Library Project

Test Plan

Ivan Davidov

Table of Content

Overview…………………………………………………………………………………………………………………………………………….…..2

Scope……………………………………………………………………………………………………………………………………………………...2

Inclusions…….……………………………………………………………………………………………………………………………..2

Test Environments………………………………………………………………………………………………………………………2

Exclusions…..……………………………………………………………………………………………………………………………….2

Test Strategy…….……………………………………………………………………………………………………………………………………..2

Defect Reporting Procedure……………………………………………………………………………………………………………………3

Roles/Responsibilities………………………………………………………………………………………………………………………………4

Test Schedule…………………………………………………………………………………………………………………………………………..4

Test Deliverables……….…………………………………………………………………………………………………………………………….5

Pricing……………………………………………………………………………………………………………………………………………………..5

Entry and Exit Criteria………………………………………………………………………………………………………………………………5

Suspension and Resumption Criteria………………………………………………………………………………………………………..6

Tools………….…………………………………………………………………………………………………………………………………………….6

Risks and mitigations……………………………………………………………………………………………………………………………….6

Approvals…………………………………………………………………………………………………………………………………………………7

1. **Overview**

Quality is the degree to which a component, system or process meets specified requirements and/or user/customer needs and expectation. As part of the project, all functionalities on Library Project have to be tested.

This document serves as high level test planning document with details on the scope of the project, test strategy, test schedule and resource requirements, test deliverables and schedule.

1. **Scope**

The scope of the project includes testing all features of Library Project ([https://qa-task.immedis.com/)](https://qa-task.immedis.com/)%20) web application.

1. Inclusions:
   1. Static testing
      1. Review of Immedis\_QA\_Assignment\_2023\_\_2\_.docx
   2. Manual Functional testing
      1. Login;
      2. User module;
      3. Books module;
      4. Get a Book module;
   3. Automation testing of UI
      1. Creating of Test Suit for regression testing of UI
   4. API testing
2. Test Environment:

|  |  |
| --- | --- |
| Windows OS / Chrome | 114.0.5735.134 |
| TBD | Latest |
| TBD | Latest |
| TBD | Latest |

1. Exclusions:
   1. All the features except that are mentioned under “Inclusions” and in SRS Documentation (Immedis\_QA\_Assignment\_2023\_\_2\_.docx);
   2. Any third-party features;
2. **Test Strategy**

As discussed with Project Owner (PO) and Project Manager (PM), we need to perform Static testing, Manual testing, Automation testing and API testing, mentioned in the above Scope section.

As part of Functional Testing, we will follow the below approach for Testing:

1. Creation of Test Scenarios and Test Cases for the different features on scope. Every Test case must have ID, specific Name, Objective, Preconditions, Priority, Steps, Test Data and Expected Results.
   1. We will apply several Test Designing techniques while creating Test Cases:
      1. Equivalence Class Partition;
      2. Boundary Value Analysis;
      3. Decision Table Testing;
      4. State Transition Testing;
      5. Use Case Testing.
   2. We also use our expertise in creating Test Cases by applying the below:
      1. Error Guessing;
      2. Exploratory Testing.
   3. We prioritize the Test Cases.
2. Our Testing process, when we get an Application for Testing will be:
   1. Performing Smoke Testing to check whether the different and important functionalities of the application are working;
   2. We reject the build, if the Smoke Testing fails and will wait for the stable build before performing in depth testing of the application functionalities;
   3. Once we receive a stable build, which passes Smoke Testing, we perform in depth testing using the created Test Cases;
   4. We report the bugs in bug tracking tool;
   5. As part of the Testing, we will perform the test types of Testing below:
      1. Smoke Testing and Sanity Testing;
      2. Regression Testing and Retesting;
      3. Functionality Testing.
   6. We repeat Test Cycles until we get the quality product.
   7. Simultaneously we will prepare an Automation Test Suit in order to perform regression testing faster, and more reliable.
3. We will follow the below best practices to make our Testing better:
   1. Context Driven Testing – We will be performing Testing as per the context of the given application;
   2. Shift Left Testing – We will start testing from the beginning stages of the development itself, instead of waiting for the stable build;
   3. Exploratory Testing – Using our expertise we will perform Exploratory Testing, apart from the normal execution of the Test cases.
   4. End to End Flow Testing – We will test the end-to-end scenario which involve multiple functionalities to simulate the end user flows.
4. **Defect Reporting Procedure:**

During the test execution:

1. Any deviation from expected behavior by the application will be noted. If it cannot be reported as a defect, it’d be reported as an observation/issue or posed as a question;
2. Any usability issues will also be reported;
3. After discovery of a defect, it will be retested to verify reproducibility of the defect. Every deviation of an actual result from the expected result have to be registered in Bug Tracking system. Every Bug Report must have Summary, Affects version, Severity, Priority, Assignee (PO), Description (Steps to reproduce, Test Data, Expected Results and Actual Results), and Environment. Screenshots/Videos can help for better understanding the nature of the Bug.
4. Every day, at the end of the test execution, defects encountered will be sent along with the observation.

Note:

1. Test Cases will be created in Excell;
2. Test Cases/Test Cycles will be executed manually;
3. Defects will be documented in Word;
4. Automation Test Suit will be written in JavaScript and will be executed by Cypress testing tool
5. API testing will be written and executed with Postman
6. **Roles/Responsibilities**

|  |  |  |
| --- | --- | --- |
| Name | Role | Responsibilities |
| Name Surname | Product Owner | Escalations |
| Ivan Davidov | Team Lead /  Testing Engineer | * Create the test plan and get the signoff; * Interact with the application; * Create and Execute Test Cases; * Create and Execute Automation Test Suit for regression testing * Create and Execute API Testing * Report Defects; * Coordinate the test execution. Verify validity of the defects being reported; * Submit daily issues updates and summary defect reports to the PO; * Attend any meeting with PO; |

1. **Test Schedule**

Following is the test schedule planned for the project:

**SPRINT 1**

|  |  |
| --- | --- |
| Manual testing US1: As a user I want to be able to login into the system using correct username and password | Time Duration |
| Test Case Creation | 1st day of the SPRINT |
| Test Case Execution | 1st day of the SPRINT |
| Summary Reports Submission | 1st day of the SPRINT |
| Manual testing US2: As a system user with proper rights, I want to be able to see the “Users” module | Time Duration |
| Test Case Creation | 2nd day of the SPRINT |
| Test Case Execution | 2nd day of the SPRINT |
| Summary Reports Submission | 2nd day of the SPRINT |
| Manual testing US3: As a system user with proper rights, I want to be able to see the “Books” module | Time Duration |
| Test Case Creation | 3th day of the SPRINT |
| Test Case Execution | 3th day of the SPRINT |
| Summary Reports Submission | 3th day of the SPRINT |
| Manual testing US4: As a system user with proper rights, I want to be able to see the “Get a Book” module | Time Duration |
| Test Case Creation | 4th day of the SPRINT |
| Test Case Execution | 4th day of the SPRINT |
| Summary Reports Submission | 4th day of the SPRINT |
| Automation testing: Regression Test Suit of UI | Time Duration |
| Test Case Creation | 5th day of the SPRINT |
| Test Case Execution | 5th day of the SPRINT |
| Summary Reports Submission | 5th day of the SPRINT |
| API testing: API Test Suit | Time Duration |
| Test Case Creation | 6th day of the SPRINT |
| Test Case Execution | 6th day of the SPRINT |
| Summary Reports Submission | 6th day of the SPRINT |

1. **Test Deliverables**

The following are to be delivered to the client:

|  |  |  |
| --- | --- | --- |
| Deliverables | Description | Target Completion Date |
| Functional Test Cases | Test Cases created for the scope defined | See Test Schedule |
| Defect Reports | Detailed description of the defects identified along with screenshots and steps to reproduce on a daily basis | NA |
| Summary Reports | Summary Report  Bugs by Bug #  Bugs by Functional Area  Bugs by Priority | See Test Schedule |
| Regression Test Suit | Test Suit created for regression testing of UI | See Test Schedule |
| API Test suit | API Test Suit created for regression testing of UI | See Test Schedule |

1. **Pricing**

NA

1. **Entry and Exit Criteria**

The below are the entry and exit criteria for every phase of Software Testing Life Cycle (STLC):

1. Requirement Analysis
   1. Entry Criteria:
      1. Once the testing team receives the Requirements Documentation or details about the Project.
   2. Exit Criteria:
      1. List of Requirements are explored and understood by the Testing team;
      2. Doubts are cleared.
2. Test Planning:
   1. Entry Criteria:
      1. Testable Requirements derived from the given Requirements Documentation or Project details;
      2. Doubts are cleared.
   2. Exit Criteria:
      1. Test Plan Document (includes Test Strategy) is signed-off by the PO/Client.
3. Test Designing
   1. Entry Criteria:
      1. Test Plan Document is signed-off by the PO/Client.
   2. Exit Criteria:
      1. Test Scenarios and Test Cases Documents are signed-off by the PO/Client.
4. Test Execution:
   1. Entry Criteria:
      1. Test Scenarios and Test Cases Documents are signed-off by the PO/Client;
      2. Application is ready for Testing.
   2. Exit Criteria:
      1. Test Case Reports and Defect Reports are ready.
5. Test Closure:
   1. Entry Criteria:
      1. Test Case Reports and Defect Reports are ready.
   2. Exit Criteria:
      1. Test Summary Reports.
6. **Suspension and Resumption Criteria**

Based on the Client/PM decision, we will suspend and resume the Project. We will ramp up and ramp down the resources as per Client needs.

1. **Tools**

The following are the list of Tools we will be using in the Project:

1. MS Office to handle the documentation;
2. MS Office for Test Management;
3. MS Office for Bug Tracking;
4. VSCode for writing Test Suit for Automation regression test suit;
5. Cypress for executing Test Suit for Automation regression test suit;
6. Postman for executing Test Suit for API testing;
7. **Risks and mitigations**

The following are the list of risks possible and the ways to mitigate them:

1. Risk: Non-Availability of a Resource – Mitigation: Backup Resource Planning
2. Risk: Application Build is not working – Mitigation: Resources will work on other tasks
3. Risk: Less time for Testing – Mitigation: Ramp up the resources based on the Client needs dynamically, Prioritizing Test Cases.
4. **Approvals**

Team will send different types of documentation for Client/PO Approval like below:

1. Test Plan;
2. Test Scenarios;
3. Test Cases;
4. Reports.

Testing will only continue to the next steps once these approvals are done.