**Software Test Plan for**

**<<Advanced Web Applications Project>>**

**Team 10**

1. Thong Dang. Email : [t1daph00@students.oamk.fi](mailto:t1daph00@students.oamk.fi)

2. Jere Muikku. Email : [t0muje00@students.oamk.fi](mailto:t0muje00@students.oamk.fi)

3. Kiet Le. Email : [t1lean02@students.oamk.fi](mailto:t1lean02@students.oamk.fi)

4. Bao Tran. Email : [t1trxu00@students.oamk.fi](mailto:t1trxu00@students.oamk.fi)

1. INTRODUCTION

The purpose of this Software Test Plan is to develop a test plan to test all functionalities of the Advanced Web Applications Project, which is implemented by Team 10.

1. TEST PLAN.

* 1. UI/UX
* Responsiveness: The UI should scale if the browser window is resized. Text and visualizations must be readable. Smallest supported window size is 600px horizontal.

2.2. Functionality

- Password validation: Check if user gives a valid passwd.

The valid password should contain at least 8 characters with

numbers and letters…

- Email validation: Check if user gives a valid email. The

valid email should contain letters, numbers and @ symbol…

- Username validation: Check if user give a valid username.

The username should be alphanumeric and contain 5-12

characters.

- Chart Data Change: Check that the data will be shown and changed based on the data from the database if it’s changed.

-Register/Login functionality when application deployed to Render.

2.3. Security

- Check that created password is hashed, the valid password should be at

least 8 characters, includes uppercase, lowercase, number and special

character.

- Check that jsonwebtoken created, when register new user and when

User login with created username and password.

- Check that user can delete his/her own account with jsonwebtoken

and userId.

Changelog

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Change Date** | **By** | **Description** |
| 0.1 | 31.10.2020 | JK | Initial document creation |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contents

[1. Terms/Acronyms 3](#_Toc85035757)

[2. Introduction 4](#_Toc85035759)

[2.1. Scope 4](#_Toc85035760)

[2.1.1. In scope 4](#_Toc85035761)

[2.1.2. Out of scope 4](#_Toc85035762)

[2.2. Quality Objective 4](#_Toc85035763)

[2.3. Roles and Responsibilities 4](#_Toc85035764)

[3. Test methodology 5](#_Toc85035765)

[3.1. Overview 5](#_Toc85035766)

[3.2. Test levels 5](#_Toc85035767)

[3.3. Bug triage 5](#_Toc85035768)

[3.4. Suspension Criteria and Resumption Requirements 5](#_Toc85035769)

[3.5. Test Completeness 5](#_Toc85035770)

[4. Test Deliverables 7](#_Toc85035773)

[5. Resource & Environment Needs 8](#_Toc85035774)

[5.1. Test Environment 8](#_Toc85035775)

[5.2. Testing Tools 8](#_Toc85035776)

1. Terms/Acronyms

A mention of any terms or acronyms used in the project

|  |  |
| --- | --- |
| TERM/ACRONYM | DEFINITION |
| API | Application Program Interface |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |



# Introduction

Brief introduction of the test strategies, process, workflow and methodologies used for the project.

## Scope

### In scope

Scope defines the features, functional or non-functional requirements of the software that will be

### Out of scope

Out Of Scope defines the features, functional or non-functional requirements of the software that will NOT be tested

## Quality Objective

Make a mention of the overall objectives that you plan to achieve with your testing

Some objectives of your testing project could be:

Ensure the Application Under Test conforms to functional and non-functional requirements

Ensure the AUT meets the quality specifications defined by the client

Bugs/issues are identified and fixed before going live

## Roles and Responsibilities

Description of the Roles and responsibilities of different team members.

# Test methodology

## Overview

Mention the reason of adopting a particular test methodology for the project. The test methodology selected for the project could be

WaterFall

Iterative

Agile

Extreme Programming

Etc.

The methodology selected depends on multiple factors.

## Test levels

Test Levels define the Types of Testing to be executed on the Application Under Test (AUT). The Testing Levels primarily depends on the scope of the project, time and budget constraints.

## Bug triage

The goal of the triage is to

To define the type of resolution for each bug

To prioritize bugs and determine a schedule for all “To Be Fixed Bugs’.

## Suspension Criteria and Resumption Requirements

Suspension criteria define the criteria to be used to suspend all or part of the testing procedure while Resumption criteria determine when testing can resume after it has been suspended

## Test Completeness

Here you define the criteria’s that will deem your testing complete.

For instance, a few criteria to check Test Completeness would be

100% test coverage

All Manual & Automated Test cases executed

All open bugs are fixed or will be fixed in next release



# Test Deliverables

Here mention all the Test Artifacts that will be delivered during different phases of the testing lifecycle.

**Test Plan** (this document filled)\*

**Test Cases** (see example and separate excel file)\*

Requirement Traceability Matrix

**Bug Reports** (see example and separate excel file)\*

Test Strategy

Test Metrics

\*Documents that are in bold are minimum requirement for this course!

# Resource & Environment Needs

## Test Environment

The minimum hardware requirements that will be used to test the application. Following software’s are required in addition to client-specific software.

## Testing Tools

Make a list of tools required to test the project.

Server, database, operating system(s), test tools, network requirements, hw requirements etc.

**Test case example**

Test Case ID: 1

Test Priority (Low/Medium/High):

Module Name:

Test Title:

Description:

Test Designed by:

Test Designed date:

Test Executed by:

Test Execution date:

Pre-conditions:

Dependencies:

Post-conditions:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Test Steps | Test Data | Expected Result | Actual Result | Status (pass/fail) |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
| 5 |  |  |  |  |  |
| 6 |  |  |  |  |  |
| 7 |  |  |  |  |  |
| 8 |  |  |  |  |  |
| 9 |  |  |  |  |  |
| 10 |  |  |  |  |  |

Notes:

**Bug report example**

|  |  |
| --- | --- |
| ID: |  |
| Name: |  |
| Reporter: |  |
| Date: |  |
|  |  |
| Summary |  |
| URL |  |
| Platform |  |
| Operating system |  |
| Browser |  |
| Severity (low/medium/high) |  |
| Assigned to: |  |
| Priority: |  |

|  |
| --- |
| Description: |
|

|  |
| --- |
| Steps to reproduce: |
|

|  |
| --- |
| Expected result: |
|

|  |
| --- |
| Actual result: |
|

|  |
| --- |
| Notes: |
|