



Assignment 3

Test Plan



4 Mar, 2017

Semester: Spring 2017
Course: Software Technology
Author: Sarpreet Singh Buttar

Contents

1	Introduction	1
1.1	Purpose	1
1.2	System Background	1
1.3	Stakeholder’s Goals	1
1.4	Scope	1
2	Requirements for Test	2
2.1	Function Testing	2
2.2	Unit Testing	2
2.3	API Testing	2
3	Test Strategy	3
3.1	Function Testing	3
3.2	Unit Testing	3
3.3	API Testing	3
4	Resources	4
4.1	Man Power	4
4.2	Tools	4
5	Milestones	4
6	Test Plan	5

1 Introduction

1.1 Purpose

The purpose of this test plan document is to support the followings objectives:

- Identifying requirements from assignment # 2
- Creating test cases for the use cases which have been identified in assignment #2
- Implementation of the unit tests for the test cases
- Implementation of API tests (Only for those requirements which were implemented during assignment #2)
- Illustrated the tests result

1.2 System Background

The given web application is a book library system and its purpose is to create, read, update and delete (CRUD) a book. It is a half-finished web application consists of two modules, the first module is a fully functional front end (client) which we will use to verify the code that we will write during the development of the partially implemented back end (server). The application stack is developed in a modern way which means these two modules will communicate via HTTP request specified in a given API. In result, no rendering is performed on server side because front end is implemented using Single Page Application (SPA) architecture and the only way to know how to communicate is to follow the API. Furthermore, for executing the application server, a virtual machine is required that is managed by Vagrant whereas the client side is executed in web browser.

1.3 Stakeholder's Goals

Customers	Developers
Fully working and tested CRUD functionality of the system	All the functionality should work according to the API as well as fully tested

1.4 Scope

This test plan applies to Unit Testing, Function Testing and API Testing. All these tests will be implemented and executed in this iteration. Information regarding the aim of these tests can be find in **Section 2** (Requirements for Test).

2 Requirements for Test

The listing below (uses cases, functional-requirements and non-functional-requirements) identifies all items that have been identified as targets for testing. It represents what will be tested.

2.1 Function Testing

Requirement #	Use Case
2.1.1	Use case 1 (Get a list of books)
2.1.2	Use case 2 (Delete a book)

2.2 Unit Testing

Requirement #	Class
2.2.1	book
2.2.2	catalog
2.2.3	booksDAO
2.2.4	GetBooksResources
2.2.5	RemoveBookResource

2.3 API Testing

Requirement #	Class
2.3.1	GetBooksResource
2.3.2	RemoveBookResource

3 Test Strategy

The Test Strategy presents a recommended approach of testing the identified testing-targets. The previous section described what are the testing-targets. This section will describe how it will be tested. The main considerations for the test strategy are the techniques to be used and the criterion for knowing when the testing is completed.

3.1 Function Testing

Test objectives	Ensure proper functioning of the application
Techniques	Execute each use case flow and identify test target using valid and invalid inputs data to verify the expected result
Completion criteria	All the tests should be executed and failure tests should have to be reported
Special consideration	None

3.2 Unit Testing

Test objectives	Verify the proper functioning of the classes
Techniques	Create and run the unit tests. In addition, mock the dependencies
Completion criteria	All the tests should be executed and failure tests should have to be reported
Special consideration	The server must be running before the execution of tests

3.3 API Testing

Test objectives	Ensure the application response according to the API
Techniques	Create automotive test and mock the dependencies
Completion criteria	System must response according to API and failure tests should have to be reported
Special consideration	None

4 Resources

Since it is individual assignment, so all the roles will be played by myself.

4.1 Man Power

Role	Resources	Responsibilities
Test Designer	1	Identify and implement test cases
Tester	1	Executes tests
Programmer	1	Implement some missing functionality (if needed)

4.2 Tools

Name	Type	Purpose
VS Code	Editor	Implementation
Vagrant	Virtual Machine	Code executor
Postman	API Tester	Function Testing

5 Milestones

Planning and working will be done in an agile-way as defined in the test-plan.

Milestone task	Efforts	Start data	En date
Test Plan	30 m	4 Mar 2017	4 Mar 2017
Test Design	30 m	4 Mar 2017	4 Mar 2017
Test Implementation	1 h	4 Mar 2017	4 Mar 2017
Test Execution	30 m	4 Mar 2017	4 Mar 2017
Test Evaluation	30 m	4 Mar 2017	4 Mar 2017

6 Test Plan

Test Type	Goals	Requirement #	Start data	End date
Function Testing	Test UC1 and UC2	<ul style="list-style-type: none">• 2.1.1• 2.1.2	4 Mar 2017	4 Mar 2017
Unit Testing	Implementing unit tests for all the classes	<ul style="list-style-type: none">• 2.2.1• 2.2.2• 2.2.3• 2.2.4• 2.2.5	4 Mar 2017	4 Mar 2017
API Testing	Test the resources	<ul style="list-style-type: none">• 2.3.1• 2.3.2	4 Mar 2017	4 Mar 2017