EGERTON UNIVERSITY



**FACULTY OF SCIENCE**

**COMPUTER SCIENCE DEPARTMENT**

**TEST PLAN FOR ONLINE SUSPICIOUS DISCUSSION DETECTOR**

**PRESENTED BY: BRENTONE ALISTAR**

**S13/15358/16**

**SUPERVISOR: DR. PHOEBE**

**COORDINATOR: DR. GIKARU**

Table of Contents

[1. Test plan identifier 3](#_Toc59092005)

[1.1. Introduction 3](#_Toc59092006)

[1.2. Objectives 3](#_Toc59092007)

[1.3. Scope 3](#_Toc59092008)

[2. Features to be tested 4](#_Toc59092009)

[2.1. Client 4](#_Toc59092010)

[2.2. Supplier 4](#_Toc59092011)

[2.3. Admin 4](#_Toc59092012)

[3. Features not to be tested 4](#_Toc59092013)

[4. Process overview 5](#_Toc59092014)

[5. Testing process 5](#_Toc59092015)

[6. Test strategy 6](#_Toc59092016)

[6.1. Unit testing 6](#_Toc59092017)

[6.2. Integration testing 6](#_Toc59092018)

[6.3. System testing 6](#_Toc59092019)

[6.4. Performance testing 6](#_Toc59092020)

[6.5. User interface testing 7](#_Toc59092021)

[7. Item pass/fail criteria 7](#_Toc59092022)

[7.1. Suspension criteria 7](#_Toc59092023)

[8. Test deliverables 7](#_Toc59092024)

[9. Environmental needs 8](#_Toc59092025)

[9.1. Client-side environments 8](#_Toc59092026)

[9.2. Server-side environments 8](#_Toc59092027)

[10. Staffing and training needs 8](#_Toc59092028)

[11. Schedule 9](#_Toc59092029)

[11.1. Approval 9](#_Toc59092030)

# Test plan identifier

This is a test plan for OSDD application. The system targets to analyse online contents including major social media chat rooms and notorious websites. It provides a mechanism to help monitor social media contents through auto detection, identification, classification and an auto feedback to major social security agencies including cybersecurity agencies to acknowledge such incidences.

The system presents a registration and authentication paradigm to new user’s assigning the accounts. Users are categorized into levels improving system’s performance as well as a security consideration.

# Introduction

The goal of this document is to develop a test plan for the OSDD System. This document defines all the procedures and activities required to prepare for testing of the functionalities of the system. The objectives of the test plan are to define the activities to perform testing, define the test deliverables and to identify the various risks and contingencies involved in testing.

## Objectives

* Identify existing project information.
* Identify the approach that should be followed.
* Identify the features that should be tested.
* List the recommended test requirements.
* Recommend and describe the testing strategies to be employed.
* Identify the required resources and provide an estimation of the test efforts.
* Fix the schedule of intended testing activities.
* Identify the risks associated with the test strategy.
* List the deliverable elements of the test activities.

## Scope

The features of the system to be tested are:

* User interface responsiveness.
* Input validation.
* Database.
* Linked pages.

The scope of test will include the User interface, the database, login and sign up modules and all the linked pages in the application.

# Features to be tested

The following features will be tested:

## Driver

* Registration and authentication.
* Functionality to view available parking areas.
* Functionality to book and reserve a parking lot.
* Functionality to manage account and update account details.
* Functionality to pay for the booked parking lot.

## Worker

* Registration and authentication.
* Functionality to report lost cars.
* Functionality to hold lost cars.

## Admin

* Registration and authentication.
* Functionality to view and generate reports.
* Functionality to manage users.
* Functionality to respond to queries.
* Functionality to manage lost cars

# Features not to be tested

All the above-mentioned features will be tested. Testing is dependent on the time available, in the event that time will not allow testing of all features, then the features with the highest priorities will be considered first. Other features that will not be tested include the following.

|  |  |
| --- | --- |
| **FEATURE** | **DESCRIPTION** |
| Network security | This is beyond scope. |

# Process overview

The following steps represent the overall flow of the testing process:

1. Identify the requirements to be tested. All test cases shall be derived using the current Program Specification.
2. Identify which particular test(s) will be used to test each module.
3. Review the test data and test cases to ensure that the unit has been thoroughly verified and that the test data and test cases are adequate to verify proper operation of the unit.
4. Identify the expected results for each test.
5. Document the test case configuration, test data, and expected results.
6. Perform the test(s).
7. Document the test data, test cases, and test configuration used during the testing process. This information shall be submitted via the Unit/System Test Report (STR).
8. Successful unit testing is required before the unit is eligible for component integration/system testing.
9. Unsuccessful testing requires a Bug Report Form to be generated. This document shall describe the test case, the problem encountered, its possible cause, and the sequence of events that led to the problem. It shall be used as a basis for later technical analysis.
10. Test documents and reports shall be submitted. Any specifications to be reviewed, revised, or updated shall be handled immediately.

# Testing process

* **Organize Project**: It involves creating a System Test Plan, Schedule & Test Approach, and assigning responsibilities.
* **Design System Test**: It involves identifying Test Cycles, Test Cases, Entrance & Exit Criteria, Expected Results. In general, test conditions, expected results will be identified by the Test Team in conjunction with the Development Team. The Test Team will then identify Test Cases and the Data required. The Test conditions are derived from the Program Specifications Document.
* **Design Test Procedure**: It includes setting up procedures such as Error Management systems and Status reporting.
* **Build Test Environment**: It includes requesting, building hardware, software and data set-ups.
* **Execute System Tests:** The tests identified in the Design Test Procedures will be executed. All results will be documented and Bug Report Forms filled out and given to the Development Team as necessary.
* **Signoff**: Signoff happens when all pre-defined exit criteria have been achieved.

# Test strategy

The testing will be done using both manual and automatic test tools. The testing will cover the requirements for all of the different roles participating in the site: sellers, buyers and administrators. The following outlines the types of testing that will be done for unit, integration, and system testing.

## Unit testing

Unit testing is a method of testing that verifies the individual units of source code are working properly. The goal of unit testing is to isolate each part of the program and show that the individual parts are correct.

Unit Testing is done at the source or code/implementation level for language-specific programming errors such as bad syntax, logic errors, or to test particular functions or code modules. The unit test cases shall be designed to test the validity of the program’s correctness. This test will be performed by the developers.

## Integration testing

Integration tests exercise an entire subsystem and ensure that a set of components play nicely together.

## System testing

The goals of system testing are to detect faults that can only be exposed by testing the entire integrated system or some major part of it. Generally, system testing is mainly concerned with areas such as performance, security, validation, load, and configuration sensitivity. But in this case focus is only on Performance and Load testing.

## Performance testing

This test will be conducted to evaluate the fulfillment of a system with specified performance requirements. This will be performed by:

* Deleting data and check if it follows the right sorting algorithm to sort the resulting data or output.
* Trying to store new data and check if it over writes the existing once.
* Trying to load the data while they are already loaded.

## User interface testing

User Interface (UI) testing verifies a user’s interaction with the software. The goal of UI testing is to ensure that the UI provides the user with the appropriate access and navigation through the functions of the target-of-test. In addition, UI testing ensures that the objects within the UI function as expected and conform to corporate or industry standards.

For the website, unit testing is done first. Because unit testing will ensure that the all components of the system are working properly or not. If a single unit does not work properly the integration test will not necessary to perform.

# Item pass/fail criteria

A test scenario will be considered a fail case if any of the following happens:

* User registration fails.
* User authentication fails.
* Publishing of tenders fails.
* Response to applications fails.
* Making of bid/prequalification applications fails.
* Generation of reports by the system admin.
* The system crashes.

## Suspension criteria

The test will be suspended if any of these events occur:

* A user cannot register.
* A user cannot login.
* The system crashes.

# Test deliverables

The following documents will be generated as a result of these testing activities:

* Master test plan (this document)
* Individual test plans for each phase of the testing cycle
* Test Design Specifications
* Test log for each phase
* Acceptance Test plan.
* Unit test plan.
* Screen Prototypes.
* Test report.
* Test scenario and expected result in an excel sheet.
* System manual.

# Environmental needs

There are essentially two parts to the SME online tender management system application in production: the client-side, which will facilitate access to the system by clients and suppliers via the Internet. The server-side which will deal with processing of information, storage of information in a database and retrieval of information from the database.

## Client-side environments

The client-side is any web browser that runs on client device. Client devices are workstations, laptops and mobile devices with internet access enabled.

Supported browsers are:

* Opera browsers
* Chrome browsers
* Mozilla browsers
* Internet Explorer
* Safari browsers

## Server-side environments

The system will be hosted on an apache or ngingx server that will provide the following functionalities:

* A MySQL database for data storage.
* Storage of system resources.

# Staffing and training needs

The following will be the required staff for testing the system:

* **Project Manager**: Responsible for managing the total implementation of the Web application.
* **Test manager**: Responsible for developing the master test plan, reviewing the test deliverables, managing the test cycles, collecting metrics and reporting status to the Project Manager, and recommending when testing is complete.
* **Test engineer:** Responsible for designing the tests, creating the test procedures, creating the test data, executing tests, preparing incident reports, analyzing incidents, writing automated test procedures, and reporting metrics to the test manager.

Each and every member assigned to this project should be experienced with:

* General development & testing techniques.
* System’s development lifecycle methodology.
* All development and automated testing tools that they may be required to use.

# Schedule

The schedule for testing the system will be as follows:

Testing will be done throughout the process of its development depending on the type testing in question.

## Approval

The test will be approved once it meets all quality requirements.