# Chapter 5 : Software Testing Documentation

## 5.1 Introduction

### Purpose

This chapter provides details related to testing of the system, including the scope of testing, test plan, test case and test report. Through testing, system defects can be discovered and fixed, allowing for better software quality. Using the contents of this chapter, developers and testers can work together to ensure that the system is thoroughly tested and functions as intended.

### Scope of Testing

* + - 1. **Test Levels**

The following four levels of testing are performed: Unit Test, Integration Test, System Test and Acceptance Test.

|  |  |
| --- | --- |
| **Test Level** | **Description** |
| Unit Test | Unit Tests are performed to ensure that individual units of source code function as intended. Unit tests are primarily used to test complicated algorithms and automatically test important functions. Jasmine is used to perform unit tests. |
| Integration Test | Integration Tests are performed to test whether combined units function as intended. |
| System Test | System Tests are performed to test the system as a whole and determine whether or not the system meets the requirements. No specific tools are used for system tests. |
| User Acceptance Test | User Acceptance Test is performed both by team members and other people in order to further ensure that the system works as intended, meets both functional and non-functional requirements, making sure that users feel comfortable when using the system. In addition, Acceptance Test helps to discover bugs and issues not found in the other types of tests. |

*Table 5-1: Test levels*

### Types of Testing

The following types of testing are performed:

* + - * + Unit testing
        + Regression testing
        + Function testing
        + GUI testing

### Range of Testing

All features as described in the Software Requirement Specification are tested.

## Test Plan

### Testing Tools and Environment

* + - 1. **Testing Tools**

The following tools are used for testing:

|  |  |
| --- | --- |
| **Tool** | **Purpose** |
| Google Chrome v62.0 | Use to view the web page, bug logging page, etc.. |
| Jamine | Use to perform unit tests. |

*Table 5-2: Testing tools*

### Testing Environment

The following table describes the testing environment:

|  |  |  |
| --- | --- | --- |
| **Type of Testing** | **Software** | **Hardware** |
| Unit Test | Visual Studio Code  Jamine | * Windows 10 Edu * Intel Core i3 1.9 Ghz * 8GB Memory * 500GB Disk Space |
| Integration Test | Visual Studio Code  Jetbrain WebStorm Google Chrome v62.0 | * Windows 10 Edu * Intel Core i3 1.9 Ghz * 8GB Memory * 500GB Disk Space * Screen resolution: 1366 x 768 |
| System Test |
| User Acceptance Test | The same as System Test, along with other software and hardware that are different for each user that participated. This includes a wide variety of software and hardware, ranging from PCs running on different types of operating systems to many different mobile devices. A full list of software and hardware used by all testers who participated in User Acceptance Test is not known and not publicly disclosed. | |

*Table 5-3: Testing environment*

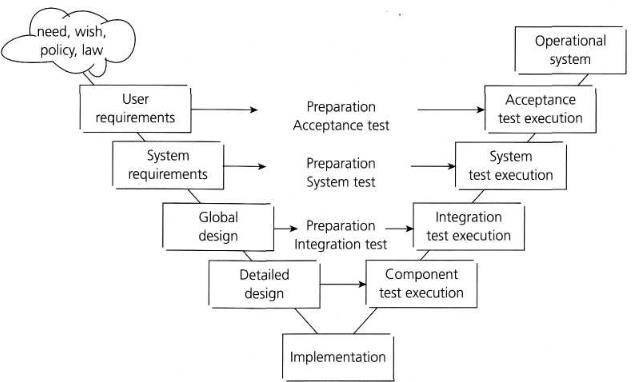
### Resources and Responsibilities

|  |  |  |
| --- | --- | --- |
| **ID** | **Resource** | **Responsibilities** |
| 1 | Project Manager | * Review test cases and test report. * Review overall quality of the project. * Participate in Acceptance Test. |
| 2 | Tester | * Create test plan * Perform tests * Write test reports * Log bugs |
| 3 | Developer | * Create and perform Unit Tests. |

*Table 5-4: Resources and responsibilities*

### Test Strategy

* + - 1. **Test Model**

This project follows the V-Model for testing.

*Figure 5-1: V-Model (Rex Black, Dorothy Graham, Erik van Veenendaal. Foundations of Software Testing: ISTQB Certification. Google Books. 2012)*

### Test Types

The following types of testing are performed:

* + - * + Unit testing

Tests individual methods to ensure that they work correctly.

Unit test has to reach 100% code coverage for the units that are tested.

Unit testing is primarily used to test complicated functions and algorithms.

* + - * + Regression testing

Regression testing ensures that bugs are fixed.

Regression testing ensures that the fixing of a bug does not impact other parts of the system.

* + - * + Functional testing

Tests whether or not functions of the system work as intended.

As opposed to unit testing, functional testing is a type of black-box testing and as such, internal structure is rarely considered.

* + - * + GUI testing

Tests whether or not elements of the websites appear and work correctly, and are positioned correctly.

Tests whether or not the website is usable and appears as intended with various browser sizes:

1440 x 900

1366 x 768

Checks whether messages, such as error messages are displayed correctly.

Checks whether fonts appear correctly and are readable.

Checks whether images appear correctly.

### Test Levels

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Type** | **Test Level** | | | |
| **Unit**  **Test** | **Integration**  **Test** | **System**  **Test** | **User Acceptance**  **Test** |
| Unit testing | ✓ | ✓ | ✓ |  |
| Regression testing | ✓ | ✓ | ✓ |  |
| Functional testing |  | ✓ | ✓ | ✓ |
| GUI testing |  |  | ✓ | ✓ |

*Table 5-5: Test levels*

### Test Schedule

|  |  |  |
| --- | --- | --- |
| **Test Schedule** | **Start Date** | **End Date** |
| *Iteration 1* |  |  |
| Unit testing | 2017/10/15 | 2017/10/16 |
| Functional testing | 2017/10/16 | 2017/10/24 |
| GUI testing | 2017/10/18 | 2017/10/24 |
| *Iteration 2* |  |  |
| Unit testing | 2017/11/19 | 2017/11/20 |
| Functional testing | 2017/11/20 | 2017/11/27 |
| GUI testing | 2017/11/21 | 2017/11/27 |
| Acceptance testing | 2017/11/26 | 2017/12/11 |

*Table 5-6: Test schedule*

### Deliverables

|  |  |  |
| --- | --- | --- |
| **Deliverables** | **Responsibilities** | **Completion date** |
| Test plan | Test Leader | 2017/10/10 |
| Test cases | Testers | 2017/10/14 |
| Test case review | Testers + Project Manager | 2017/10/15 |
| Test data | Testers | 2017/11/26 |
| Defect report | All members | 2017/11/27 |
| Final test report | Project Manager | 2017/11/27 |

*Table 5-7: Deliverables*

### Features to be Tested

All features described in the list of use cases in the Software Requirement Specification are to be tested.

## Test Approach

### Integration and System Test

Integration and system tests are done by testers to ensure that combined units work correctly and that the system as a whole functions as intended. Each test case is tested using Google Chrome v62.0.

GUI testing is also done during this process to ensure that elements load and function correctly, text is readable, and the website interface looks good in various browser sizes.

Test case details are described in the file **PSM\_TestReport.xlsx**.

### User Acceptance Testing

User Acceptance testing is done both by the development team and by other people, primarily the supervisor and friends of the development team. Testers who participate in User Acceptance testing use a wide variety of software and hardware, often without any guidance. This ensures that the system not only works correctly but is also good in terms of usability, as in, the website interface is simple, features are easy to find, and it is easy to use the system without reading instructions or asking other people how to use it. User Acceptance testing also helps to discover bugs that are not found in the other test levels, as testers usually use the system freely without following any testing guidelines, it is easy to come across bugs that the development team is prone to miss.

In addition, two people with color blindness also participate in User Acceptance testing to ensure that the colors used in the website are colorblind-friendly.

The development team uses the following checklist as a guideline for User Acceptance testing:

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Checklist** | **Yes** | **No** |
|  | **General** |  |  |
| CL-001 | Text does not have grammatical and spelling errors. |  |  |
| CL-002 | All buttons are functional. |  |  |
| CL-003 | All mandatory fields are validated and indicated by asterisk (\*) symbol. |  |  |
| CL-004 | All error messages are displayed using red color. |  |  |
| CL-005 | All "Delete" functions ask for confirmation. |  |  |
| CL-006 | All numeric values are formatted properly. |  |  |
|  | **GUI and Usability** |  |  |
| CL-007 | Screen design follows the standard of the project. |  |  |
| CL-008 | Features are intuitive and easy to use. |  |  |
| CL-009 | All important features can be easily accessed. |  |  |
| CL-010 | All fields on page (e.g. text box, radio options, dropdown lists) are aligned properly. |  |  |
| CL-011 | Information is presented in the order that the user needs it. |  |  |
| CL-012 | The static text is clear, concise, and meaningful. |  |  |
| CL-013 | Use abbreviation only when space is limited. |  |  |
| CL-014 | All icons and elements are flat. |  |  |
|  | **Database** |  |  |
| CL-015 | Correct data is getting saved in database upon successful page submit. |  |  |
| CL-016 | Values columns are not accepting null values. |  |  |
| CL-017 | Radio button options are saved correctly in database. |  |  |
| CL-018 | Database fields are designed with correct data type and data length. |  |  |
| CL-019 | Null values are not allowed for Primary key column. |  |  |