Software Test Plan

Personal Finance Tracker

For user Self Service in Finance Tracker

version 1.0

Yarin Alkalai

QA College

|  |  |  |  |
| --- | --- | --- | --- |
| 1.0 | Test Plan Version | Personal Finance Tracker | **Product Name** |
| 1.0 | Test Plan Template Version | 1.0 | **Product Version** |

revision history

|  |  |  |
| --- | --- | --- |
| **date** | **comments** | **author** |
| 15/08/2025 | Initial draft | yarin |

test plan review/approval

|  |  |  |
| --- | --- | --- |
| **date** |  | **title** |
|  |  | Product manager |
|  |  | Program manager |
|  |  | Qa project lead |
|  |  | Qa team |
|  |  | Review team |

Table of contents

1. introduction
2. objectives
3. scope
4. Test items
5. Features to be tested
6. Features not to be tested
7. Test approach
8. Test environment
9. Test schedule
10. Test deliverables
11. Responsibilities
12. Entry and exit criteria
13. Risk and contingencies
14. Approval

**1. Introduction**

The purpose of this document is to outline the test plan for the user module in Personal Finance Tracker. system Version 1.0. This test plan defines the testing strategy and approach for ensuring the quality and functionality of the user module, which allows users to manage their transactions, budget, and monitor expenses and income status.

**2. Objectives**

- Verify the functionality and performance of the user module.

- Ensure all user interface elements and features work as expected.

- Validate the security and access control mechanisms.

- Confirm compatibility with supported browsers and operating systems.

**3. Scope**

- Functional testing of the user module.

- User interface testing.

- Security testing.

- Compatibility testing across different browsers and devices.

**4. Test Items**

- Transactions Module

- Budget Module

- Reports Module

- Export Module

**5. Features to be Tested**

**5.1 Transactions Module**

- Add Transaction

- Delete Transaction

**5.2** **Budget Module**

- Add Budget

- Delete Budget

**5.3 Reports Module**

- Expenses by Category graph

- Income vs Expenses graph

**5.4 Export Module**

- View Transactions

- Change Currency

- Export Transactions file (Exel)

**6. Features Not to be Tested**

- Local storage.

- Data base.

**7. Test Approach**

**7.1 Functional Testing**

Validate all functions in the user module against the requirements.

Use both positive and negative test cases to ensure robustness.

**7.2 User Interface Testing**

Ensure that the interface elements are correctly displayed and functional.

Verify usability and accessibility of the interface.

**7.3 Security Testing**

Verify access controls and data security measures.

Ensure unauthorized users cannot access restricted information.

**7.4 Compatibility Testing**

Test the user module on different browsers (Chrome, Firefox, Safari, Edge).

Ensure compatibility with different operating systems (Windows, macOS, Linux).

**8. Test Environment**

- Test Servers: Dedicated test servers for the ShoppingList application.

- Browsers: Latest versions of Chrome, Firefox, Safari, Edge.

- Devices: Desktops, Laptops, Tablets.

- Operating Systems: Windows 10, macOS, Linux distributions.

**9. Test Schedule**

- Test Planning: 2 weeks

- Test Case Design: 3 weeks

- Test Execution: 4 weeks

- Bug Reporting and Fixing: Ongoing during test execution

- Final Testing and Sign-off: 1 week

**10. Test Deliverables**

- Test Plan Document

- Test Cases

- Test Scripts (if applicable)

- Test Reports

- Defect Reports

- Final Test Summary Report

**11. Responsibilities**

- Test Manager: Oversee the entire testing process.

- Test Engineers: Execute test cases and report defects.

- Developers: Fix reported defects.

- Project Manager: Ensure timely delivery of the project.

**12. Entry and Exit Criteria**

**Entry Criteria**

- All necessary documentation is available.

- Test environment is set up and ready.

- Application build is stable and deployed.

**Exit Criteria**

- All planned tests have been executed.

- All critical defects have been resolved.

- Test summary report is completed and reviewed.

**13. Risk and Contingencies**

- Delays in receiving stable builds – Ensure early communication with the development team.

- Inadequate test environment setup – Allocate additional time and resources for setup.

**14. Approval**

- Test Manager: [Name]

- Project Manager: [Name]

- QA Lead: [Name]

- Development Lead: [Name]