

Test Plan Report

(Name of the Product)

Books management System

Prepared by:

(Names of Group Members)

| | |
|---------------|------|
| Rutik Rojekar | A-25 |
| Sohil Sarode | A-26 |
| Vidit Bapat | A-27 |
| Aditya Gade | A-28 |

TABLE OF CONTENTS

1.0 Introduction

2.0 Objectives and Tasks

2.1 Objectives

2.2 Tasks

3.0 Scope

4.0 Testing Strategy

4.1 Unit Testing

4.2 Integration Testing

5.0 Hardware Requirements

6.0 Software Requirements

7.0 Features To Be Tested

8.0 Features Not To Be Tested

9.0 Deliverables

10.0 Tools

11.0 Approvals

1.0 INTRODUCTION

The Books Management System is a system to record the information of books in a store. The books' information is in the form of following attributes : title, author, year of publication and the ISBN number of the book. Book stores require this kind of systems to keep track of books present in the store and retrieve book information easily. User can insert new books, update existing ones, search for them and delete them as well.

2.0 OBJECTIVES AND TASKS

2.1 Objectives of Test Plan

Objective of the test plan is to describe scenarios to be tested and covering corner cases of the application as well.

2.2 Tasks of Test Plan

| Sr. No . | #Business Requirements (BR) | #Functional Requirements (FR) | #Test Scenarios (TS) | #Testing Approaches /Strategies (TA) |
|----------|---------------------------------------|-------------------------------|---------------------------------------------------------------------|----------------------------------------------------------------------------------|
| 01 | Operations on existing book records | Updation, Searching, Deletion | Update book, Search a book, Delete a book, validation on all fields | Functional Testing (Standard Testing) - Unit Testing - Integration Testing |
| 02 | Operations to create new book records | Insertion, View All | Insert a book, View all books, validation on all fields | |

3.0 SCOPE

General

Scope of the Test Plan will be testing all the fundamental validation units and modules , and further test the integration of all the CRUD operations on the application.

4.0 TESTING STRATEGY

1. **Functional testing**
2. **Unit testing**
3. **Integration Testing**

| Approach | Type of Testing | Manual Testing | | Automated Testing on Device | Tools/APIs/Libraries |
|---------------------------------------|---------------------|----------------|----------------|-----------------------------|---------------------------------------------|
| | | Using Device | Using Emulator | | |
| Standard Testing (Functional Testing) | Unit Testing | Yes | No | No | 1. Unittest (python unit testing framework) |
| | Integration Testing | Yes | No | No | |

4.1 Unit Testing

| | |
|----------------------------|----------|
| MODULE/FUNCTIONALITY NAME: | Insert |
| UNIT/CLASS: | Database |
| CREATED BY: | - |
| DATE OF CREATION: | - |
| DATE OF REVIEW: | - |

| TEST CASE ID | TEST CASE | PRECONDITION | TEST DATA | EXPECTED RESULT | POST CONDITION | ACTUAL RESULT | STATUS (PASS/FAIL) |
|--------------|---------------------------------------------|--------------|--------------------------------------------------------------------|--------------------------|----------------------|---------------------|--------------------|
| 1 | Non numeric year as input | - | insert(title="book",author="author",year="abcd",isbn="isbn0001") | Input should be rejected | Insert return false | Input is rejected | Pass |
| 2 | Empty fields in input | - | insert(title="",author="",year="1990",isbn="") | Input should be rejected | Insert returns false | Input is rejected | Pass |
| 3 | Floating point value in year field as input | - | insert(title="book",author="author",year="1922.6",isbn="isbn0001") | Input should be rejected | Insert returns false | Input is rejected | Pass |
| 4 | Too short title as input | - | insert(title="b",author="author",year="1989",isbn="isbn0001") | Input should be rejected | Insert returns false | Insert return true | Fail |
| 5 | Too short author as input | - | insert(title="book",author="a",year="1999",isbn="isbn0001") | Input should be rejected | Insert returns false | Insert returns true | Fail |

| | |
|-----------------------------------|----------|
| MODULE/FUNCTIONALITY NAME: | View |
| UNIT/CLASS: | Database |
| CREATED BY: | - |
| DATE OF CREATION: | - |
| DATE OF REVIEW: | - |

| TEST CASE ID | TEST CASE | PRECONDITION | TEST DATA | EXPECTED RESULT | POST CONDITION | ACTUAL RESULT | STATUS (PASS/FAIL) |
|--------------|----------------------|----------------------------------|-----------|---------------------------------|--------------------------------|-----------------------|--------------------|
| 6 | View all the records | Atleast one record present in DB | view() | All records should be extracted | Pre condition should hold true | Records are extracted | Pass |

| | |
|-----------------------------------|----------|
| MODULE/FUNCTIONALITY NAME: | Search |
| UNIT/CLASS: | Database |
| CREATED BY: | - |
| DATE OF CREATION: | - |
| DATE OF REVIEW: | - |

| TEST CASE ID | TEST CASE | PRECONDITION | TEST DATA | EXPECTED RESULT | POST CONDITION | ACTUAL RESULT | STATUS (PASS/FAIL) |
|--------------|-------------------------------|--------------|------------------------|--------------------------|----------------------|-------------------------------------|--------------------|
| 7 | No parameter passed to search | - | Search(“”, “”, “”, “”) | Query should be rejected | Search returns false | Query is rejected and returns false | Pass |

4.2 Integration Testing

| | |
|-----------------------|------------------------|
| PROJECT NAME: | Book Management System |
| MODULE/FUNCTIONALITY: | INSERT |
| CREATED BY: | - |
| DATE OF CREATION: | - |
| DATE OF REVIEW: | - |

| TEST CASE ID | TEST CASE | PRE-CONDITION | TEST DATA | EXPECTED RESULT | POST CONDITION | ACTUAL RESULT | STATUS (PASS/FAIL) |
|--------------|----------------------------|---------------|-------------------------------------------------------------------|--------------------|-------------------------------------------|--------------------|--------------------|
| 8 | INSERT a valid book record | - | Title="test", Author="test", year="1990", isbn="ISBN000" | Record is inserted | The record inserted is found using search | Record is inserted | Pass |

| | |
|-----------------------|------------------------|
| PROJECT NAME: | Book Management System |
| MODULE/FUNCTIONALITY: | UPDATE |
| CREATED BY: | - |
| DATE OF CREATION: | - |
| DATE OF REVIEW: | - |

| TEST CASE ID | TEST CASE | PRE-CONDITION | TEST DATA | EXPECTED RESULT | POST CONDITION | ACTUAL RESULT | STATUS (PASS/FAIL) |
|--------------|--------------------------------|--------------------------------|-----------------------------------------------------------------------------------------------|-------------------|--------------------------------|-------------------|--------------------|
| 9 | UPDATE an existing book record | A valid entry is present in DB | Title="test", Author="test", year="1990", isbn="ISBN000", updatedisbn = "isbn001" | Record is updated | The existing record is updated | Record is updated | Pass |

| | |
|------------------------------|------------------------|
| PROJECT NAME: | Book Management System |
| MODULE/FUNCTIONALITY: | DELETE |
| CREATED BY: | - |
| DATE OF CREATION: | - |
| DATE OF REVIEW: | - |

| TEST CASE ID | TEST CASE | PRE-CONDITION | TEST DATA | EXPECTED RESULT | POST CONDITION | ACTUAL RESULT | STATUS (PASS/FAIL) |
|--------------|--------------------------------|--------------------------------|-------------------------------------------------------------------|-------------------|--------------------------------|-------------------|--------------------|
| 10 | DELETE an existing book record | A valid entry is present in DB | Title="test", Author="test", year="1990", isbn="ISBN000" | Record is deleted | The existing record is deleted | Record is deleted | Pass |

| | |
|------------------------------|------------------------|
| PROJECT NAME: | Book Management System |
| MODULE/FUNCTIONALITY: | SEARCH |
| CREATED BY: | - |
| DATE OF CREATION: | - |
| DATE OF REVIEW: | - |

| TEST CASE ID | TEST CASE | PRE-CONDITION | TEST DATA | EXPECTED RESULT | POST CONDITION | ACTUAL RESULT | STATUS (PASS/FAIL) |
|--------------|--------------------------------|--------------------------------|-------------------------------------------------------------------|---------------------|----------------|---------------------|--------------------|
| 11 | SEARCH an existing book record | A valid entry is present in DB | Title="test", Author="test", year="1990", isbn="ISBN000" | Record is extracted | - | Record is extracted | Pass |

5.0 HARDWARE REQUIREMENTS

1. 4GB RAM
2. 100MB HDD
3. i5- 8th gen processor

6.0 SOFTWARE REQUIREMENTS

1. Ubuntu 20.04 (focal fossa)
2. Python 3
3. Python tkinter
4. Sqlite DB
5. Python unittest

7.0 FEATURES TO BE TESTED

1. Insertion of records
2. Updation of records
3. Retrieval of records
4. Deletion of records
5. Validation of input

8.0 FEATURES NOT TO BE TESTED

1. GUI of the applications - (GUI testing is not under the scope of this test plan)

9.0 DELIVERABLES

| Deliverable | Date / Milestone |
|--------------------|------------------|
| Test Plan | - |
| Test Result Report | - |

10.0 TOOLS

1. Python modules - (unittest)

11.0 APPROVALS

| Name (In Capital Letters) | Signature | Date |
|---------------------------|-----------|------|
| 1. | | |