SHELVD

Test Plan

Version 1.4

4/4/2024

Revision History

Revision	Description of Change	Approved by	Date
1.0	Preliminary Version	Kenneth	27/3/2024
1.1	Outline, Introduction, Test Items, Software Risk Issues, Features to be Tested	Sufyan	04/04/2024
1.2	Features not to be Tested, Approach, Suspension Criteria, Deliverables, Environmental Needs, Staffing Training Needs, Responsibilities, Schedule, Planning Risks	Maw Htun	04/04/2024
1.3	Item Pass/Fail	Arun	04/04/2024
1.4	Approvals	Maw Htun	04/04/2024

1 Test Plan Identifier

The identifier for this system test plan for the ***shelvd*** web application project is ***shelvd-TPLV1.0-SL1-MTP1.***

Throughout the testing process we will be applying the test documentation specifications described in the IEEE Standard 829-2008 for software test documentation. The plan contains information about the resources and procedures of testing of ***shelvd***, and how the testing process is controlled, and configuration managed throughout the product lifecycle.

2 Introduction

This is the Master Test Plan for the ***shelvd*** web application project. This plan will address the features and elements in the app. The primary focus of this plan is to ensure that the ***shelvd*** application provides a functional and user-friendly experience without any inconveniences and bugs. The project will have three levels of testing, Unit, System/Integration and Acceptance. The details for each level are addressed in the approach section and will be further defined in the level specific plans. The acceptance testing is expected to take two (2) weeks from the date of application delivery from system test and is to be done in parallel with the current application process.

3 Test Items (Functions)

The following is a list, by version and release, of the items to be tested:

A. ***shelvd*** software, initial version to be Version 1.0

   A detailed listing of programs, databases, screens and reports will be provided in the system and detailed design documents.

B. ***shelvd*** Systems Requirement Specification document

C. ***shelvd*** Quality Plan document

D. ***shelvd*** Project Plan document

E. ***shelvd*** Software Configuration Management Plan document

4 Software Risk Issues

There are several parts of the ***shelvd*** application that must be checked as well.

A. Delivery of Third-Party Product: Assess risks associated with integrating third-party services or APIs for features like book recommendations.

B. New Version of Interfacing Software: Evaluate risks related to updating interfacing software used for user authentication and data retrieval.

C. Backup and Recovery of files, local databases and restart of the translation process, must be carefully checked

D. Modifications to Components with Past Failures: Assess risks linked to modifying components that previously experienced defects or failures.

E. Database security and access must be defined and verified

5 Features to be Tested

The following is a list of the areas to be focused on during testing of the application.

A. User account creation and login

B. Displaying of the different book lists: ‘Trending’, ‘Featured’, ‘Popular’, etc.

C. Search books

D. Creation of book collection

E. Editing of book collection (rename collection, add/remove books, etc.)

F. Sharing of book collection to other users

6 Features not to be Tested

The following is a list of the areas that will not be specifically addressed. All testing in these areas will be indirect as a result of other testing efforts.

A. Third-Party APIs
Features like book recommendations which rely on external APIs such as Google Books and Open Library will be tested only for integration, not for the external service’s functionality.

B. Stable Features
Features that have not changed from previous versions and have been validated in prior testing cycles will not be re-tested.

7 Approach (Strategy)

7.1 Testing Levels

The testing for the BookCatalogue project will consist of three primary levels: Unit, System/Integration, and Acceptance Testing.

Unit Testing will be conducted by developers as they complete each module. Each unit test must be passed and approved by the development team leader before it can be considered complete. Developers will provide documentation including test case lists, sample outputs, and any defect information to both the team leader and the quality assurance (QA) team for verification.

System/Integration Testing will be carried out by the QA team under the supervision of the test manager, with assistance from developers as needed. This phase will begin only after all critical defects from unit testing have been addressed. A module can proceed with up to two major defects provided they do not hinder further testing activities (i.e., there are effective workarounds).

Acceptance Testing will be conducted by selected end users alongside the test manager and the development team leader. This stage is scheduled to start after system testing concludes and will run in parallel with the ongoing deployment activities. Acceptance testing will only proceed once all critical and major defects have been rectified. The program may contain minor defects if they do not affect the overall operation and user experience.

The initial phase of acceptance testing will include a limited number of end users to ensure the application meets all business requirements and user expectations before a broader rollout. This limitation is addressed in the Planning Risks and Contingencies section.

7.2 Configuration Management/Change Control

Configuration management will be maintained through a version control system (e.g., Git) integrated with an issue tracking platform (e.g., JIRA). This system will manage changes and ensure that all team members are working with the correct versions of software components. Changes will be reviewed and approved by the project management team before being merged into the development branch.

The deployment from the development environment to the test environment will be controlled through automated pipelines, ensuring that only approved changes are tested.

7.3 Meetings

Regular testing status meetings will occur weekly, involving the QA team, test manager, development team leader, and project manager. These meetings will review the testing progress, discuss any significant issues, and adjust plans as necessary.

Ad-hoc meetings may be scheduled to address urgent issues or significant deviations from the test plan.

7.4 Measures and Metrics

Data collected during all testing phases will include:

- Defects by module and severity: Tracking where defects occur and their impact.
- Defect origins (Requirement, Design, Code): Understanding the source of defects to improve processes.
- Time spent on defect resolution: For critical and major defects only; minor defects may be aggregated.
- Test coverage metrics: Ensuring that all critical paths and functionalities are tested.

8 Item Pass/Fail Criteria

A. Test Item: User Account Creation and Login

Pass Criteria:

User Account Creation:

- User can successfully navigate to the account registration page from the homepage
- The registration form includes fields for entering required information such as username, email address, password, and optional fields for additional details.
- Upon submitting valid registration details, the system creates a new user account and redirects the user to the login page.
- An email confirmation is sent to the registered email address, containing a verification link or code to activate the account.

User Login:

- Registered users can access the login page and enter their credentials (username/email and password) to log in.
- The login process validates user credentials against the stored user database and grants access upon successful authentication.
- After successful login, the user is redirected to the application's dashboard or landing page.
- The system maintains user login status across sessions, allowing users to remain logged in until they explicitly log out or their session expires.

Fail Criteria:

User Account Creation:

- The account registration page fails to load or is inaccessible from the homepage.
- Errors occur during registration, such as validation errors for missing or invalid input data.
- Submission of registration details does not result in the creation of a new user account, or the process encounters technical errors.
- Confirmation email fails to send or is not received by the user within a reasonable time frame.

User Login:

- The login page is inaccessible or returns errors when users attempt to log in.
- User credentials are not validated correctly during the login process, resulting in authentication failures for valid credentials.
- Successful login does not redirect the user to the expected destination or results in a server error.
- User login status is not maintained across sessions, causing users to be logged out unexpectedly or unable to log in.

B. Test Item: Displaying Different Book Lists (e.g., 'Trending', 'Featured', 'Popular', etc.)

Pass Criteria:

Trending Books List:

- The "Trending" books list is prominently displayed on the homepage or dedicated section of the application.
- The list showcases books that are currently trending or experiencing increased user interest based on relevant metrics such as sales, ratings, or reviews.
- Books in the "Trending" list are updated dynamically to reflect real-time trends and user preferences.
- Each book entry in the "Trending" list includes relevant information such as title, author, cover image, and a brief description or summary.
- Clicking on a book in the "Trending" list redirects the user to the corresponding book details page for more information.

Featured Books List:

- The "Featured" books list is prominently displayed on the homepage or designated section of the application.
- Featured books are handpicked or curated by the application's editorial team to highlight noteworthy or recommended titles.
- The list includes diverse and high-quality books spanning different genres, themes, or categories.
- Each featured book entry includes eye-catching visuals, such as cover images or banners, to attract user attention.
- Users can easily navigate through the "Featured" books list and access additional details or purchase options for featured titles.

Popular Books List:

- The "Popular" books list is prominently displayed and easily accessible within the application's navigation or homepage layout.
- Popular books are determined based on metrics such as sales, ratings, reviews, or user engagement.
- The list showcases books that have garnered significant popularity and positive reception among users.
- Books in the "Popular" list are updated periodically to reflect current trends and preferences.
- Users can explore popular books, view detailed information, and engage with user-generated content such as reviews or ratings.

Fail Criteria:

Trending Books List:

- The "Trending" books list is missing or not displayed prominently on the application's homepage or designated section.
- Books listed as "Trending" do not accurately reflect current trends or user interests, leading to outdated or irrelevant recommendations.
- Clicking on a book in the "Trending" list results in broken links or redirects users to incorrect or unrelated content.
- The "Trending" books list fails to update dynamically, resulting in stale or unchanged content over time.

Featured Books List:

- The "Featured" books list is absent or not easily discoverable within the application's interface, leading to poor visibility for featured titles.
- Featured books do not represent a diverse range of genres, themes, or categories, limiting the appeal and relevance of the list to users.
- Visual elements such as cover images or banners are missing or improperly formatted, detracting from the visual appeal of featured titles.
- Navigation issues prevent users from accessing additional details or purchase options for featured books, resulting in a poor user experience.

Popular Books List:

- The "Popular" books list is not accessible or fails to load properly, preventing users from discovering popular titles.
- Popular books listed do not align with actual popularity metrics or user engagement data, leading to inaccurate recommendations.
- The "Popular" books list remains static and does not reflect changes in popularity trends or user preferences over time.
- Users encounter errors or broken functionality when attempting to interact with books listed in the "Popular" list, such as viewing details or submitting reviews/ratings.

C. Test Item: Search Books

Pass Criteria:

Search Functionality:

- The search functionality is prominently accessible within the application's interface, such as through a dedicated search bar or prominently displayed search icon.
- Users can enter search queries using keywords, titles, authors, or other relevant search criteria.
- The search functionality supports partial matching, allowing users to find books even with incomplete search queries.

Search Results Display:

- Search results are presented in a clear and organized manner, with each book entry displaying relevant information such as title, author, genre, cover image, and a brief summary.
- The search results page includes pagination or infinite scroll functionality, allowing users to navigate through multiple pages of search results seamlessly.
- Clicking on a book in the search results redirects the user to the corresponding book details page for more information or actions such as purchasing or adding to a wishlist.

Fail Criteria:

Search Functionality:

- The search feature is inaccessible or not prominently displayed within the application's interface, making it difficult for users to find and use.
- Users encounter errors or unexpected behavior when entering search queries, such as input validation errors or system crashes.
- Search functionality fails to return relevant results for valid search queries, or search results contain irrelevant or inaccurate information.
- Search results are not updated dynamically or fail to refresh when users modify their search queries or filters.
-

Search Results Display:

- Search results are not presented in a clear or organized manner, making it difficult for users to interpret or navigate through the results.
- Pagination or infinite scroll functionality is non-functional, preventing users from accessing additional search results beyond the initial page.
- Filtering or refining search results does not produce accurate or relevant outcomes, leading to incorrect or inconsistent results.
- Clicking on a book in the search results does not redirect the user to the expected destination or results in broken links or errors.

9 Suspension Criteria and Resumption Requirements

A. Critical System Failures

- Criteria for Suspension: If a critical system failure occurs that prevents any form of testing (e.g., database crashes, loss of network connectivity, severe application crashes).
- Resumption Requirements: Testing will resume once the system stability is restored and verified. A root cause analysis will be performed to ensure the same issue does not recur, and corrective actions must be implemented and reviewed before proceeding.

B. Major Security Breach

- Criteria for Suspension: If a major security breach occurs during testing that could potentially compromise test data or the integrity of the test results.
- Resumption Requirements: Testing will only resume after the breach has been fully contained, all vulnerabilities have been addressed, and security enhancements have been tested. A security audit will be conducted to validate the effectiveness of the corrective measures.

C. Unavailability of Critical Testing Tools

- Criteria for Suspension: If key testing tools or environments are unavailable for an extended period (e.g., due to licensing issues, hardware failures).

- Resumption Requirements: Testing can continue once the tools or environments are fully operational. The project timeline will be reassessed and adjusted to account for the delay. Additional resources may be allocated to maintain the project schedule.

D. High Defect Rates

- Criteria for Suspension: If the defect rate exceeds a predetermined threshold that makes further testing impractical (e.g., multiple major bugs in core functionalities that block further tests).
- Resumption Requirements: Testing will resume after a significant number of critical and major defects have been fixed, and a new build has been certified by the development team as stable for retesting. A review of the testing and development processes might be conducted to prevent future occurrences.

E. Regulatory Compliance Issues

- Criteria for Suspension: If during testing, it's found that the application does not comply with critical regulatory requirements which must be addressed to continue.
- Resumption Requirements: Testing will resume after adjustments are made to the application to ensure compliance. Compliance verification will be performed by a third party if necessary, and documented approval from the compliance team will be required.

F. Resource Unavailability

- Criteria for Suspension: Significant unavailability of key personnel or other critical resources (e.g., third-party service downtime) that prevents the execution of planned tests.
- Resumption Requirements: Testing will resume once adequate staffing levels are restored or the third-party services are back online. If delays are extensive, task reallocation or additional hiring may be considered.

For each of these criteria, detailed monitoring will be necessary to promptly identify potential triggers for suspension. Regular updates will be required to keep all stakeholders informed about the status of testing and any delays.

10 Test Deliverables

Test Deliverables are the test artifacts which are given to the stakeholders of a software project during the SDLC (Software Development Life Cycle). Some of the deliverables are provided before the testing phase commences and some are provided during the testing phase and rest after the testing phase is completed.

The deliverables included in this Test Plan are:

- Test Cases and Test Coverage Report
  Test cases are the set of positive and negative executable steps of a test scenario which has a set of pre-conditions, test data, expected result, post-conditions, and actual results. This also contains the test results and the summary of test execution activities.

- Test Plan Document

Test plan document is a document which contains the plan for all the testing activities to be done to deliver a quality product. The test Plan document is derived from the Product Description, SRS, or Use Case documents for all future activities of the project. It is usually prepared by the Test Lead or Test Manager.

- Revision Logs
  The revision history of all the documents will be duly documented and stored.

- Defect logs with solutions implemented
  All the defects identified will be documented and logged in the appropriate documents as well as the solution implemented for the respective defect.

11 Environmental Needs

For the testing of ***shelvd***, following specifications and requirements have to be met:

- Linux, Mac OS X, or Windows.
- git (used for source version control).
- Python (used by some of our tools)
- PostgreSQL database
- An appropriate web browser supporting JavaScript, preferably Chrome
- Node.js and bun

12 Staffing and Training needs

Training will be required to setup the application on the testers' machines as well as to get acquainted with the Flutter testing frameworks.

The Project Manager and Release Manager will be responsible for providing training on how to run the application locally and the structure of the code. The QA Engineers can consult the front end and back-end developers if they run into any problems. The QA Manager will be responsible for finding methods to train the testers.

13 Responsibilities

Role	Responsibilities
Project Manager	Deciding which features should be tested, monitoring the errors found, collecting and analyzing final testing report
Release Manager	Collecting details of test runs to make sure software meets requirements before deployment
DEV Front-End	Conducting white-box testing on the front-end components
DEV Back-End	Conducting white-box testing on the back-end components

QA Manager	Stating the risk and contingency plan for the different phases of the test. Monitoring testing activities and ensuring all testing resources are available, setting overall testing strategy. Conducting black-box testing, reporting test logs, generating additional test cases as needed

14 Schedule

Time has been allocated within the project plan for the following testing activities. The specific dates and times for each activity are defined in the project plan time line. The persons required for each process are detailed in the project time line and plan as well. Coordination of the personnel required for each task, test team, development team, management and customer will be handled by the project manager.

A. Review of Requirements document by test team personnel (with other team members) and initial creation of Inventory classes, sub-classes and objectives.
B. Development of Master test plan by test manager and test with time allocated for at least two reviews of the plan.
C. Review of the System design document by test team personnel. This will provide the team with a clearer understanding of the application structure and will further define the Inventory classes, sub-classes and objectives.
D. Development of System/Integration and Acceptance test plans by test manager and other essential personnel with time allocated for at least two reviews of the plans.
E. Review of the Detail design document(s) by test team personnel as required. This will provide the team with a clearer understanding of the individual program structure and will further define the Inventory classes, sub-classes and objectives.
F. Unit test time within the development process.
G. Time allocated for both System/Integration and Acceptance test processes.

15 Planning Risks and Contingencies

The risks and methods to mitigate them are as follows:

Risk	Contingency
Shortage of testers	In the event of an understaffed QA team, members from the development team and management will step in to assist with testing tasks to manage workload effectively.
Inadequate Training for Testers	The QA Manager and Lead Developer will ensure comprehensive training is provided to all testers to equip them with the necessary skills and knowledge for effective testing.

16 Approvals

Role - Name
Project Manager - Maw Htun
Release Manager - Crystal
QA Manager - Kenneth