EGERTON UNIVERSITY

FACULTY OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

TEST PLAN FOR IMMUNOTRAK

PREPARED BY: ERICK OGARO

REG. NUMBER: S13/09690/15

PROJECT SUPERVISOR: MR. G. NDIRANGU

PROJECT COORDINATOR: DR. W. GIKARU

DATE: MAY 2019

Table of Contents

INTRODUCTION....3

1.0 OBJECTIVES....3

1.1 SCOPE....4

1.2 CONSTRAINTS....5

2.0 TESTING PLAN....5

2.1 FUNCTIONALITY....6

2.2 USABILITY....7

2.3 COMPATIBILITY....8

2.4 PERFORMANCE....8

2.5 SECURITY....9

2.6 USER INTERFACE....10

3.0 ENVIRONMENT FOR TESTING....11

4.0 PROCEDURE....12

4.1 CASES....12

4.1.1 FUNCTIONALITY....12

4.1.2 USABILITY....13

4.1.3 USER INTERFACE....14

4.1.4 CROSS-PLATFORM COMPATIBILITY....15

4.1.5 APPLICATION PERFORMANCE....15

4.1.6 PLATFORM SECURITY....16

INTRODUCTION

This Test Plan Document details the whole process and the schedule of intended test activities that are carried out during the testing of the ImmunoTrak Platform. It gives an overview of the test specifications for ImmunoTrak. This document describes the way the testing of the system will be carried out, the test types and conditions which will be tested. This test plan will enable the testers to have definite procedure when testing a system. Testing is done to ensure that the system meets the stipulated objectives.

1.0 OBJECTIVES

To achieve a perfectly working system based on the its proposal document, software requirements specification and the software design document, the system has to go through a series of tests before its final release is deployed. The main objectives include the following:

a) To check the system for any errors
b) Provide a guide for the validation of the system
c) To check whether the system is compliance with the requirements specifications.
d) To check whether the data input is valid
e) To have a test specification to counter any difficulties that may impact the development and the future performance of the software.
f) The team's goal is to assist the project team in developing a strategy to deal with any errors. For this, the team will take a look at the most common errors to some very uncommon errors.
g) All (or at least most) of the test case will be listed, the development team will follow it step by step, item by item, to test all the necessary objects, data flows, limits, boundaries, and constraints of the software.

1.1 SCOPE

This section describes the scope for testing that is the extent to which the system's functionality is tested. There are very many units of testing but this document specifically focuses on the following units of testing:

a.) Data Integrity.

This entails making sure that the data being committed to the database is sanitized, ensuring that the data is of the right data type and that no other type of data can be stored in its place.

Data Inputs

The system utilizes the graphical interfaces that are user friendly, appealing and easy to use. Should there be any difficulty in operating the system, an online help has been implemented to assist the users of the system where necessary. The graphical user interface also makes the use of navigation panes and links, input text boxes and select panes.

Data outputs

In order to output some information from the database, the system utilizes table views which have implemented a search functionality for easy identification of the information the user needs. The displayed data is also limited with respect to the role the user of the system has.

Error Management

The possibilities of error occurrences are vast when it comes to web applications such as input errors, output errors, no exception handling among others. This document focuses on means of avoiding errors by providing guidelines on its implementation. All possible areas where errors may occur are tested and measures will be put into the system or improved based on the result.

Data Security

Not all users need to be registered in order to use the system. A user (registered) is the only one with the permission to delete or update his/her details. i.e . no user with a different role will be able to access functionality that are beyond their assigned permissions.

1.2 CONSTRAINTS

This section contains information about the business technical or resource related constraints that keep one from performing all tests necessary. However, each of these constraints represents a significant product quality risk and risk mitigation strategies should be considered.

1) Time constraint.
2) Funding
3) Security

   The team does not know any hackers that can help us test the security flaws of the system. So we have to rely on our own knowledge and have to trust the security measures that we have put in place for the system.
4) Simultaneous stress testing constraints

   The team does not have large enough group to have many people use the applications at the same time to perform real stress related testing.

2.0 TESTING PLAN

This section describes the format used during the testing of the system. The main aim is to ensure that the system is free of bugs and has no defects in it. The system will also be verified and validated to meet all the requirements outlined in the requirement definition document. To see this through, some testing procedures and strategies that will be used have been described in the following section and they include the following:

i.) Functionality Testing

ii.) Usability testing

iii.) Compatibility testing

iv.) Performance testing

v.) Security testing

vi.) Interface testing

2.1 FUNCTIONALITY

Functionality testing is the process designed to verify that a piece of code or software components operate correctly. This will entail testing for all the links in web pages, database connection, forms used for submitting or getting information from the user in the web pages etc. ImmunoTrak uses manual testing this is whereby ImmunoTrak is tested by simulating a user's process. This means that most of the testing is at black box testing that is interface level testing to ensure that the testers confirm the interfaces function as intended. White box testing is testing to check for bugs and errors. This is done during the development of ImmunoTrak since when a code has an error the system refuses to run on the web browser and the errors are documented in the log file. The following major categories of testing are implemented on ImmunoTrak.

1.) **Unit testing**

This is testing done by the developer who writes the code. The developer checks if particular units of code function properly. For ImmunoTrak method coverage is used to ensure that a particular method functions as expected.

2.) **Sanity testing**

This is testing done to ensure that major and critical functionality of ImmunoTrak function correctly. For example, a seller posts the items he/she wishes to be displayed and the said items are then uploaded and able to be viewed from both the home and posts page.

3.) **Smoke testing**

This is also known a build testing to ensure that each build of ImmunoTrak is stable. For example the beta version of ImmunoTrak which did not entail communication functionality functions properly then the alpha version of ImmunoTrak which entails the functionality not present in the beta version function.

4.) **Regression Testing**

This is testing to ensure that an additional line of code, fixing of errors or any enhancements in ImmunoTrak do not interfere with the existing functionality of ImmunoTrak. This testing ensures that the system is stable.

5.) **Integration testing**

This is testing to ensure that different modules of a system which can work independently can work coherently together. In this case, ImmunoTrak has two major modules: the seller and the buyer. Integration ensures that this two modules work together on the web without any interference.

6.) Usability testing

This is testing done by the consumer of the system to ensure that the system suits the consumer needs. ImmunoTrak testing is dependent on both the seller and the buyer testing the application.

2.2 USABILITY

Usability testing is the process by which the human-computer interaction characteristics of a system are measured, and weaknesses are identified for correction. Some of the areas that will be considered include:

- Ease of learning
- Navigation
- Subjective user satisfaction
- General appearance
- The website should be easy to use.
- Instructions provided should be very clear.
- Check if the instructions provided are perfect to satisfy its purpose.
- The main menu should be provided on each page.
- It should be consistent enough.

Test for navigation:

Navigation means how a user surfs the web pages using the available navigation bars and side panels, different controls like buttons, boxes or how the user uses the links on the pages to surf different pages.

Content checking:

Content should be logical and easy to understand. There will be checking for spelling errors and the appropriate usage of colors. Content should be meaningful. All the anchor text links should be working properly. Images should be placed properly with proper sizes.

User help:

The search functionality deployed in the data tables should be functioning appropriately to ensure that the users of the system can locate the content that they are looking for. An online help should also be available and should be as simple as possible so that users can familiarize with the operations that they wish to perform.

2.3 COMPATIBILITY

Browser compatibility:

Some applications are very dependent on browsers. Different browsers have different configurations and settings that your web page should be compatible with. Testing web application on different browsers like Internet Explorer, Firefox, Safari, Microsoft Edge and Opera browsers with different versions will be done.

Operating system compatibility:

Some functionality in your web application is that it may not be compatible with all operating systems. All new technologies used in web development like graphic designs, interface calls like different API's may not be available in all Operating Systems. Hence testing the web application on different operating systems like Windows, Unix, MAC, Linux, Solaris with different OS flavors will be done.

Mobile browsing:

There will be a testing of the web pages on mobile browsers to see how responsive the web application will have turned out. Compatibility issues however may arise on mobile browsers as well.

2.4 PERFORMANCE

Web performance testing should include web load testing and web stress testing.

Web load testing:

The system will be tested when many users are accessing or requesting the same page. Can system sustain in peak load times? The expected result is that the site should handle many simultaneous user requests, large input data from users, simultaneous connection to the database, heavy load on specific pages etc.

Web stress testing:

Generally, stress means stretching the system beyond its specified limits. Web stress testing will be performed to possibly break the site by giving stress and checked as for how the system reacts to stress and how it recovers from crashes. Stress is generally given on input fields, login and sign up areas.

2.5 SECURITY

The following are some of the test cases for the system's security that will be carried out:

- Test by pasting internal URL directly into the browser address bar without login. Internal pages should not open.
- If you are logged in using username and password and browsing internal pages then try changing URL options directly. I.e. If you are logged in as a user department, a supplier or a committee member, try directly changing the URL pattern which is not related to the logged in user, say that of the procurement manager. Access should be denied for this user to view others stats.
- Try some invalid inputs in input fields like login username, password, input text boxes etc. Check the system's reaction to all invalid inputs.
- Web directories or files should not be accessible directly unless they are given download option.

2.6 USER INTERFACE

Interface testing is done by verifying that communication is done properly. Compatibility of the server with software, hardware, network, and the database is to be tested. The main interface includes:

- Web server and application server interface
- Application server and Database server interface

Checking if all the interactions between these servers are executed and errors are handled properly will be done. If database or web server returns an error message for any query by application server then application server should catch and display these error messages appropriately to the users.

3.0 ENVIRONMENT FOR TESTING

Hardware:

- Server to setup web application.

Software:

Operating systems:

- Windows
- Linux

Browsers:

- Microsoft Internet Explorer
- Google Chrome
- Mozilla Firefox

Technology:

- MySQL Database
- Python3
- HTML
- CSS
- Visual Studio Code

Frameworks:

- Django (Python Framework)
- Chart.js (JavaScript graphical visualization framework)
- Jquery
- Materialize (CSS Framework)

4.0 PROCEDURE

4.1 CASES

4.1.1 FUNCTIONALITY

Test No.	Description	Expected Result
1	Testing all internal links	All available clickable links should be functioning and have a predefined function that they commit to.
2	Testing link jumping on the same page	Users should be able to jump from one link to another, either by typing in the URL or using the available navigation links.
3	Testing password reset function	Users who may have forgotten their password have the ability to reset their password. This is made possible by the system sending an email to the registered user with a password reset link.
4	Test for any orphan pages	There should be no page that cannot be accessed by clicking on the navigation links. Likewise, there should be no available pages that a user can access without the page being accessible from the main navigation tree.
5	Test all validations on each field	Each input field should have some form of validation. First and foremost, no field should be able to commit to the database a null or empty input
6	Test default and values that can	Default and values that can be selected

	be selected	should be able to reflect the same once they have been committed to the database
7	Test wrong input to the fields in the form	With working validation, no invalid and inconsistent data should be able to submit to the database unless all data is correct
8	Test all available forms	The available forms include registration form, login form, submit bid form, add requisition form, add tender form and evaluation form. All these forms should be able to submit the right data to their respective tables in the database.
9	Test database integrity	The database should be able to store the right data with the predefined data types and conforming to the available constraints. This will ensure no inconsistent data is stored in the database.

4.1.2 USABILITY

Test No.	**Description**	**Expected Result**
1	Test ease of learning	The system should be easy to use and easy to learn how to operate its functionality
2	Test navigation	The navigation should be easily identifiable and understandable. The navigation links should perform the function exactly as the user expects. The links should be easy to access and provide all possible functionality of the system.
3	General appearance	The web application should be appealing and the graphical user appearance should reflect the

		appearance of the organization.
4	Content checking	Content should be logical and easy to understand. Check for grammar errors, capitalization and such.
5	Test online help	There should be an available online help that constitutes all the functionality of the system. Should the user experience any difficulties in operating the system, they can access the online help and view the instructions based on their needs.

4.1.3 USER INTERFACE

Test No.	Description	Expected Result
1	Web server	The web server is functioning appropriately . The web server should be able to run and stop from the command interface.
2	Web server and application server interaction	The web server and application server should be able to communicate smoothly. In the case of any errors, they should be handled properly.
3	Application server and database server interaction	The web server should be able to connect to the database. Any failures to connect should be captured by the application server and the error message should be displayed to the user. This should also be the case if some query executed by a user fails.

4.1.4 CROSS-PLATFORM COMPATIBILITY

Test No.	Description	Expected Result
1	Browsers	The system will be tested on Microsoft Edge, Mozilla Firefox, Google Chrome, and other web browsers. It is expected that the system should run on most browsers.
2	Operating Systems	The system should be able to run smoothly on Windows and Linux as per the requirements documents.
3	Mobile platform	The materialize framework should be a sufficient addition that will make the system somewhat responsive. However some mobile browsers such as the Opera Mini mobile web browser may raise some compatibility issues.

4.1.5 APPLICATION PERFORMANCE

Test No.	Description	Expected Results
1	Response Time	The users of the system should get real time data upon any value changes and a page refresh should not take more than 3-5 seconds.
2	Data collision	All data in the system should be consistent with no discrepancies to the users. Data should be the same for all users of the same type.
3	Stress Testing	The system is expected to maintain a steady load state if multiple users are all accessing and requesting data from the system at the same time.

4.1.6 PLATFORM SECURITY

Test No.	Description	Expected Result
1	Test internal URLs by pasting them directly without login	The system should redirect you to the login page as you do not have the credentials and session to access the system
2	URL pattern crossing	If you are logged in as one type of user, try accessing the URL pattern of another type of user. The system is expected to respond by redirecting you to the login page as you do not have the rightfully assigned role to access that particular URL pattern.
3	Invalid Credentials	The system should be able to deny you access if you provide incorrect credentials.
4	Test if the back button works after logging out of them system	Once a user has logged out of the system and try pressing the back button to go back into the system, they should be redirected to the login page as they have already invalidated their session.