Linnéuniversitetet

Kalmar Växjö

Test Plan

Author: Irene Koech
Computer Science
Course code: 1dv600

Table of Contents

1 Introduction

The purpose of the test plan document to identify requirements from the previous assignment #2, create test cases for the use cases, implementing the unit test for the test cases and implementing API tests. Finally, illustrating the results.

1.1 Background

A partially half-finished web application has been given, which is the basis for a library system with the purpose to create, update, read and delete books. The web application based on two modules. The first is a completed front-end that will be used to test the code developed throughout the process. The second module is partially completed and requires more back-end development for the library system to be fully functional. Stack application meaning communication between the both modules through http request that is given API. The implementation will be using single page application architecture through API. Execution will be use in a virtual machine is also required to execute the code from the back-end. This virtual machine will be run by the software Vagrant.

1.2 Stakeholder's goals

Client\Customer: A fully functioning and tested CRUD functionality of the system
Developer: Fully tested and working functional system which should work according to API.

1.3 Scope

Test plan will apply to Unit testing, function testing and API testing. Implementation and iteration

2   Roles and Responsibilities

This project is done by one person, the development process is based on t apparently a postman will be do the API test using to execute the code by using Atom as an editor for the sever development.

2.1 Test Requirements

Most requirements that are needed includes the functionality of the testing and the uses cases. The requirements will be as following:
- Use case 1: GetBooksResource- Get a book in the library
- Use case 2: RemoveBookResource- Add a book to the library

2.1.1   Unit Test

Unit testing will use this class
- Use case 1: book
- Use case 2: catalog
- Use case 2: bookDAO

Resources are:
- Use case 1: GetBooksResource
- Use case 2: RemovebookResource

2.1.2 API Test

The objective in the api testing is the application should respond according to the Api, in this task we will need to create automotive test and a mock for testing the independencies and the server needs to be running before the execution.
The requirements in api testing will use this classes:
GetBooksResource
AddBookResource

3   Milestone

Milestone task	Efforts	Start Date	End Date
Test plan	30 mins	27 March 2018	27 March 2018
Test Design	30 mins	27 March 2018	27 March 2018
Implementation	30 mins	27 March 2018	27 March 2018
Execution	60 mins	27 March 2018	27 March 2018