the software being tested [1]. It will only be performed on the system as a whole a few times throughout the project due to lack of resources; after integration of main components is successful a ‘brute force’ test will be performed on the system to try to identify obscure bugs.

REFERENCES

|  |  |
| --- | --- |
| [1] | C. Maynard, “Software Testing for Continuous Delivery,” [Online]. Available: https://www.atlassian.com/continuous-delivery/software-testing. [Accessed 31 October 2019]. |

DOCUMENT HISTORY

| *Version* | *CCF No.* | *Date* | *Changes made to document* | *Changed by* |
| --- | --- | --- | --- | --- |
| 0.1 | N/A | 29/10/2019 | N/A – Initial Creation | DOP2 |
| 0.1 | NA | 31/10/2019 | Added Section 2: TESTING | DOP2 |
|  |  |  |  |  |
|  |  |  |  |  |