School of Engineering and Applied Science

Winter 2021 Semester

Software Engineering(SE)

Test Plan Document

Personal Finance Manager

Guided By :
Prof. Khushru Doctor

Team -14
Members :
Priyanshi Shah(AU1841009)
Varshil Shah(AU1841095)
Meet Kadiya (AU1841099)
Kahaan Patel(AU1841110)
Vidit Vaywala (AU1841128)
Sharvil Patel (AU1841134)
Hemil Shah(AU1841135)
Yash A Patel(AU1841141)

Contents

1.Test Plan Identifier

- **Unique Test plan Identifier is-**
  'Test plan for Personal Finance Manager TP_1'

2. References

The list of documents referenced for this test plan are:

- Project Plan
- Software Requirements Specifications
- Design Document

3. Project Background

- Our application revolves around creating a user-friendly interface allowing the users to manage their personal finances more efficiently, tracking their incomes and expenses, getting a detailed analysis of their income and expenses and much more. This web application should be free and accessible from any browser on desktop. This software is a way through which a user can easily get access to his/her income and expense reports and also analyze his/her personal finance requirements. A user can add and track his/her income and expenditure and get various insights about it. A user will get various options to add expenses (e.g. food, entertainment, etc.) and detailed analysis and graphs. A user will get a detailed statement about his income and expenditure with various filters (e.g. daily, weekly, monthly, only food expenses, etc.). A user can set savings goals for the future (e.g. wedding, to buy a car, etc.). Through this interface, a user can decide how much he wants to save and in what timeframe he wants to achieve the goal and also track how his "goal-progress" is going. This system should be user appropriate, easy to use, provide easy recovery of errors and have an overall end user high subjective satisfaction.

4. Introduction

- The test plan document for the Personal Finance Manager project's main function is to explain the research specifics of the personal finance manager's use cases. The aim of the test plan is to identify the different testing methods and techniques that will be used during the project's testing life cycle. The reach, strategy, tools, deliverables, setting, and schedule of all testing activities for the Personal Finance Manager project are all outlined in this Test Plan. Test Plan helps us determine the effort needed to validate the quality of the application under test. This document will cover the various requirements that will be applied to the listed application's device, integration, and system testing. The Personal Finance Manager project will be used to design, produce, and evaluate these reports. We will use the same test documentation requirements in the testing process.

5. Purpose

The main purpose of the test plan for the Personal Finance Manager is as follows:

- To specify the testing criteria and the features of the Personal Finance Manager that is to be tested.
- To conduct the resource planning and determine all the resources that will be needed for the testing of the Personal finance manager.
- To define the pass/fail criteria for each item that will be tested.
- To define the testing objectives. The objective of the testing is finding as many software defects as possible; ensure that the software under test is **bug free** before release.
- To make a note of the testing techniques that will be used to test the personal finance manager.

6. Scope

6.1 Functions to be tested:

Module	Applicable Roles	Description
Sign up	User	A user should be able to sign up to the platform. Necessary details of the user should be verified including his/her email mobile number, password length, etc.
Login	User	The user will be able to login to the system by entering the correct details. Appropriate error messages will be shown for incorrect login attempts.
Dashboard	User	The user will be able to view the overview of his/her income, expense, goals, and bills on the dashboard page.
Add Income	User	The user should be able to add income by specifying the details. Appropriate error messages should be shown if the details are invalid
View Income Statement	User	The user will be able to view his/her income statement and will be able to filter those statements. The user will also be able to see detailed charts for analysis of income.
Add Expense	User	The user should be able to add expenses by specifying the details. Appropriate error messages should be shown if the details are invalid

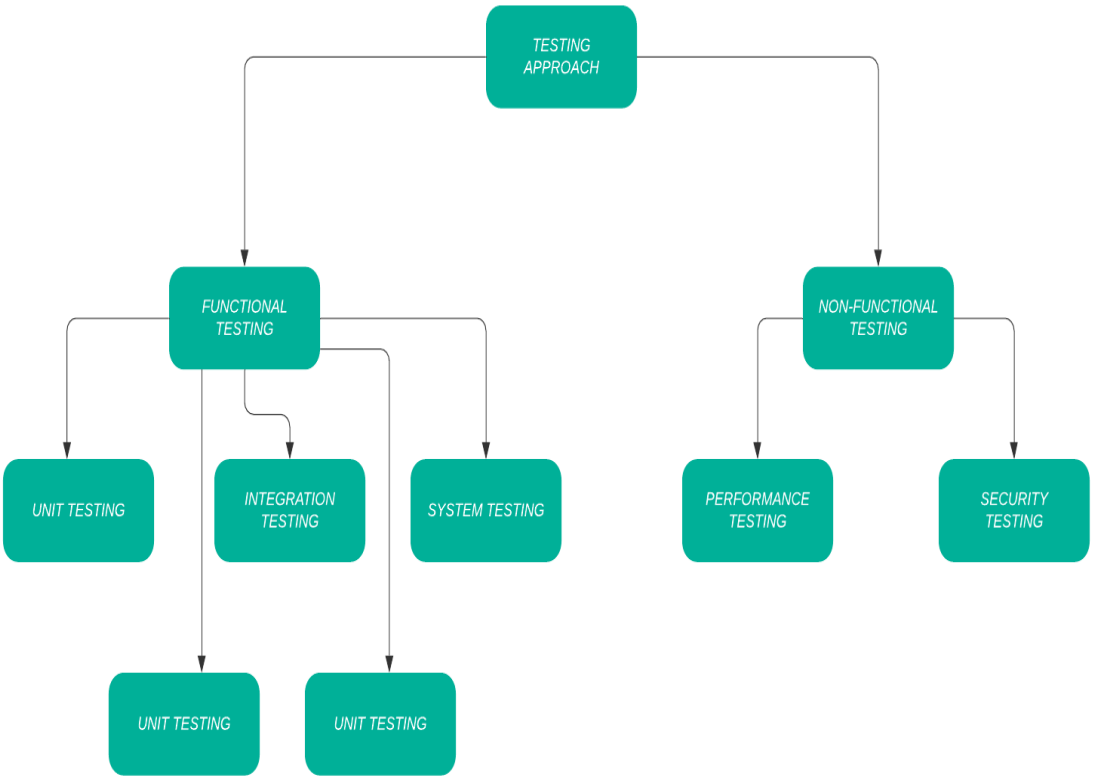
View Expense Statement	User	The user will be able to view his/her expense statement and will be able to filter those statements. The user will also be able to see detailed charts for analysis of expenses.
Add goal	User	A user should be able to create goals for future and add it to the database. He/she has to input various parameters like goal length, amount to save, reason (goal name), goal description etc.
View Edit and Delete Goals	User	A user should be able to view his/her current goals, as well as edit the goal if required. If the user feels that a goal has become irrelevant for him now then he/she can even delete it.
Add Bills	User	A user should be able to add bills and add it to the database. He/she has to input various parameters like bill amount, due date, bill name, bill description etc. Hence the system will give the user an option to track his/her bills in an efficient manner.
View and Delete Bills	User	A user should be able to view his/her current bill. If the user feels that a bill has become irrelevant for him or he has already paid it, then he/she can even delete it.

6.2 Functions Not to be tested:

- Google OAuth Login will not be tested as the functionality is validated and already tested by google services and is not provided by our system.
- Material-UI components work in isolation. They are self-supporting, and will only inject the styles they need to display. They don't rely on any global style-sheets such as normalize.css.
- React Router is the standard routing library for React. React Router keeps your UI in sync with the URL. It has a simple API with powerful features like lazy code loading, dynamic route matching, and location transition handling built right in. Make the URL your first thought, not an after-thought.
- react-router-dom: It contains the DOM bindings for React Router. In other words, the router components for websites react-router-dom: It contains the DOM bindings for React Router. In other words, the router components for websites.
- So these four modules are used directly from the standard libraries or authorized sources which are already tested and verified. That’s why we don’t have to test them again.

7. Testing Approach

- A test method is a project's test plan that specifies how testing will be carried out. Test cases will be created during exploratory testing. All test types are determined in Test Strategy. The project is using an agile approach, more specifically a Scrumban approach, with weekly iterations. At the end of each week the work completed during that week will be reviewed and tested. This section of the test plan describes the overall approach for testing the Personal Finance Manager project. The method used to test the Personal Finance Manager ensures that all of the project's main features are thoroughly reviewed. The research will take place on the Personal Finance Manager when logged in as a User and using the device further.



8. Entry and Exit Criteria

8.1 Entry Criteria

“Specific requirements or ongoing activities that must be present before a process can begin” can be specified as entry criteria for testing. During each testing process, the Software Testing Life Cycle (STLC) determines the entry requirements that must be met. It also specifies the time period or estimated lead time for the entry requirements item to be made available to the method. The requirements needed to be fulfilled for the entry criteria from the testing phase include:

- Availability of code that can be tested.
- A Testing Plan
- The readiness of test cases
- Setting up of test environment with all the necessary resources like tools and devices
- Appropriate Test Environment with all the necessary resources like tools and devices
- Executing the primary functional flows successfully by leveraging various test inputs

8.2 Exit Criteria

Exit requirements in testing are often thought of as a single document that marks the end of a life cycle process. It is described as "the particular requirements or ongoing activities that must be met before the software testing life cycle can be completed." At each testing process, the STLC determines the exit requirements are required.” The exit criteria will help you classify intermediate deliverables and monitor them as separate events.

The following exit criteria should be considered for completion of a testing phase:

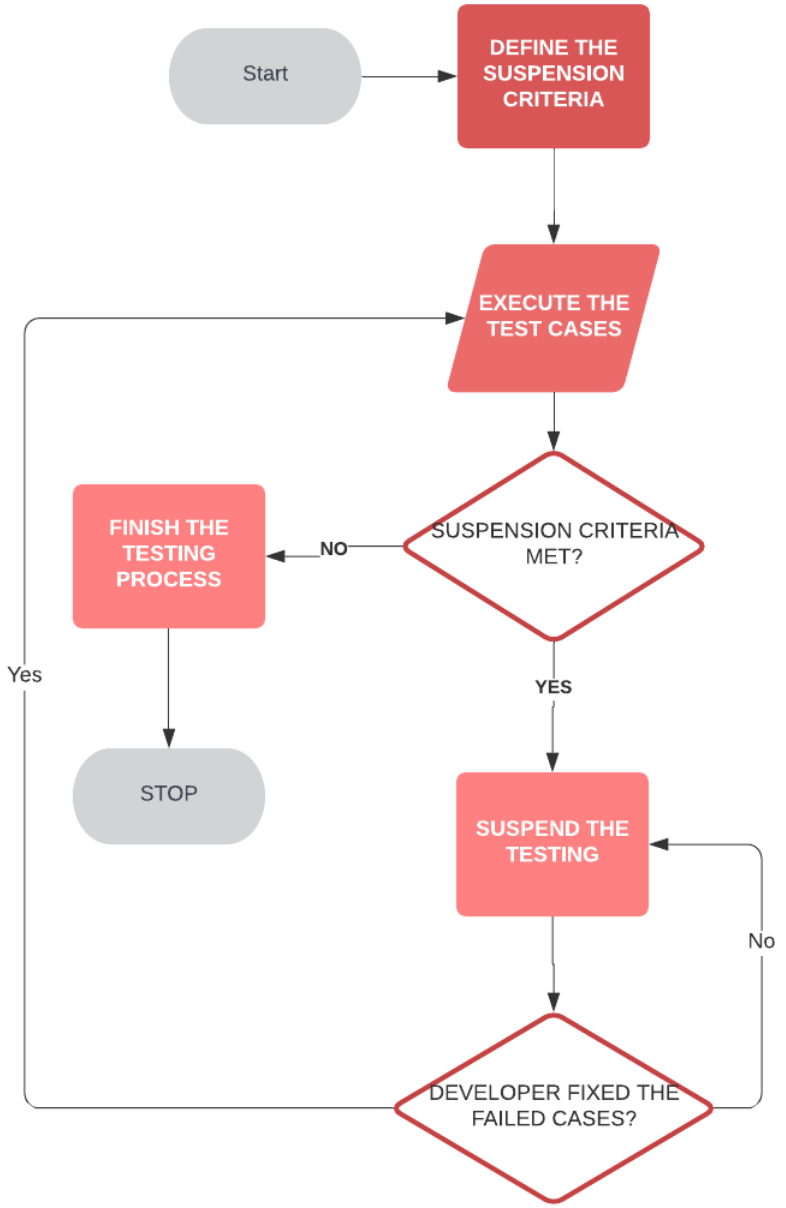
- Ensuring all important test cases are passed
- Obtaining complete functional coverage
- Test summary report generated
- Test Logs generated
- Re-testing and closing all the high-priority defects to execute corresponding Regression scenarios successfully
- Fixing all the ‘Showstopper defects’ or ‘Blockers’ and ensuring that none of the identified Critical/Severity 1 defects are in Open Status

9. Suspension Criteria & Resumption Requirements

9.1 Suspension Criteria

Define the test's vital suspension conditions. The active test period will be suspended until the suspension conditions are resolved if the suspension criteria are met during testing.

- If the code contains many serious defects then the test should be suspended.
- Suspend the test if more than 40% of the test cases fail.
- Suspend the test if assigned resources are not available when needed by the test team.



9.2 Resumption Criteria

Resumption means to resume the testing process that has been halted or suspended. The resumption of the testing process will take place when all the failed test cases are corrected.

10. Test Deliverables

The following test documentation will be produced:

- **Test Plan** - This document deals with what needs to be done in UAT.
- **Test Cases** – a spreadsheet containing all the test cases and their necessary details as well as the output obtained.
- **Unit Test Cases** – The test cases of the unit testing method.
- **Integration Test Cases -** The test cases of the integration testing method.
- **Regression Test Cases –** the test cases of the regression testing method
- **System Test Cases –** the test cases of the system testing method
- **Test Summary Report** - Summary of the testing process carried out.

11. Testing Strategy

11.1 Test Process

The testing process outlines the testing process to be applied and can be considered to have four steps: Develop Tests, Prepare to Test, Run Tests and Review Test Results. These four steps are controlled by Plan Testing and Change Management.

11.1.1 Understanding Requirements:

- **Requirement specifications** taken from stakeholders
- **Understanding of requirements:** Thoroughly going through the requirements and checking out the inconsistencies between the requirements and actual product during testing.

11.1.2 Develop Tests

- **Derive Acceptance Criteria** - The collection of questions to be asked about the device to see if it meets the capability required is prepared after the previous two activities are underway or completed.
- **Construct Test Cases** – proper test cases need to be created which covers all the aspects of the system that requires proper testing.

11.1.3 Preparing Test Matrix

- Preparing a test matrix which maps test cases to respective requirements.
- This will ensure the coverage for requirements.

11.1.4 Reviewing test cases and matrix

This is a key quality process of checking all documentation produced during the development of the system.

- The tester will perform peer review for test cases and test matrix.
- The reviewer will include any feedback or recommendations on test cases and test coverage to the Test Case Author.
- Suggestions or improvements will be re-worked by author and will be sent for approval
- Re-worked improvements will be reviewed and approved by reviewer

11.1.5 Prepare to Test

- **Preparing the environment to run the tests** - Assuring that the necessary personnel, systems, hardware, and software are in place to allow for testing.
- **Preparing Test Data** - Building the data files that are required to run the test cases and record the response.

11.1.6 Run Tests

- **Running the tests** involves using the input and expected results from the Test Cases and applying the Test Scripts and other elements of the Test Procedure to run them.

  Recording the results entails writing down the tasks that were completed and in what order in the Test Log, as well as the events that occurred during the test. Any instances where the actual results differ from the predicted results are recorded in an Incident Report. At this point, the severity of the incident is also determined.

11.1.7 Review Test Results

After the experiments are done, the system's acceptability is determined. Checking how many unresolved Incidents there are and their seriousness is an easy process. However, this is insufficient since a simple count of incidents provides little insight into their effect on the organization's goals

for the system. It is much preferable to have a defective system that provides capability to an organization than to have a flawless system that does not. Therefore the test results need to be checked and traced to see what effect they have on:

- Scenarios,
- Requirements and their
- Business or System Impact

11.2 System Resources Required

Serial Number	Resource	Description
1	Server	The application runs on a server and hence the server should be running smoothly.
2	Network	The application uses internet to function and hence it should be running properly
3	Computer / PC	The application would be run on PC on any browser like google chrome, mozilla firefox, etc.

11.3 Testing Types

Unit testing

Test Objective	It concentrates on the tiniest aspect of software development. We test a single unit or a group of interconnected units in this process.
Technique	Manual
Completion Criteria	If a module passes all the Test Cases written for it, then it has Passed the test.
Example	Checking if a function is working fine, checking incorrect initialization, etc.

Integration testing

Test Objective	The goal is to take unit-tested components and use them to create a program structure that is determined by design. We make sure that all components are integrated without any errors.
Technique	Manual
Completion Criteria	After integrating the modules, if they work without introducing any errors it has passed the test.
Example	Black-box testing, White-box testing, etc.

Regression testing

Test Objective	Any time a new module is introduced, the software is updated. This method of testing ensures that the whole component functions properly even after it has been added to the entire program.
Technique	Manual
Completion Criteria	The addition of a new module to the system does not introduce any new errors and the system works properly.
Example	When we combine the income module with login module, both should work properly.

Smoke testing

Test Objective	This test is done to make sure that software under testing is ready or stable for further testing. This testing covers most of the major functions.
Technique	Manual
Completion Criteria	When two modules are combined, the modifications that go along with it are tested non-exhaustively to ensure that no new errors are added, i.e. that the most critical functions work.
Example	Before testing the income module, we need to make sure that all the functions of the login module are working properly

Security testing

Test Objective	A form of software testing that aims to find system flaws and ensure that the system's data and resources are safe from intruders.
Technique	Manual
Completion Criteria	The system ensures protection from hackers and data stealers.

11.4 Resource and Environment Needs

➔ **Testing Tools**

Process	**Tool**
Creating test cases	Microsoft Excel
Maintaining test case records	Microsoft Excel
Executing the test cases	Manual – in the application
Test Case reporting	Microsoft Excel

➔ **Test Environment**

- Windows 10: Chrome, Firefox
- Linux: Ubuntu

11.5 Testing Schedule

Task Name	Date	Comments
Review Requirements documents	03/05/2020	Reviewing the requirements gathered.
Planning the test schedule	03/05/2020	Deciding the approach to be followed during the entire testing phase.
Creating the test specifications	04/05/2020	Creating the test specifications in a spreadsheet with a proper format
Performing test executions	04/05/2020	Executing the test cases and testing all the components of our application
Reviewing the test results	06/05/2020	Reviewing the test results to determine number of faults and bugs found
Preparing test report in the spreadsheet	06/05/2020	Preparing the test report in a proper format as to which test cases failed and which passed
Resolving the final errors	07/05/2020	Resolving the errors found in the earlier phase.
Testing the resolved application again	07/05/2020	Testing the application again to make sure that the errors are resolved.

12. Risk, Assumptions and Dependencies

12.1 Risks

No.	Description	Status	Impact (I)	Probability (P)	Severity (I*P)	Mitigation Type	Detail of mitigating action to be taken
1	Schedule risk	Closed	4	1	Medium	Avoidance	The schedule of the project gets delayed
2	Continuously changing requirements	Work in Progress	3	5	High	Acceptance	Continuously adapting to the changing requirements
3	Lack of personal resources when test--ing is to begin	Closed	2	2	Medium	Avoidance	Test leader ensures the constant supply of resources for testing
4	Developing features which are not required	Closed	5	3	High	Reduction	Continous feedback to be provided in order to mitigate this problem.
5	Complex implementation of project	Closed	2	3	Medium	Avoidance	The implementation of project is very complex

6	Delays in training the application	Closed	3	4	High	Reduction	Developers in team communicate and find ways to modularize work for faster delivery

12.2 Assumptions

Number	Description	Reason for Assumption	Action to Validate	Impact if Assumption is Incorrect
1	Our user has an active internet connection	The application can also be used with the help of internet	The system is hosted on a server	The user won't be able to access the application
2	User knows the URL of the website	The user needs to find the website in order to register.	The user is able to navigate	The user will not be able to register successfully
3	Sufficient RAM is available for the application to run	The application will crash if sufficient memory isn't available	The user has the needed RAM	The application will crash if sufficient memory isn't available
4	The database and the server are running properly.	The application deals with storage and retrieval of data and hence the database needs to be active and running all the time.	The user will be able to perform tasks in the application	The user will not be able to perform tasks on the application
5	The incomes and expenses entered by the user are correct	There is no facility to connect bank account with the application and hence the user needs to ensure to input the correct details	The user enters his/her correct records and gets proper analysis	The user enters incorrect records and gets improper analysis

12.3 Dependencies

Number	Description	Status	Priority
1	The system depends on the constant availability of Internet	Closed	Medium
2	The system is dependent on the PostgreSQL database in order to function properly	Closed	Medium

13. Summary

- This paper is a descriptive report that explains the research plan, goals, schedule, estimation, deliverables, and resources. The Evaluation Plan assists us in determining the amount of effort required to verify the consistency of the application being tested. The test plan is a blueprint for conducting software testing activities as a fixed process that the test manager closely monitors and controls. It aids people outside of the research team, such as developers, business managers, and consumers, in comprehending the nuances of testing.
- **Test plans** are essential in the development of software as they outline what **testing** needs doing to ensure the software is up to standard and is working exactly how it should. Test plan document also helps us to review testing strategies and select the best ones for the project. We have outlined all the testing strategies that we are going to use, the testing plan, the testing schedule and the testing deliverables for our Personal finance manager in this test plan document.