



**UNIVERSITY
OF MALAYA**



Software Testing (WIF3001)

Semester 2

Session 2021/2022

Group Assignment Report

Lecturer: Dr. Siti Hafizah Binti Ab Hamid

No.	Name	Responsible for Feature	Matric No.
1.	Kisyok Indran A/L Veerasamy (Leader)	- F003 View Daily Expenses - F006 Manage Loan	17203431/1
2.	Muhammad Irfan bin Nasruddin	- F002 View Monthly Expenses - F005 Filter Expenses	17206532/1
3.	Muhammad Hafiz bin Baharudin	- F007 Manage Savings - F011 Edit Currency Setting	17203185/1
4.	Fahad Aqeel Thakur	- F001 Add New Expense - F004 View Overall Stats	17205336/1

* Every team member conducted tests and reported the result in Test Design Specification, Test Case Specification, Test Procedure Specification, Test Log and Test Incident Report for their own features.

* A detailed responsibilities can be found in section 3.3 of Test Management.

Table of Contents

Table of Contents.....	1
Software Brief Description	5
1.0 Project Description	6
2.0 Project Link	12
3.0 Hardware Requirements	12
4.0 Product Features	13
5.0 Actors/Users Characteristics	14
6.0 Functional Requirements	14
6.1 Use Case Diagram	14
6.2 Use Case Description	15
6.3 System States.....	26
7. Non-Functional Requirements	28
Test Item Transmittal Report.....	29
1.0 Transmitted Items	30
2.0 Location	30
3.0 Status.....	31
3.1 Description.....	31
3.2 Deviation.....	31
Test Plan	32
1.0 Introduction	33
1.1 Purpose	33
1.2 Scope	33
1.3 References	33
2.0 Details of Test Plan.....	34
2.1 Test Items	34
2.2 Test Traceability Matrix.....	34
2.3 Features to be Tested.....	34
2.4 Features Not to be Tested	35
2.5 Test Approach or Test Strategy	35
2.6 Item Pass/Fail Criteria	35
2.7 Suspension Criteria and Resumption Requirements	36
2.8 Test Deliveries	36

2.9 Entry Criteria.....	36
2.10 Exit Criteria	37
3.0 Test Management	38
3.1 Planned Tasks and Activities	38
3.2 Environment & Infrastructure	39
3.3 Responsibilities and Authority.....	40
3.4 Resources and Allocation	42
3.5 Training.....	42
3.6 Schedules, Estimates and Costs	43
3.7 Risk & Contingency	44
4.0 General	46
4.1 Metrics	46
4.2 Glossary.....	46
4.3 Incident Classification	47
Test Design Specification.....	48
1.0 Introduction	49
1.1 Purpose	49
1.2 Scope	49
1.3 References	49
2.0 Test Design.....	50
2.1 Features to be Tested.....	50
2.2 Approach Refinements	51
2.3 Test Identification.....	116
2.4 Feature Pass/Fail Criteria	128
2.5 Test Deliverables	128
Test Case Specification	129
1.0 Introduction	130
1.1 Purpose	130
1.2 References	130
2.0 Test Cases.....	130
2.1 Environment.....	130
2.2 Test Case	131
Test Procedure Specification.....	155

1.0 Introduction	156
1.1 Purpose	156
1.2 Scope	156
1.3 References	157
2.0 Test Procedures.....	158
2.1 Create New Expense Test Procedure	158
2.2 View Monthly Expenses Test Procedure	160
2.3 View Daily Expenses Test Procedure	161
2.4 View Overall Stats Test Procedure	162
2.5 Filter Expenses Test Procedure.....	163
2.6 Manage Loan Test Procedure	164
2.7 Manage Savings Test Procedure.....	166
2.8 Edit Currency Setting Test Procedure	168
Test Log	169
Test Incident Report.....	178
Document Control	179
Version History.....	179
Distribution List	179
Test Incident Report.....	180
Static Testing	220
GUI Testing.....	222
Error Guessing.....	230
Test Summary Report	238
Test Completion Report.....	243
Version History.....	244
Distribution List	244
1.0 Summary of Testing Performed	245
2.0 Deviations from Planned Testing	246
3.0 Test Completion Evaluation	246
4.0 Factor that Blocked Progress.....	246
5.0 Test Measure	247
6.0 New / Changed / Residual Risk	247
7.0 Test Deliverables	247

8.0 Reusable Test Assets	247
9.0 Lessons Learnt	248

Software Brief Description

Expense Manager (EM)

Independent Verification and Validation

Version: 1.0.0

Date: 17/06/2022

1.0 Project Description

Expense Manager is an expense tracking app made using React JS front-end framework and Firebase back-end. It is a progressive Web application which means it can be used across multiple devices provided there is an internet connection. It is developed by Yogesh Prasanna who is a Bangalore, India based front-end engineer. It has received 41 stars on GitHub as of 25 April 2022.

Expense Manager comprises of wide range of features such as expenditure tracker, filter expenditure records, loan manager, savings manager, UI feature settings and currency settings. Figure 1.1 shows the home page of Expense Manager that includes summary of expenses and list of expenses in a data table. The “+” button allows the user to add in new expenses. The navigation bar shows all the pages offered by Expense Manager.

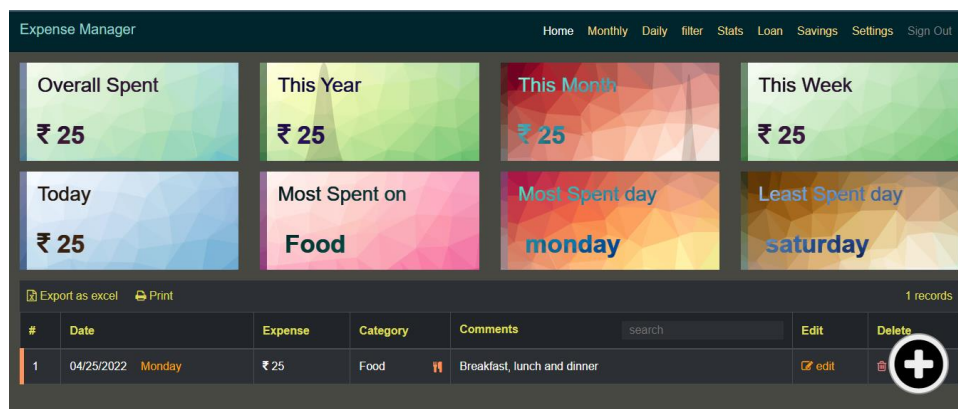


Figure 1.1 Home Page

Monthly Page shows a peek of the monthly spendings as shown in Figure 1.2. It has a calendar that shows the total of spendings on each day and an indication if the spendings is under monthly limit. User is allowed to change this limit in the settings page. It has a linear graph to depict the spendings over the days in the chosen month and a doughnut graph to depict the spendings according to categories. User is allowed to edit or delete the expense in the data table and is allowed to download the data in excel file. Figure 1.3 shows Daily Page which is almost similar to Monthly Page except that it is used to show expenses of a chosen date.



Figure 1.2 Monthly Page



Figure 1.3 Daily Page

Figure 1.4 illustrates the Filter Page that enables the user to filter the expenses according to date range, expense range and category. The filtered data will be displayed in a data table which a user can edit, delete or export as an Excel file. The user is also allowed to print the page.

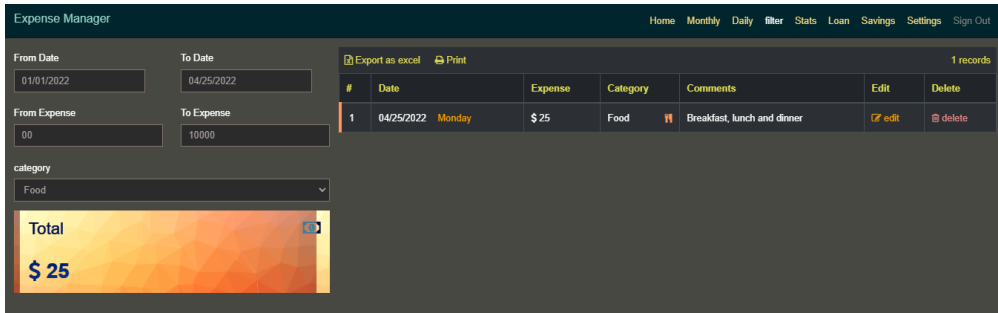


Figure 1.4 Filter Page

The whole picture of expenditure can be viewed in Stats Page as shown in Figure 1.5. It shows the expenditure according to categories in a doughnut chart and according to months in a bar chart. The user is allowed to visualize data in the charts according to the chosen year.

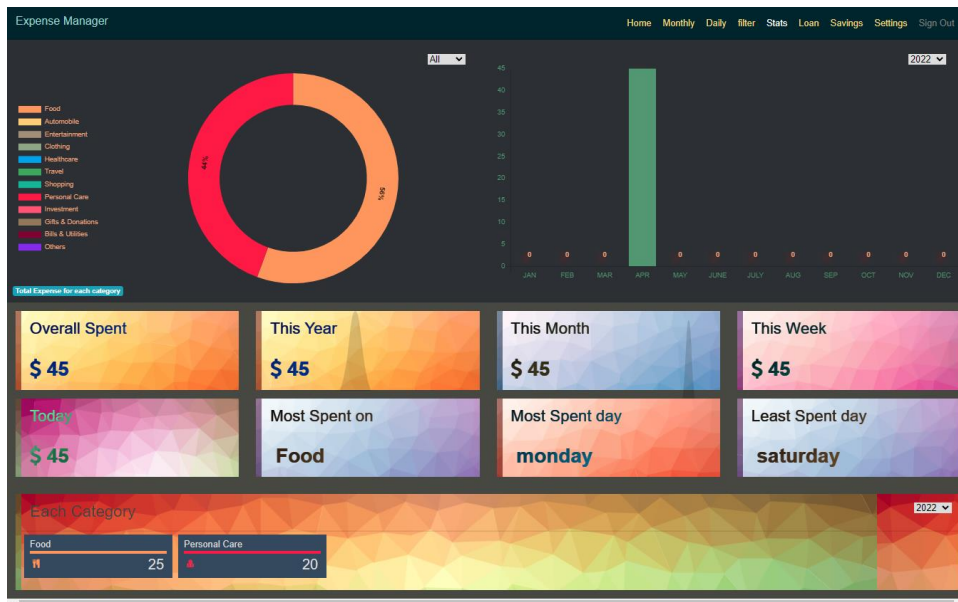


Figure 1.5 Stats Page

Loan management is another feature of Expense Manager that can be managed via Loan Page as depicted in Figure 1.6. It allows a user to add a loan which could either be given or taken by filling up a form. The total amount that a user needs to repay and get are shown together with the list of all loans. The user is also allowed to edit or delete the loans.

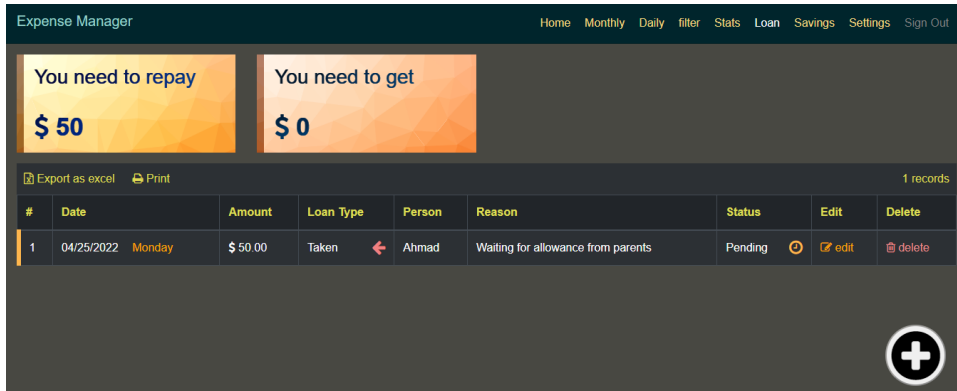


Figure 1.6 Loan Page

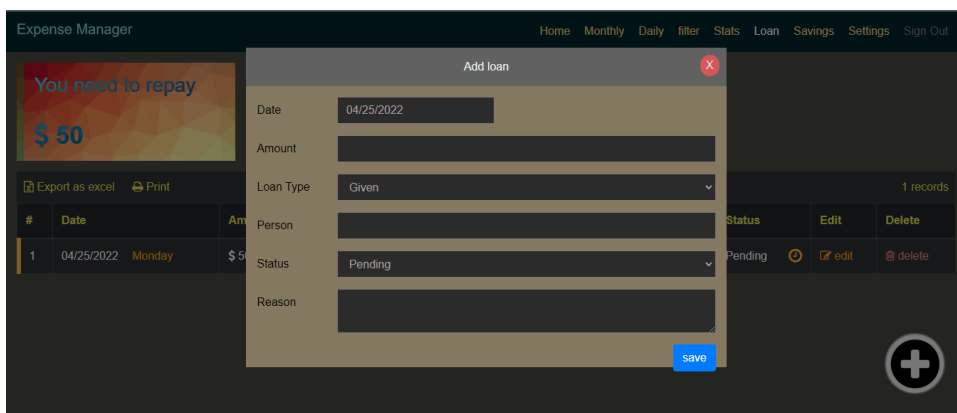


Figure 1.7 Add Loan Form

Besides loan management, Expense Manager also offers savings management via Savings Page as illustrated in Figure 1.8. Once a user fills up the form shown in Figure 1.9, the savings will be added to the Savings Page. Here, the user can add savings for a particular goal or even edit or delete it.

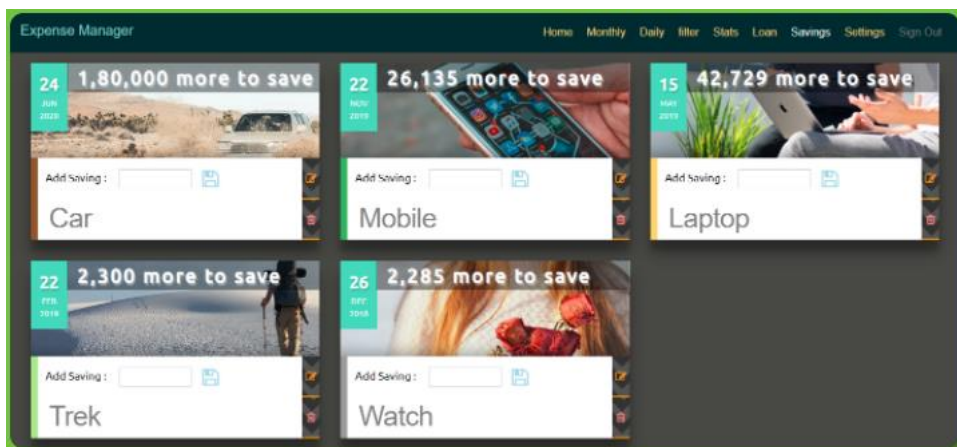


Figure 1.8 Savings Page

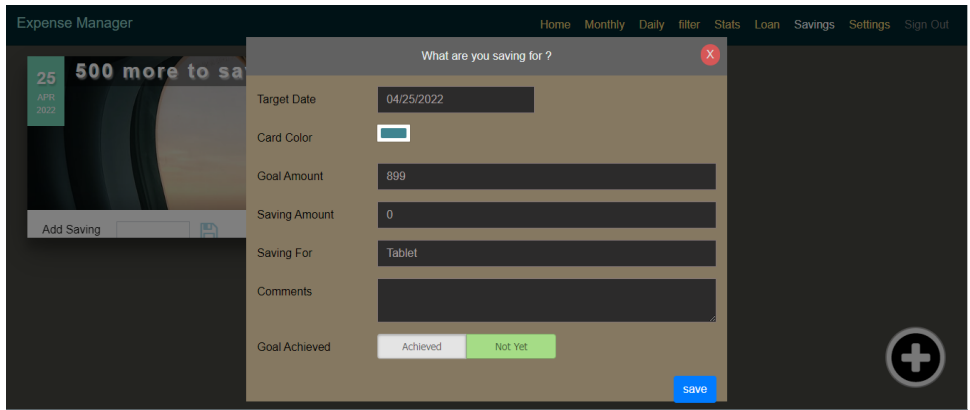


Figure 1.9 Add Savings Form

Unlike most of expense management apps in the market, Expense Manager enables the user to change the settings like UI settings, currency settings and category names as shown in Figure 1.10. UI settings enable the user to change the font type and light mode. The travel mode allows the user to insert the value in the currency of the country they are travelling in and it will be automatically changed to the default currency.

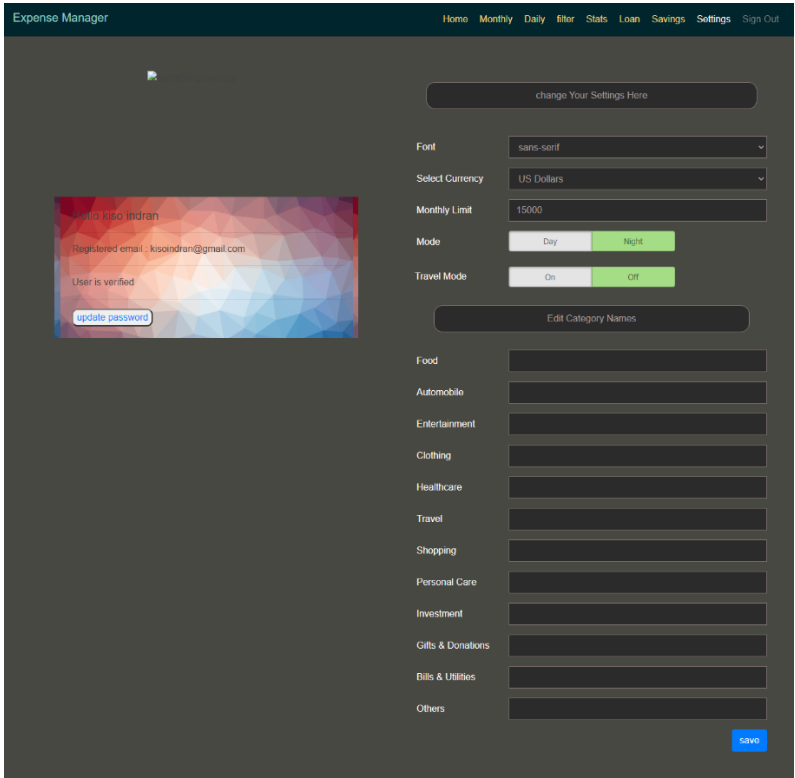


Figure 1.10 Settings Page

Expense Manager is hosted on Firebase, a Backend-as-a-Service platform created by Google. It is used to manage user access and authorization as well as to host real-time database. Figure 1.11 shows the Sign In Page of Expense Manager.

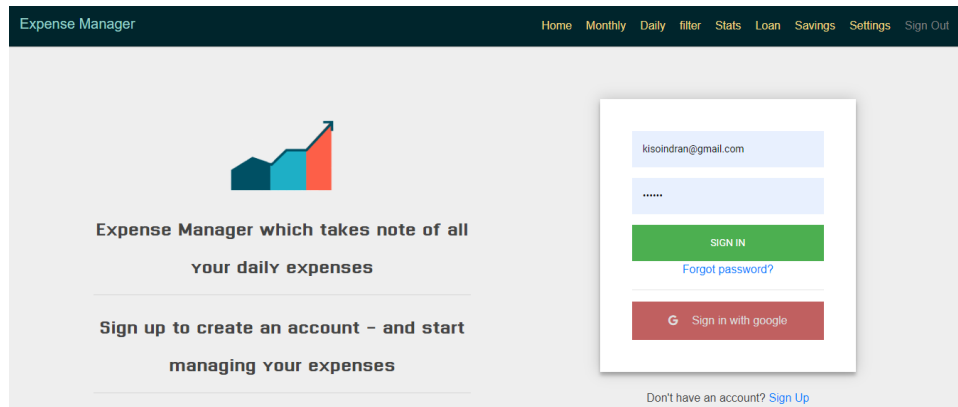


Figure 1.11 Sign In Page

2.0 Project Link

<https://github.com/YogeshPrasanna/expense-manager-react>

3.0 Hardware Requirements

Requirement ID	Item	Description
HR01	Processor: Pentium 4 (or equivalent)	The minimum and recommended processor to run Expense Manager is Pentium 4 or equivalent
HR02	Memory (RAM): Minimum 2 GB, Recommended 4 GB or above	The minimum memory to run Expense Manager is 2 GB and recommended memory to run Expense Manager is 4 GB or above
HR03	Hard Drive: Minimum 10 GB, Recommended 20 GB or more	The minimum available disk space to run Expense Manager with React is 10 GB and recommended is 20 GB or more
HR04	Ethernet connection (LAN) OR a wireless adapter (Wi-Fi)	A connection to the internet to connect Expense Manager to Firebase server

4.0 Product Features

Feature ID	Feature	Description	Actor/Role
F001	Expenditure tracker	To allow user to perform add expenses for a particular date and saves it	User
F002	Statistical view of the monthly expenditure	To allow the user to view a statistical view of their monthly expenditure	User
F003	Statistical view of the daily expenditure	To allow the user to view a statistical view of their daily expenditure	User
F004	Statistical view of the overall expenditure	To allow the user to view a statistical view of their overall expenditure	User
F005	Filter expenditure records function	To allow the user to perform filter expenses records by their specified criteria	User
F006	Loan manager	To allow the user to perform adding, editing and deleting of their loans.	User
F007	Savings manager	To allow the user to perform adding, editing and deleting of their saving's goals	User
F008	Profile settings	To allow the user to perform changes to the user profile details	User
F009	User interface feature settings	To allow the user to perform changes to the user interface	User
F010	Spending category settings	To allow the user to perform changes to spending category names	User
F011	Currency settings	To allow the user to perform changes to the currency used	User

5.0 Actors/Users Characteristics

Actor/Role	Description	Required Knowledge
User	Users are anyone that uses the “Expense Manager” system to manage their personal expense	<ul style="list-style-type: none"> • Basic understanding on how to use browsers • Basic understanding of personal expense concepts and terms

6.0 Functional Requirements

6.1 Use Case Diagram

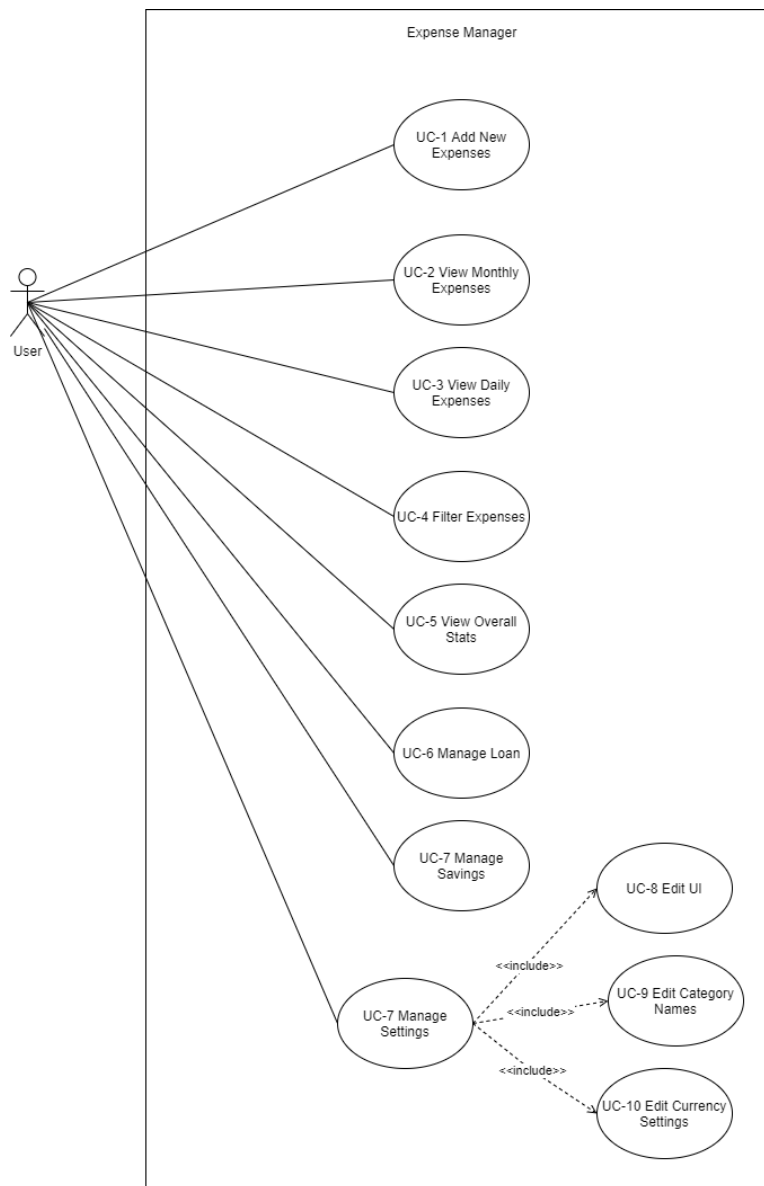


Figure 6.1 Use Case Diagram

6.2 Use Case Description

Use Case Name	Add New Expense
Use Case ID	UC-1
Description	The system shall allow actor to add new expenses for the particular day
Priority	High
Actor(s)	User
Triggering Event	Actor clicks the “+” button in the “Home” page
Pre-condition	Actor inserts expense
Post-condition	The system inserts expense for the user and updates the home page accordingly
Flow of Events	<ol style="list-style-type: none">1. The system displays various key expense stats on the home page2. When the actor clicks on the “+” button, it will bring up a modal to add new expense<ol style="list-style-type: none">2.1. User enters the date for the expense, the amount for the expense, the category and any other comments2.2. The user clicks on the “Save” button in the Add an expense modal3. Home page updates to reflect the added expense

Use Case Name	View Monthly Expenses
Use Case ID	UC-2
Description	The system shall allow actor to view monthly expenses based on the chosen month
Priority	Medium
Actor(s)	User
Triggering Event	Actor clicks the “Monthly” button in the top navigation bar
Pre-condition	Actor inserts date
Post-condition	The system displays expense calendar, total expense, line graph, doughnut chart and data table
Flow of Events	<ol style="list-style-type: none"> 1. The system displays expense calendar, total expense, doughnut chart and data table of current months’ expenses 2. If the actor changes the date, the system will display expense calendar, total expense, doughnut chart and data table of expenses of the month of the chosen date 3. If the actor clicks “edit” button from a specific expense, the system will display the expense in an editable form <ol style="list-style-type: none"> 3.1. The actor edits the new expense in the available fields 3.2. The actor clicks “save” button 3.3. The system verifies the date and expenses amount and updates the expense in the database 4. If the actor clicks ‘delete’ from a specific expense, the system will prompt a confirmation message whether to delete the expense <ol style="list-style-type: none"> 4.1. If the actor clicks the ‘OK’ button, the system will delete expense record in the database 4.2. If the actor clicks the ‘cancel’ button, the system will not delete the expense record in the database

Use Case Name	View Daily Expenses
Use Case ID	UC-3
Description	The system shall allow actor to view daily expenses based on the chosen date.
Priority	Medium
Actor(s)	User
Triggering Event	Actor clicks the "Daily" button in the top navigation bar.
Pre-condition	User inserts date.
Post-condition	The system displays the total expense, doughnut chart and data table.
Flow of Events	<ol style="list-style-type: none"> 1. The system displays total expense, doughnut chart and table of current date's expenses. 2. If the actor changes the date, the system will display total expense, doughnut chart and table of chosen date's expenses. 3. If the actor clicks 'edit' button from a specific expense, the system will display the expense in an editable form. <ol style="list-style-type: none"> 3.1. The actor edits the new expense in the available fields. 3.2. The actor clicks 'save' button. 3.3. The system verifies the date and expenses amount and updates the expense in the database. 4. If the actor clicks 'delete' from a specific expense, the system will prompt a confirmation message whether to delete the expense. <ol style="list-style-type: none"> 4.1. If the actor clicks the 'OK' button, the system will delete the expense record in the database. 4.2. If the actor clicks the 'cancel' button, the system will not delete the expense record in the database.

Use Case Name	Filter Expenses
Use Case ID	UC-4
Description	The system shall allow actor to filter expenses based on specified criteria
Priority	Medium
Actor(s)	User
Triggering Event	Actor clicks the “filter” button in the top navigation bar
Pre-condition	Actor specify filter criteria
Post-condition	The system displays a data table of expenses fulfilling the filter criteria
Flow of Events	<ol style="list-style-type: none"> 1. The system displays data table of expenses of current date 2. If the actor changes the date range, the system will display data table of expenses within the date range 3. If the actor specifies the expense value range, the system will display data table of expenses within the expense value range 4. If the actor specifies the expense category, the system will display data table of expenses of the specified category 5. If the actor clicks “edit” button from a specific expense, the system will display the expense in an editable form <ol style="list-style-type: none"> 5.1. The actor edits the new expense in the available fields 5.2. The actor clicks “save” button 5.3. The system verifies the date and expenses amount and updates the expense in the database 6. If the actor clicks ‘delete’ from a specific expense, the system will prompt a confirmation message whether to delete the expense <ol style="list-style-type: none"> 6.1. If the actor clicks the ‘OK’ button, the system will delete the expense record in the database 6.2. If the actor clicks the ‘cancel’ button, the system will not delete the expense record in the database

Use Case Name	View Overall Stats
Use Case ID	UC-5
Description	The system shall allow actor to view overall expenses statistics
Priority	Medium
Actor(s)	User
Triggering Event	Actor clicks the "Stat" button in the top navigation bar
Pre-condition	There are expenses added into the system by the actor
Post-condition	The actor is shown entire stats about their expenses
Flow of Events	<ol style="list-style-type: none"> 1. The system displays a pie chart for the expenses spent on each category, and a bar chart for the expenses spent for each month 2. The system displays the statistics of overall spending, spending in the current year, current month, current week & today, the most spending on a particular category, the day most money was and the day least money was spent 3. The system displays category wise analysis of the expenditure in that particular category 4. If the actor specifies the year for the pie chart, the system shall display the total expense for each category in the pie chart for that year 5. If the actor changes the year selected in the bar chart, the system shall display the bar chart for the expenditure in that year 6. If the actor changes the year selected on the category analysis for expenditures, the system shall show the category analysis for the expenditures in that year

Use Case Name	Manage Loan
Use Case ID	UC-6
Description	The system shall allow actor to manage (add, edit, delete) loan.
Priority	High
Actor(s)	User
Triggering Event	Actor clicks the “Loan” button in the top navigation bar.
Pre-condition	Loan data exists in the database.
Post-condition	The system displays the total loan to repay, total loan to get and data table.
Flow of Events	<ol style="list-style-type: none"> 1. The system displays total loan to repay, total loan to get and data table. 2. If the actor clicks ‘+’ button, the system will display add loan form. <ol style="list-style-type: none"> 2.1. The actor fills in the available fields. 2.2. The actor clicks ‘save’ button. 2.3. The system verifies the input fields and updates the loan in the database. 3. If the actor clicks ‘edit’ button from a specific loan, the system will display the loan in an editable form. <ol style="list-style-type: none"> 3.1. The actor edits the new loan in the available fields. 3.2. The actor clicks ‘save’ button. 3.3. The system verifies the input fields and updates the loan in the database. 4. If the actor clicks ‘delete’ from a specific loan, the system will prompt a confirmation message whether to delete the loan. <ol style="list-style-type: none"> 4.1. If the actor clicks the ‘OK’ button, the system will delete the loan record in the database. 4.2. If the actor clicks the ‘cancel’ button, the system will not delete the loan record in the database.

Use Case Name	Manage Savings
Use Case ID	UC-7
Description	The system shall allow actor to manage (add, edit, delete) saving's goals.
Priority	High
Actor(s)	User
Triggering Event	Actor clicks the "Savings" button in the top navigation bar.
Pre-condition	Savings data exists in the database
Post-condition	The system displays the saving's goal(s)
Flow of Events	<ol style="list-style-type: none"> 1. The system displays saving's goal(s) of the actor. 2. If the actor clicks '+' button, the system will display add saving's goal form. <ol style="list-style-type: none"> 2.1. The actor fills in the available fields. 2.2. The actor clicks 'save' button. 2.3. The system verifies the input fields and updates the saving's goal in the database. 3. If the actor fills amount of savings in the fields of add savings from a specific saving's goal. <ol style="list-style-type: none"> 3.1. The actor clicks the 'save' icon. 3.2. The system calculates the entered amount and updates the remaining amount of saving's needed to reach saving's goal. 4. If the actor clicks 'edit' icon from a specific saving's goal, the system will display edit saving form. <ol style="list-style-type: none"> 4.1. The actor edits the new saving's goal in the available fields. 4.2. The actor clicks 'save' button. 4.3. The system verifies the input fields and updates the saving's goal in the database. 5. If the actor clicks 'delete' icon from a specific saving's goal, the system will prompt a confirmation message whether to delete the saving's goal. <ol style="list-style-type: none"> 5.1. If the actor clicks 'OK' button, the system will delete the saving's goal record in the database. 5.2. If the actor clicks 'cancel' button, the system will not delete the saving's goal record in the database.

Use Case Name	Manage Settings
Use Case ID	UC-8
Description	The system shall allow actor to manage (edit) user profile.
Priority	High
Actor(s)	User
Triggering Event	Actor clicks the “Settings” button in the top navigation bar.
Pre-condition	Actor’s profile has successfully verified
Post-condition	The system displays the actor’s profile details
Flow of Events	<ol style="list-style-type: none"> 1. The system displays the actor’s profile details 2. If the actor clicks ‘update password’ button, the system will prompt actor to the update password page. <ol style="list-style-type: none"> 2.1. The actor fills in their new password in the enter new password form. 2.2. If the actor clicks ‘UPDATE PASSWORD’ button, the system prompts actor to log in again for firebase precaution. 2.3. The system updates the actor’s user profile password in the database. 2.4. The actor is redirect back to settings page.

Use Case Name	Edit User Interface
Use Case ID	UC-9
Description	The system shall allow actor to edit their preferred user interface
Priority	Low
Actor(s)	User
Triggering Event	Actor clicks the “Settings” button in the top navigation bar.
Pre-condition	Actor’s profile has successfully verified
Post-condition	The system displays the actor’s preferred user interface
Flow of Events	<ol style="list-style-type: none"> 1. The system displays ‘Font’ select field and ‘Day’ and ‘Night’ mode button. 2. If the actor clicks on ‘Font’ select field, the system will display actor with different font choices. <ol style="list-style-type: none"> 2.1. The actor clicks their preferred font’s 3. If the actor clicks on ‘Day’ mode button, the system will display actor that ‘Day’ mode is selected. 4. If the actor clicks on ‘Night’ mode button, the system will display actor the ‘Night’ mode is selected. 5. The actor clicks ‘save’ button, the system will update the actor’s preferred user interface in database. 6. The system will display the updated actor’s preferred user interface.

Use Case Name	Edit Category Names
Use Case ID	UC-10
Description	The system shall allow actor to edit their preferred category names.
Priority	Medium
Actor(s)	User
Triggering Event	Actor clicks the "Settings" button in the top navigation bar.
Pre-condition	Actor's profile has successfully verified
Post-condition	The system displays the actor's preferred category names
Flow of Events	<ol style="list-style-type: none"> 1. The system displays 'Edit Category Names' form. 2. If the actor fills in their preferred category names in one of the category's fields. <ol style="list-style-type: none"> 2.1. The actor clicks 'save' button, the system will update the actor's preferred category names in database. 2.2. The system will display the updated actor's preferred category names.

Use Case Name	Edit Currency Settings
Use Case ID	UC-11
Description	The system shall allow actor to edit their preferred currency settings
Priority	High
Actor(s)	User
Triggering Event	Actor clicks the "Settings" button in the top navigation bar.
Pre-condition	Actor's profile has successfully verified
Post-condition	The system displays the actor's preferred currency settings
Flow of Events	<ol style="list-style-type: none"> 1. The system displays 'Select Currency' select field, 'Monthly Limit' field, and 'Travel Mode' field. 2. If the actor clicks 'Select Currency' select field, the system will display actor with different currency choices. <ol style="list-style-type: none"> 2.1. The actor clicks their preferred currency. 3. The actor fills in the 'Monthly Limit' field 4. If the actor clicks 'On' button on the 'Travel Mode' field, the system displays actor travel mode fields. <ol style="list-style-type: none"> 4.1. The actor clicks their preferred currency from. 5. The actor clicks 'save' button, the system will update the actor's preferred currency settings in database. 6. The system will display the updated actor's preferred currency settings.

6.3 System States

A system is described as stateful if it is designed to remember preceding events or user interactions; the remembered information is called the state of the system.

We will be discussing the system states for the features of expenditure tracker and the filter expenditure record.

6.3.1 System States of Expenditure Tracker Feature

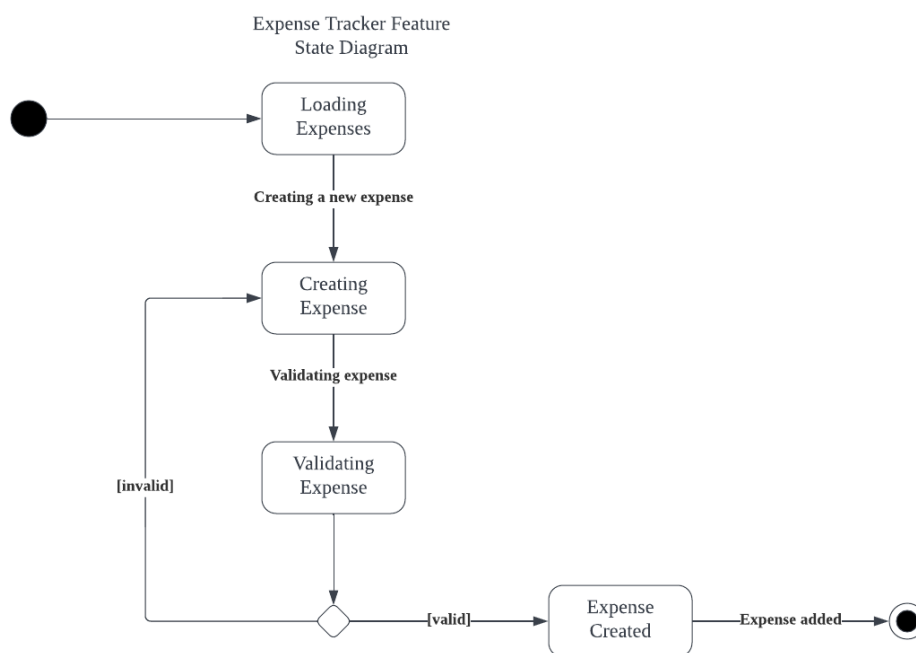


Figure 6.2: State Diagram Expense Tracker Feature

This feature covers the creation and tracking of expenses in the application. There are 4 states that can be identified in this feature. These are as follows:

- **Loading Expenses:** This is the state the application is in when we first load into it. The application loads the expenses data by the user in this state
- **Creating Expense:** This is the state of the application which the user is adding new expense to the system
- **Validating Expense:** This is the state where the input of the expense is validated such as validating the amount of expense. The system will then check if the expense is valid (i.e., if the expense is greater than 0 and if a date has been chosen)
- **Expense Created:** This is the state where the expense is created and stored in the database of the system

6.3.2 System States of Filter Expenditure Record Feature

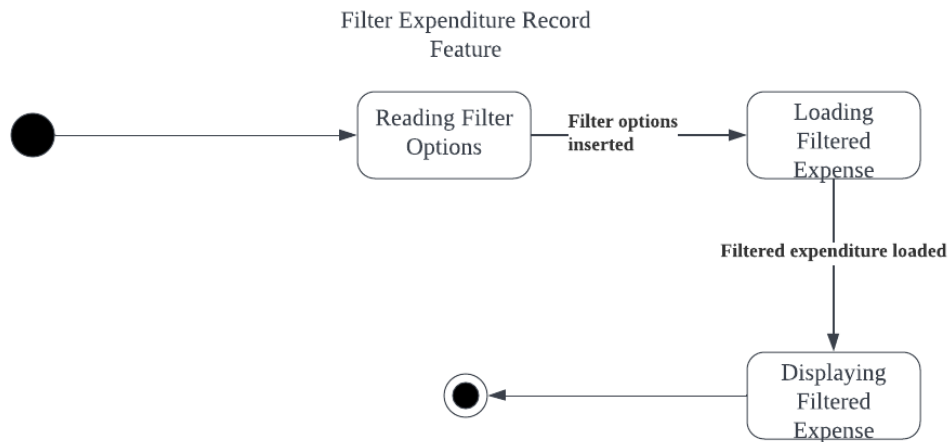


Figure 6.3: State Diagram of Filter Expenditure Feature

This feature covers the filtering of expenditure records based on the inputs provided by the user. There are 3 states to this feature. They are as follows:

- **Reading Filter Options:** This is the state the application is in when the user loads the “Filter” page. The application is waiting for user inputs for the Date, Category & Range options
- **Loading Filtered Expense:** Once the user has input the filter options, the system will load the expenditure record based on it
- **Displaying Filtered Expense:** Once the filtered expenditure record is loaded, the application will display these results to the user

7. Non-Functional Requirements

Requirement ID	Quality	Description	Priority
NFR01	Usability	Level of which a software can be used by intended users to complete specified goals efficaciously, efficiently and sufficiently in a certain context of use.	High
NFR02	Reliability	Level of which a software achieves specified functions under certain conditions for a defined period of time.	Medium
NFR03	Security	Level of which a software protects data such that other software has the correct level of privilege and authorization to access data.	High
NFR04	Portability	Level of efficacy and sufficiency of a software to be transferred from one environment to another environment.	Medium

Test Item Transmittal Report

Expense Manager (EM)

Independent Verification and Validation

Version: 1.0.0

Date: 17/06/2022

1.0 Transmitted Items

The items transmitted are listed in Table 1.0 below:

Version	Name	Reference	People Responsible
1.0	Group Contract	WIF3001-P-Kisyok_GC.pdf	EM Test Team
1.0	Software Brief Description	WIF3001-P-Kisyok_SBD.pdf	EM Test Team
1.0	Expense Manager Source Code	Expense-Manager.zip	EM Test Team

Table 1.0.1 Test Transmittal Items

2.0 Location

The locations of the transmitted items are listed in Table 2.0 below:

Transmittal Item	Item Location	Link to Item (if applicable)
WIF3001-P-Kisyok_GC.pdf	OneDrive	https://1drv.ms/b/s!AI7v6WpXcUJyjHIF47BcY4I_8ni8?e=1n9u8l
WIF3001-P-Kisyok_SBD.pdf	OneDrive	https://1drv.ms/b/s!AI7v6WpXcUJyjQZD6AYuaVneWvHr?e=JpuUr2
Expense-Manager.zip	Expense Manager GitHub	https://github.com/YogeshPrasanna/expense-manager-react

Table 2.0.1 Transmittal Item Location

3.0 Status

3.1 Description

The system under test (SUT), Expense Manager is being tested at system level for the first time after development. There is no previous tests or test document provided for the SUT by the developer in the GitHub.

3.2 Deviation

The deviation of the transmitted items is described below:

3.2.1 From Previous Transmittal

Not applicable

3.2.2 From Test Plan

Not applicable

3.2.3 From Incident Report

Not applicable

Test Plan

Expense Manager (EM)

Independent Verification and Validation

Version: 1.0.0

Date: 17/06/2022

1.0 Introduction

1.1 Purpose

This test plan supports the following objectives:

- i. To detail the activities required to prepare for and conduct the system test.
- ii. To communicate with all responsible parties the tasks that they are to perform and the schedule to be followed in performing the tasks.
- iii. To define the sources of the information used to prepare the plan.
- iv. To define the test tools and environment needed to conduct the system test.

1.2 Scope

The scope of testing only covers black box-functional testing for features developed in the EM from F001 to F007, and F011. The test plan only covers system level test which excludes unit test, static test, integration test, acceptance test, regression test, confirmation test as well as the other types of tests that are not mentioned in this Test Plan. The test plan will solely focus on meeting the requirements of the system and to make sure that the system is fit for purpose.

1.3 References

The test basis for this test design is based on EM Software Brief Description (SBD).

2.0 Details of Test Plan

2.1 Test Items

EM Software Brief Description (SBD)

2.2 Test Traceability Matrix

The following table contains the functions in EM and its traceability

Function ID	Risk Level	Function Description	Source of Function
F001	High	Add new expenses	SBD
F002	Moderate	View monthly expenses	SBD
F003	Moderate	View daily expenses	SBD
F004	Moderate	View overall stats	SBD
F005	Moderate	Filter expenses	SBD
F006	High	Manage loan	SBD
F007	High	Manage savings	SBD
F011	High	Edit Currency Setting	SBD

Table 2.2.1 Test Traceability Matrix

2.3 Features to be Tested

The following table contains the features to be tested based on EM Software Brief Description (SBD). Listed together are the function ID and its corresponding functions and estimated risk level.

Function ID	Functions	Risk Level
F001	Add new expenses	High
F002	View monthly expenses	Moderate
F003	View daily expenses	Moderate
F004	View overall stats	Moderate
F005	Filter expenses	Moderate
F006	Manage loan	High
F007	Manage savings	High
F011	Edit Currency Setting	High

Table 2.3.1 Features to Be Tested

2.4 Features Not to be Tested

Aside from the features listed in Section 2.2, other aspects of the system are not covered in this test plan. This includes:

- i. Operation procedure
This test plan is for system level test, operation procedure is not covered here.
- ii. Network security
This test plan is designed for functional test, security is not covered here.

2.5 Test Approach or Test Strategy

The test on EM is a system level functional test that focuses only on the functional part of the system. Testing will be using risk-based strategy.

This test will also be a black-box test where the test cases are derived using black-box testing techniques such as:

- i. Equivalence partitioning
- ii. Boundary value analysis
- iii. Decision table testing
- iv. State transition testing
- v. Use case testing

2.6 Item Pass/Fail Criteria

The system must satisfy the following criteria in order to pass:

- i. All test cases must be passed.
- ii. 0% calculation error in any function involving expenditures, loans, and savings.
- iii. No deadlock during operation.

2.7 Suspension Criteria and Resumption Requirements

Suspension Criteria:

In case when the Firebase services such as Firestore Database is not available, all tests will be suspended.

Resumption Requirements:

The Firebase services is prepared.

2.8 Test Deliveries

The following documents will be generated by the system test group and will be delivered during the system testing.

Test documentation:

- i. Test Plan
- ii. Test Transmittal Report
- iii. Test Design Specification
- iv. Test Case Specification
- v. Test Procedure Specification
- vi. Test Logs
- vii. Test Incident Reports
- viii. Test Summary Report
- ix. Test Completion Report

2.9 Entry Criteria

The following items are needed before the testing can begin:

- i. Software Brief Description (SBD) for EM 1.0
- ii. Source Code for EM 1.0
- iii. Software Under Test: EM 1.0

2.10 Exit Criteria

The following items are needed before the testing can end:

- All test deliverables in section 2.8 Test Deliverables have been generated.
- Test execution has been completed.
- Open defect at the end of final iteration shall be:
 - Zero fatal defect.
 - Zero major defect.
 - Maximum three minor defects.

3.0 Test Management

This section describes the test activities and tasks for each specified level of progression of the test of EM. It is here to identify the planned activities, responsibilities and authorities, resources, training, schedules, and risk(s) of this test.

3.1 Planned Tasks and Activities

The following table shows the set of tasks necessary to prepare for and perform testing for the EM. It identifies dependencies of other tasks and significant constraints such as test item availability, testing resources availability and deadlines.

No	Task	Activities	Target Date
1	Test Planning and Control	<ol style="list-style-type: none"> 1. Define test project scope and objectives 2. Requirement View 3. Identified risks and scheduling 4. Documenting test plan 5. Technical review 6. Submission and approval of test plan 	09/05/2022
2	Test Analysis and Design	<ol style="list-style-type: none"> 1. Analyze test object 2. Develop test design specification 3. Prepare test case specification 	16/05/2022
3	Test Environment	<ol style="list-style-type: none"> 1. Setup EM 2. Identify and acquired test data 	25/04/2022
4	Test Implementation and Execution	<ol style="list-style-type: none"> 1. Prepare test procedure specification 2. Perform functional test 3. Record test result 4. Analyze test result 	30/05/2022
5	Documenting Test Closure	<ol style="list-style-type: none"> 1. Collect test process data from completed activities to consolidate experience, testware, facts and numbers. 	06/06/2022
6	Test Complete	<ol style="list-style-type: none"> 1. Delivery of product to client 	10/06/2022

Table 3.1.1 Planned Activities and Tasks

3.2 Environment & Infrastructure

This section describes the hardware, software and operating system that are required for this system testing.

3.2.1 Hardware

There are no specific hardware requirements of EM to run it on a web browser.

3.2.2 Software

The software requirements to run EM are any desktop-based web browser with JavaScript enabled. To run EM locally we would need to have Node JS software installed.

3.2.3 Publications

Software Brief Description (SBD) is required to support system testing along with the documentation on the systems GitHub repository.

3.3 Responsibilities and Authority

This section will identify the individuals or groups responsible for managing, designing, preparing, executing, witnessing and checking results of the testing. In addition, it will identify the persons responsible for providing the test items in Section 2.0 and environmental needs identified in Section 3.2.

Name	Role	Responsibilities
Kisyok Indran A/L Veerasamy	Test manager	<ul style="list-style-type: none"> • Deciding on the execution of reviews • Choosing the testing objects and documents to be used • Providing the necessary resources • Determining if the review objectives have been met • Preparing Test Transmittal Report • Preparing Test Design Specification, Test Case Specification, Test Procedure Specification, Test Log and Test Incident Report for F003 View Daily Expenses and F006 Manage Loan • Preparing Test Summary Report
Muhammad Irfan bin Nasruddin	Test moderator	<ul style="list-style-type: none"> • Leading the review and responsible for the planning activities • Mediating between various points of views • Preparing Test Design Specification Template • Preparing Test Design Specification, Test Case Specification, Test Procedure Specification, Test Log and Test Incident Report for F002 View Monthly Expenses and F005 Filter Expenses • Formatting final report
Muhammad Hafiz bin Baharudin	Test reviewer	<ul style="list-style-type: none"> • Ensuring defective parts of the review object are marked • Representing different perspectives and taking part in all examinations • Preparing Test Plan • Preparing Test Design Specification, Test Case Specification, Test Procedure Specification, Test Log and Test Incident Report for F007 Manage Savings and F011 Edit Currency Setting • Preparing Test Completion Report

Fahad Aqeel Thakur	Test scriber	<ul style="list-style-type: none"> • Documenting all issues and problems that were identified • Writing up the discussions and formulating concisely • Preparing Test Plan • Preparing Test Design Specification, Test Case Specification, Test Procedure Specification, Test Log and Test Incident Report for F001 Add New Expense and F004 View Overall Stats • Preparing Test Completion Report
-----------------------	-----------------	---

Table 3.3.1 Responsibilities and Authorities

3.4 Resources and Allocation

This section will describe any addition required resources that are not already documented by other parts of the plan. This is including internal or external resources.

3.4.1 External Resources

Firebase is required to be setup as a Backend-as-a-Service to use the EM system on our local machines. It is needed to be setup along with the Real Time database.

3.4.2 Internal Resources

There are no internal resources required in system testing.

3.5 Training

Training is not required for the testing personnel.

3.6 Schedules, Estimates and Costs

This section will describe the project schedule and task milestones. The schedule will include estimated time that is required for each testing task. Cost is not included for this testing.

Task	Milestone	Days	Duration
Test Planning and Control	Completed Test Plan	5	04/05/2022 - 09/05/2022
Test Analysis and Design	Completed Test Design Specification, Test Case Specification and Test Procedure Specification	7	09/05/2022 - 16/05/2022
Test Environment	Setup test environment	3	22/04/2022 - 25/04/2022
Test Implementation and Execution	Execution of functional test for EM	16	14/05/2022 - 30/05/2022
Documenting Test Closure	Document all results from Test Design Specification, Test Case Specification and Test Procedure Specification. (Producing Test Log)	2	04/06/2022 - 06/06/2022
Test Complete	Compiled everything with backup	1	09/06/2022 - 10/06/2022

Table 3.6.1 Estimated Testing Schedule

3.7 Risk & Contingency

Rating For Likelihood for Each Risk	
1	Rated as Low
2	Rated as Medium
3	Rated as High
4	Rated as Extreme

Table 3.7.1 Likelihood Rating Table

Rating For Impact for Each Risk	
1	Rated as Low
2	Rated as Medium
3	Rated as High
4	Rated as Extreme

Table 3.7.2 Priority Rating Table

Function ID	Risk Description	Likelihood	Impact	Risk Exposure	Risk Treatment Plan	People Responsible	Cost
F001	User is unable to add new expenses	2	5	9	Perform functional test and use at least 3 test techniques	Test Team	-
F002	User is unable to view monthly expenses	2	3	7	Perform functional test and use at least 3 test techniques	Test Team	-
F003	User is unable to view daily expenses	2	3	7	Perform functional test and use at least 3 test techniques	Test Team	-
F004	User is unable to view overall stats	2	3	7	Perform functional test and use at least 3 test techniques	Test Team	-
F005	User is unable to filter expenses	2	2	6	Perform functional test and use at least 3 test techniques	Test Team	-
F006	User is unable to manage loans	3	3	7	Perform functional test and use at least 3 test techniques	Test Team	-
F007	User is unable to manage savings	3	3	7	Perform functional test and use at least 3 test techniques	Test Team	-
F011	User is unable to Edit Currency Setting	3	3	7	Perform functional test and use at least 3 test techniques	Test Team	-

Table 3.7.3 Function List and Risk Ratings

4.0 General

4.1 Metrics

The following metrics will be collected and used to monitor progress:

Metric List	Metric Description	Goals
Percentage of test cases run during test execution	Metric to monitor total number of test cases executed	100%
Actual duration versus plan (variance)	Metric to monitor the project progress compared to the plan	Allow 10% delay
Percentage of test cases passed against total of test cases executed	Metric to monitor total number of test cases executed with passed criteria	All high-risk test cases passed
Percentage of baseline requirement covered against total test cases	Metric to monitor total number of baseline requirement coverage	100%

Table 4.1.1 Metrics To Be Collected and Used to Monitor Progress

4.2 Glossary

Term	Definition
EM	Expense Manager
SBD	Software Brief Description
TDC	Test Design Specification
TCS	Test Case Specification
TPS	Test Procedure Specification
TIR	Test Incident Report
TL	Test Log

Table 4.2.1 Glossary Used

4.3 Incident Classification

Impact
Mission Critical: Application will not function, or system fails
Major: Severe problems but possible to work around
Minor: Does not impact the functionality, or the usability of the system's processes are not based on the requirements or design specifications
Priority
Immediate: Must be fixed as soon as possible
Delayed: System is unstable, but incident must be fixed prior to next level of test
Deferred: Defect can be left in if necessary due to time or costs constraints

Table 4.3.1 Incident Classification

Test Design Specification

Expense Manager (EM)

Independent Verification and Validation

Version: 1.0.0

Date: 17/06/2022

1.0 Introduction

1.1 Purpose

This test design specification supports the following objectives:

- i. To identify the test items covered
- ii. To further refine the test approach from Test Plan
- iii. To define the sources of the information used to prepare the Test Plan
- iv. To identify the general cases of test to be executed

1.2 Scope

This test design specification covers Expenditure Tracker Features developed in Expense Manager (EM_1.0).

1.3 References

The following documents provide the test basis for this test design:

- i. EM Software Brief Description (EM_SBD_2.0)

2.0 Test Design

2.1 Features to be Tested

The following table contains the features to be tested based on EM Software Brief Description for this iteration. Listed together are the function ID and its corresponding functions and estimated risk level.

Function ID	Function	Risk Level
F001	Create New Expense	High
F002	View Monthly Expenses	Moderate
F003	View Daily Expenses	Moderate
F004	View Overall Stats	Moderate
F005	Filter Expenses	Moderate
F006	Manage Loan	High
F007	Manage Savings	High
F011	Edit Currency Setting	High

Table 2.1.1 Features to Be Tested

2.2 Approach Refinements

Test cases for EM will be designed using five techniques which are covered in this section. Techniques are applied based on suitability of techniques according to nature of features.

The techniques that will be applied are as follows:

- i. Equivalence Partitioning
- ii. Boundary Value Analysis
- iii. Decision Table Testing
- iv. State Transition Testing
- v. Use Case Testing

Nature of feature as below:

- F001 Create New Expenses – add expense for a particular date and save
- F002 View Monthly Expenses – expenses can be viewed in monthly context
- F003 View Daily Expenses – expenses can be viewed in daily context
- F004 View Overall Stats – overall expense stats can be viewed
- F005 Filter Expenses – filtered expenses based on specified criteria can be viewed
- F006 Manage Loan – loan can be created, edited, or deleted.
- F007 Manage Savings – savings can be created, edited, or deleted
- F011 Edit Currency Setting – currency settings can be edited

2.2.1 F001 Create New Expenses

Five techniques are identified for designing test for this feature, including:

- i. Equivalence Partitioning
- ii. Boundary Value Analysis
- iii. Decision Table Testing
- iv. State Transition Testing
- v. Use Case Testing

Equivalence partitioning is used to split the expense amount and date into possible partitions for test. Boundary value analysis will test the boundary values of the expense amount and date input. Decision table is used to cover major branches of decision. State transition testing will test for major states of EM.

2.2.1.1 Equivalence Partitioning

Test Condition ID	Test Condition	Test Coverage ID	Test Coverage	Example of Test Data
TCON-01-001	Expense <= 0	TCOV-01-001	Expense < 0 (Invalid)	-5
TCON-01-002	Expense > 8000	TCOV-01-002	Expense > 0 (Invalid)	10000
TCON-01-003	0 < Expense <= 8000	TCOV-01-003	0 < Expense <= 8000 (Valid)	500
TCON-01-004	Date of month <= 0	TCOV-01-004	Date of month <= 0 (Invalid)	Date = 0/06/2022
TCON-01-005	0 < Date of Month <= 30	TCOV-01-005	0 < Date of Month <= 30 (Valid)	Date = 06/06/2022
TCON-01-006	Date of Month > 30	TCOV-01-006	Date of Month > 30 (Invalid)	Date = 33/06/2022
TCON-01-007	Date of month != integer	TCOV-01-007	Date of month != integer (invalid)	Date = "abc"
TCON-01-008	Comment length = 0	TCOV-01-008	Comment length = 0 (invalid)	Comment = ""
TCON-01-009	0 < Comment length <= 300	TCOV-01-009	0 < Comment length <= 300 (valid)	Comment = "Grocery"
TCON-01-010	Comment length > 300	TCOV-01-010	Comment length > 300 (invalid)	Comment = 350 chars
TCON-01-011	Category = "Food"	TCOV-01-011	Category = "Food" (valid)	Category = "Food"

Table 2.2.1.1.1 Create New Expense Equivalence Partitioning Test Condition & Coverage

***Note:** 8000 has been used as the upper limit considering the average monthly salary in Malaysia is RM8000.

We will be using the month of June so the date of month has a max value of 30.

For the max length of the comment, we are assuming that to be 300 based on the text field size on the UI

2.2.1.2 Boundary Value Analysis

Test Condition ID	Test Condition	Test Coverage ID	Test Coverage	Example of Test Data
TCON-01-012	Expense <= 0	TCOV-01-012	Expense <= 0 (Invalid)	0
TCON-01-013	Expense > 8000	TCOV-01-013	Expense > 8000 (Invalid)	8001
TCON-01-014	0 < Expense <= 8000	TCOV-01-014	0 < Expense <= 8000 (Valid)	1
TCON-01-015	0 < Expense <= 8000	TCOV-01-015	0 < Expense <= 8000 (Valid)	2
TCON-01-016	0 < Expense <= 8000	TCOV-01-016	0 < Expense <= 8000 (Valid)	7999
TCON-01-017	0 < Expense <= 8000	TCOV-01-017	0 < Expense <= 8000 (Valid)	8000
TCON-01-018	Date of month <=0	TCOV-01-018	Date of month <= 0 (invalid)	0/06/2022
TCON-01-019	Date of month > 30	TCOV-01-019	Date of month > 30 (invalid)	31/06/2022
TCON-01-020	0 < Date of month <= 30	TCOV-01-020	0 < Date of month <= 30 (valid)	01/06/2022
TCON-01-021	0 < Date of month <= 30	TCOV-01-021	0 < Date of month <= 30 (valid)	02/06/2022
TCON-01-022	0 < Date of month <= 30	TCOV-01-022	0 < Date of month <= 30 (valid)	29/06/2022
TCON-01-023	0 < Date of month <= 30	TCOV-01-023	0 < Date of month <= 30 (valid)	30/06/2022
TCON-01-024	Category = "Food"	TCOV-01-024	Category = "Food"	Category = "Food"
TCON-01-025	Category = "Others"	TCOV-01-025	Category = "Others"	Category = "Others"
TCON-01-026	Comment length = 0	TCOV-01-026	Comment length = 0 (invalid)	Comment = ""
TCON-01-027	Comment length > 300	TCOV-01-027	Comment length > 300 (invalid)	Comment length = 301 chars

TCON-01-028	0 < Comment length <= 300	TCOV-01-028	0 < Comment length <= 300 (valid)	Comment = "a"
TCON-01-029	0 < Comment length <= 300	TCOV-01-029	0 < Comment length <= 300 (valid)	Comment = "ab"
TCON-01-030	0 < Comment length <= 300	TCOV-01-030	0 < Comment length <= 300 (valid)	Comment length = 299 chars
TCON-01-031	0 < Comment length <= 300	TCOV-01-031	0 < Comment length <= 300 (valid)	Comment length = 300 chars

Table 2.2.1.2.1 Create New Expense Boundary Value Analysis Test Condition & Coverage

2.2.1.3 Decision Table Testing

Test Condition ID	Test Condition
TCON-01-032	0 < Expense <= 8000
TCON-01-033	0 < Date of Month <= 30
TCON-01-034	0 < Comment length <= 300
TCON-01-035	Error Dialog Message
TCON-01-036	Create new expense

Table 2.2.1.3.1 Create New Expense Decision Table Test Conditions

Condition	TCOV-01-032	TCOV-01-033	TCOV-01-034	TCOV-01-035	TCOV-01-036
0 < Expense <= 8000	Y	N	N	N	Y
0 < Date of Month <= 30	Y	N	N	Y	N
0 < Comment length <= 300	Y	N	Y	N	N
Output					
Error Dialog Message	N	Y	Y	Y	Y
Create new expense	Y	N	N	N	N

Table 2.2.1.3.2 Create New Expense Decision Table

2.2.1.4 State Transition Testing

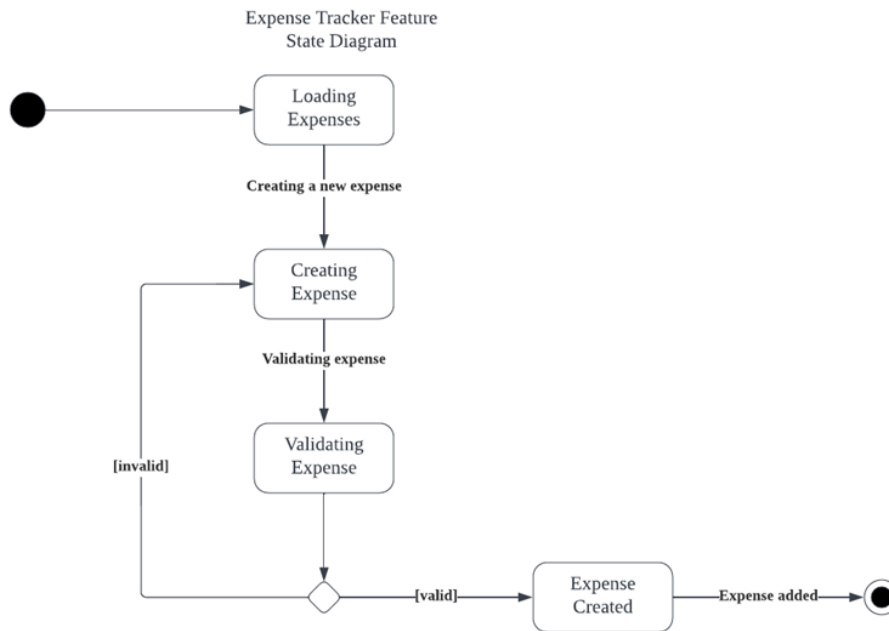


Figure 2.2.1.4.1 Create New Expense State Transition Diagram

The test condition for state transition testing is as following:

- i. TCON-01-037 Create new expense state transition

	Initial	Loading Expenses (S1)	Creating Expense (S2)	Validating Expense (S3)	Expense Created (S4)
Constructor (C)	Loading Expenses (TCOV-01-037)	N/A	N/A	N/A	N/A
Create Expense Button (CE)	N/A	Creating Expense (TCOV-01-038)	?	?	?
Save Button (SB)	N/A	?	Validating Expense (TCOV-01-039)	?	?
Valid Expense (VE)	N/A	?	?	Expense Created (TCOV-01-040)	Final (TCOV-01-042)

Invalid Expense (IE)	N/A	?	?	Creating Expense (TCOV-01-041)	?
-----------------------------	-----	---	---	--------------------------------	---

Table 2.2.1.4.1 Create New Expense State Table

Test Coverage:

Test Coverage ID	Test Coverage
TCOV-01-037	Initial to S1 with C
TCOV-01-038	S1 to S2 with CE
TCOV-01-039	S2 to S3 with SB
TCOV-01-040	S3 to S4 with VE
TCOV-01-041	S3 to S2 with IE
TCOV-01-042	S4 to Final with VE

Table 2.2.1.4.2 Create New Expense State Coverage

2.2.1.5 Use Case Testing

Use Case ID	UC-1	
Use Case	Add New Expense	
Purpose	The system shall allow the actor to add new expenses for that particular day	
Actor	User	
Trigger	Actor clicks on the "+" button on the "Home" page	
Precondition	-	
Scenario Name	Step	Action
Main Flow	1	The system displays various key expense stats on the home page
	2	User clicks on the "+" button on the home page
	3	System displays an Add Expense modal
	4	User enters Date, Expense Value, Category and Comment
	5	User clicks on the "Save" button
	6	System saves the expense in the backend of the system
	7	Home page updates to reflect the added expense
Alternative Flow – Expense less than equal to 0	4.1.1	User enters invalid expense values
	4.1.2	User clicks on the "Save" button
	4.1.3	Error message displayed "Expense should be between 0 and 8000"
	4.1.4	Back to Step 4
Alternative Flow – Expense more than 8000	4.1.1	User enters invalid expense values
	4.1.2	User clicks on the "Save" button
	4.1.3	Error message displayed "Expense should be between 0 and 8000"
	4.1.4	Back to Step 4
Alternative Flow – Empty Expenses	4.1.1	User does not enter expense value
	4.1.2	User clicks on the "Save" button
	4.1.3	Error message displayed "Expense should be between 0 and 8000"
	4.1.4	Back to Step 4
Alternative Flow – Date of Month \leq 0	4.1.1	User enters invalid date value
	4.1.2	User clicks on "Save" button
	4.1.3	Error message displayed "Date should be valid"
	4.1.4	Back to Step 4
Alternative Flow – Date of Month $>$ 30	4.1.1	User enters invalid date value
	4.1.2	User clicks on "Save" button
	4.1.3	Error message displayed "Date should be valid"

	4.1.4	Back to Step 4
Alternative Flow – Invalid Comment	4.1.1	User enters invalid comment
	4.1.2	User clicks on “Save” button
	4.1.3	Error message displayed “Comment cannot be empty & should be less than 300 characters”
	4.1.4	Back to Step 4

Table 2.2.1.5.1 Add New Expense Use Case

Test Condition ID	Test Condition	Test Coverage ID	Test Coverage	Test Data
TCON-01-038	Main Flow	TCOV-01-043	Main Flow	Date: 24/05/2022 Expense: 500 Category: Food Comment: Grocery
TCON-01-039	Alternative Flow – Expense less than equal to 0	TCOV-01-044	Alternative Flow – Expense less than equal to 0	Date: 24/05/2022 Expense: -100 Category: Food Comment: Grocery
TCON-01-040	Alternative Flow – Expense more than 8000	TCOV-01-045	Alternative Flow – Expense more than 8000	Date: 24/05/2022 Expense: 90000 Category: Car Comment: Car
TCON-01-041	Alternative Flow – Empty Expenses	TCOV-01-046	Alternative Flow – Empty Expenses	Date: 24/05/2022 Expense: - Category: Food Comment: Grocery
TCON-01-042	Alternative Flow – Date of Month <= 0	TCOV-01-047	Alternative Flow – Date of Month <= 0	Date: 0/05/2022 Expense: 500 Category: Food Comment: Grocery
TCON-01-043	Alternative Flow – Date of Month > 30	TCOV-01-048	Alternative Flow – Date of Month > 30	Date: 32/05/2022 Expense: 500 Category: Food Comment: Grocery

TCON-01-044	Alternative Flow – Invalid Comment	TCOV-01-049	Alternative Flow – Invalid Comment	Date: 28/05/2022 Expense: 500 Category: Food Comment: -
-------------	------------------------------------	-------------	------------------------------------	--

Table 2.2.1.5.2 Add New Expense Test Conditions & Coverage

2.2.2 F002 View Monthly Expenses

Five techniques are identified for designing test for this feature, including:

- i. Equivalence Partitioning
- ii. Boundary Value Analysis
- iii. Decision Table Testing
- iv. State Transition Testing
- v. Use Case Testing

Equivalence partitioning is used to split date input fields into possible partitions for test. Boundary value analysis will test the boundary values of the date input fields. Decision table is used to cover major branches of decision. State transition testing will test for major states of EM.

2.2.2.1 Equivalence Partitioning

Test Condition ID	Test Condition	Test Coverage ID	Test Coverage	Example of Test Data
TCON-02-001	Year dropdown value is lesser than current year	TCOV-02-001	Year is lesser than current year (Valid)	Year = 2019
TCON-02-002	Year dropdown value is equals to current year	TCOV-02-002	Year is equals to current year (Valid)	Year = 2022
TCON-02-003	Year dropdown value is greater than current year	TCOV-02-003	Year is greater than current year (Valid)	Year = 2025
TCON-02-004	Month dropdown value is lesser than current month	TCOV-02-004	Month dropdown value is lesser than current month (Valid)	Month = "May"
TCON-02-005	Month dropdown value is equals to current month	TCOV-02-005	Month dropdown value is equals to current month (Valid)	Month = "June"
TCON-02-006	Month dropdown value is greater than current month	TCOV-02-006	Month dropdown value is greater than current month (Valid)	Month = "July"

Table 2.2.2.1.1 View Monthly Expenses Equivalence Partitioning Test Conditions & Coverage

2.2.2.2 Boundary Value Analysis

Test Condition ID	Test Condition	Test Coverage ID	Test Coverage	Example of Test Data
TCON-02-007	Lowest possible Year dropdown value with lowest possible Month dropdown value	TCOV-02-007	Lowest possible Year dropdown value with lowest possible Month dropdown value (Valid)	Month = "January" Year = 2018
TCON-02-008	Highest possible Year dropdown value with highest possible Month dropdown value	TCOV-02-008	Highest possible Year dropdown value with highest possible Month dropdown value (Valid)	Month = "December" Year = 2026
TCON-02-009	Year dropdown value and Month dropdown value immediately preceding current year	TCOV-02-009	Year dropdown value and Month dropdown value immediately preceding current year (Valid)	Month = "December" Year = 2021
TCON-02-010	Year dropdown value and Month dropdown value immediately proceeding current year	TCOV-02-010	Year dropdown value and Month dropdown value immediately proceeding current year (Valid)	Month = "January" Year = 2023

Table 2.2.2.2.1 View Monthly Expenses Boundary Value Analysis Test Condition & Coverage

2.2.2.3 Decision Table Testing

Decision table testing cannot be performed onto this feature. This is because the only input fields provided through this feature are dropdowns. Therefore, the input space is restricted such that only valid result may be produced by this feature.

2.2.2.4 State Transition Testing

The test condition for state transition testing is as follows:

- i. TCON-02-011 View Monthly Expenses State Transition

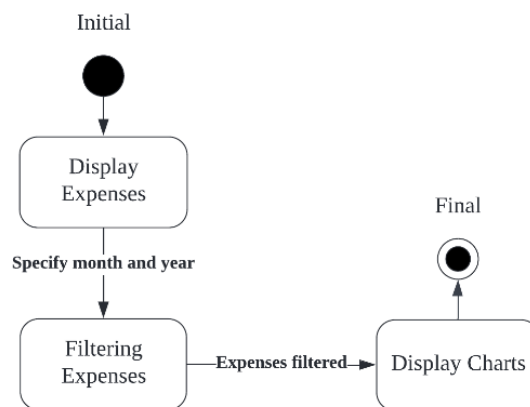


Figure 2.2.2.4.1 View Monthly Expenses State Transition Diagram

Event/State	Initial (S1)	Display Expenses (S2)	Filtering Expenses (S3)	Display Charts (S4)
Constructor (C)	S2	N/A	N/A	N/A
Specify month and year (SMY)	N/A	S3	?	?
Expenses filtered (EF)	N/A	?	S4	?
Destructor (D)	N/A	?	?	Final

Table 2.2.2.4.1 View Monthly Expenses State Table

Test Coverage ID	Test Coverage
TCOV-02-011	S1 to S2 with C
TCOV-02-012	S2 to S3 with SMY
TCOV-02-013	S3 to S4 with EF
TCOV-02-014	S4 to S5 with Final

Table 2.2.2.4.2 View Monthly Expenses State Coverage

2.2.2.5 Use Case Testing

Use Case ID	UC-2	
Use Case	View Monthly Expenses	
Purpose	The system shall allow actor to view monthly expenses based on chosen month and year	
Actor	User	
Trigger	Actor clicks “Monthly” button in the top navigation bar	
Precondition	Actor is signed in	
Scenario Name	Step	Action
Main Flow	1	The system displays the Monthly Expenses page
	2	Actor chooses Month and Year
	3	The system displays Monthly Expenses chart

Table 2.2.2.5.1 View Monthly Expenses Use Case

Test Condition ID	Test Condition	Test Coverage ID	Test Coverage	Example of Test Data
TCON-02-012	Main Flow	TCOV-02-015	Main Flow (Valid)	Month = “March” Year = 2022

Table 2.2.2.5.2 View Monthly Expenses Test Condition & Coverage

2.2.3 F003 View Daily Expenses

For this feature, we will be testing the date input for the expenses doughnut chart as this is the only type of input for this feature.

Five techniques are identified for designing test for this feature, including:

- i. Equivalence Partitioning
- ii. Boundary Value Analysis
- iii. Decision Table Testing
- iv. State Transition Testing
- v. Use Case Testing

Equivalence partitioning is used to split the date into possible partitions for test. Boundary value analysis will test the boundary values of the date input. Decision table is used to cover major branches of decision. State transition testing will test for major states of EM.

2.2.3.1 Equivalence Partitioning

Test Condition ID	Test Condition	Test Coverage ID	Test Coverage	Example of Test Data
TCON-03-001	Date <= 0	TCOV-03-001	Date <= 0 (Invalid)	Date = 0/06/2022
TCON-03-002	0 < Date <= 30	TCOV-03-002	0 < Date <= 30 (Valid)	Date = 06/06/2022
TCON-03-003	Date > 30	TCOV-03-003	Date > 30 (Invalid)	Date = 33/06/2022
TCON-03-004	Date != integer	TCOV-03-004	Date != integer (Invalid)	Date = "abc"

Table 2.2.3.1.1 View Daily Expenses Equivalence Partitioning Test Condition & Coverage

***Note:** We will be testing using the month of June so the date of month has a max value of 30

2.2.3.2 Boundary Value Analysis

Test Condition ID	Test Condition	Test Coverage ID	Test Coverage	Example of Test Data
TCOV-03-005	Date <=0	TCOV-03-005	Date <= 0 (Invalid)	Date = 0/06/2022
TCOV-03-006	Date > 30	TCOV-03-006	Date > 30 (Invalid)	Date = 31/06/2022
TCOV-03-007	0 < Date <= 30	TCOV-03-007	0 < Date <= 30 (Valid)	Date = 01/06/202
TCOV-03-008	0 < Date <= 30	TCOV-03-008	0 < Date <= 30 (Valid)	Date = 02/06/202
TCOV-03-009	0 < Date <= 30	TCOV-03-009	0 < Date <= 30 (Valid)	Date = 29/06/202
TCOV-03-010	0 < Date <= 30	TCOV-03-010	0 < Date <= 30 (Valid)	Date = 30/06/2022

Table 2.2.3.2.1 View Daily Expenses Boundary Value Analysis Test Condition & Coverage

2.2.3.3 Decision Table Testing

Test Condition ID	Test Condition
TCOV-03-011	Date != null
TCOV-03-012	Prompt error message
TCOV-03-013	Display doughnut chart

Table 2.2.3.3.1 View Daily Expenses Decision Table Test Conditions

Condition	TCOV-03-011	TCOV-03-012
Date != null	Y	N
Output		
Prompt error message	N	Y
Display doughnut chart	Y	N

Table 2.2.3.3.2 View Daily Expenses Decision Table

***Note:** The condition is Date != null because the date is set to current date by default for this feature.

2.2.3.4 State Transition Testing

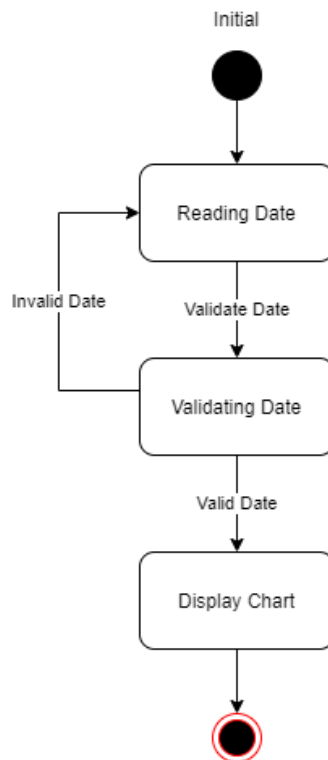


Figure 2.2.3.4.1 View Daily Expenses State Transition Diagram

The test condition for state transition testing is as follows:

- i. TCON-03-014 View Daily Expenses State Transition

Event/State	Initial (S1)	Reading Date (S2)	Validating Date (S3)	Display Chart (S4)
Constructor (C)	Reading Date (TCOV-03-013)	N/A	N/A	N/A
Validate Date (VAD)	N/A	Validating Date (TCOV-03-014)	?	?
Valid Date (VD)	N/A	?	Display Chart (TCOV-03-015)	?
Invalid Date (ID)	N/A	?	Reading Date	?

			(TCOV-03-016)	
Destructor (D)	N/A	?	?	Final (TCOV-03-017)

Table 2.2.3.4.1 View Daily Expenses State Table

Test Coverage ID	Test Coverage
TCOV-03-013	S1 to S2 with input C
TCOV-03-014	S2 to S3 with input VAD
TCOV-03-015	S3 to S4 with input VD
TCOV-03-016	S3 to S2 with input ID
TCOV-03-017	S4 to Final with input D

Table 2.2.3.4.2 View Daily Expenses State Coverage

2.2.3.5 Use Case Testing

Use Case ID	UC-3	
Use Case	View Daily Expenses	
Purpose	The system shall allow actor to view daily expenses based on chosen date.	
Actor	User	
Trigger	Actor clicks the “Daily” button in the top navigation bar	
Precondition	There is at least one expense for the chosen date	
Scenario Name	Step	Action
Main Flow	1	User chooses the date.
	2	System displays the doughnut chart of expenses for the date.
Alternative Flow – Date != integer	1.1.1	System prompts error message “Please enter a valid date!”
Alternative Flow – Date <= 0	1.2.1	System prompts error message “Please enter a valid date!”
Alternative Flow – Date > 30	1.3.1	System prompts error message “Please enter a valid date!”

Table 2.2.3.5.1 View Daily Expenses Use Case

Test Condition ID	Test Condition	Test Coverage ID	Test Coverage	Test Data
TCON-03-015	Main Flow	TCOV-03-018	Main Flow	Date = 06/06/2022
TCON-03-016	Alternative Flow – Date != integer	TCOV-03-019	Alternative Flow – Date != integer	Date = “abc”
TCON-03-017	Alternative Flow – Date <= 0	TCOV-03-020	Alternative Flow – Date <= 0	Date = 0/06/2022
TCON-03-018	Alternative Flow – Date > 30	TCOV-03-021	Alternative Flow – Date > 30	Date = 33/06/2022

Table 2.2.3.5.2 View Daily Expenses Test Conditions & Coverage

2.2.4 F004 View Overall Stats

For this feature, we will be testing the year drop down input for the expenses stats bar chart as this is the only type of input for this feature.

Five techniques are identified for designing test for this feature, including:

- i. Equivalence Partitioning
- ii. Boundary Value Analysis
- iii. State Transition Testing
- iv. Use Case Testing

Equivalence partitioning is used to split the year drop down into possible partitions for test. Boundary value analysis will test the boundary values of the year drop down input. Decision table is used to cover major branches of decision. State transition testing will test for major states of EM.

Decision table testing cannot be performed as this feature only has drop-down boxes as inputs.

2.2.4.1 Equivalence Partitioning

Test Condition ID	Test Condition	Test Coverage ID	Test Coverage	Example of Test Data
TCON-04-001	Doughnut chart year drop-down = 2022	TCOV-04-001	Doughnut chart year drop-down = 2022 (valid)	2022
TCON-04-002	Bar chart year drop-down = 2023	TCOV-04-002	Bar chart year drop-down = 2023 (valid)	2023
TCON-04-003	Each category year drop-down = 2021	TCOV-04-003	Each category year drop-down = 2021 (valid)	2021

Table 2.2.4.1.1 View Overall Stats Equivalence Partitioning Test Condition & Coverage

***Note:** The “Doughnut chart” drop-down for the year only has the options [All, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026]

The “bar chart” drop-down for the year only has the options [2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026]

The “each category” drop-down for the year only has the options [All, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026]

2.2.4.2 Boundary Value Analysis

Test Condition ID	Test Condition	Test Coverage ID	Test Coverage	Example of Test Data
TCON-04-004	Doughnut chart year drop-down = All	TCOV-04-004	Doughnut chart year drop-down = All (Valid)	All
TCON-04-005	Doughnut chart year drop-down = 2026	TCOV-04-005	Doughnut chart year drop-down = 2026 (Valid)	2026
TCON-04-006	Bar chart year drop-down = 2018	TCOV-04-006	Bar chart year drop-down = 2018 (valid)	2018
TCON-04-007	Bar chart year drop-down = 2026	TCOV-04-007	Bar chart year drop-down = 2026 (valid)	2026
TCON-04-008	Each category year drop-down = All	TCOV-04-008	Each category year drop-down = All (Valid)	All
TCON-04-009	Each category year drop-down = 2026	TCOV-04-009	Each category year drop-down = 2026 (valid)	2026

Table 2.2.4.2.1 View Overall Stats Boundary Value Analysis Test Condition & Coverage

2.2.4.3 State Transition Testing

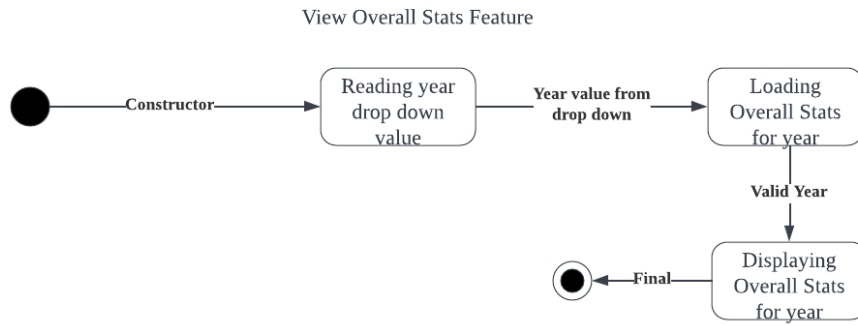


Figure 2.2.4.3.1 View Overall Stats State Transition Diagram

The test condition for state transition testing is as following:

I. TCON-04-010 View overall stats state transition

	Initial	Reading Year Drop-down Value (S1)	Loading Overall Stats for year (S2)	Displaying overall stats for year (S3)
Constructor (C)	Reading Year Drop-down Value (TCOV-04-010)	N/A	N/A	N/A
Drop-down field (DDF)	N/A	Loading overall stats for year (TCOV-04-011)	?	?
Valid Year (VY)	N/A	?	Displaying overall stats for year (TCOV-04-012)	Final (TCOV-04-013)

Table 2.2.4.3.1 View Overall Stats State Table

Test Coverage:

Test Coverage ID	Test Coverage
TCOV-04-010	Initial to S1 with C
TCOV-04-011	S1 to S2 with DDF
TCOV-04-012	S2 to S3 with VY
TCOV-04-013	S3 to Final with VY

Table 2.2.4.3.2 View Overall Stats State Coverage

2.2.4.4 Use Case Testing

Use Case ID	UC-5	
Use Case	View Overall Stats	
Purpose	The system shall allow actor to view overall expenses statistics	
Actor	User	
Trigger	Actor clicks the “Stat” button in the top navigation bar	
Precondition	-	
Scenario Name	Step	Action
Main Flow	1	User clicks on the “Stat” button on the top navigation bar
	2	System navigates to the view overall stats page
	3	User enters the Year from the “doughnut chart” drop-down box for which they wish to see the expense statistics
	4	The system loads the expense statistics for the year
	5	The system displays the expense statistics for the year chosen in the form of a doughnut chart, bar chart and general overall statistics for that year
Alternative Flow – User enters year for the “bar chart” drop-down box	3.1	User enters a year from the drop down for the “bar chart” section of the overall stats page
	3.2	Continue to step 4
Alternative Flow – User enters year for the “each category” drop-down box	3.1	User enters a year from the drop down for the “each category” section of the overall stats page
	3.2	Continue to step 4

Table 2.2.4.4.1 View Overall Stats Use Case

Test Condition ID	Test Condition	Test Coverage ID	Test Coverage	Test Data
TCON-04-011	Main Flow	TCOV-04-014	Main Flow	Year: 2022
TCON-04-012	Alternative Flow – User enters year for the “bar chart” drop-down box	TCOV-04-015	Alternative Flow – User enters year for the “bar chart” drop-down box	Year: 2019
TCON-04-013	Alternative Flow – User enters year for the “each category” drop-down box	TCOV-04-016	Alternative Flow – User enters year for the “each category” drop-down box	Year: 2020

Table 2.2.4.4.2 View Overall Stats Test Conditions & Coverage

2.2.5 F005 Filter Expenses

Five techniques are identified for designing test for this feature, including:

- i. Equivalence Partitioning
- ii. Boundary Value Analysis
- iii. Decision Table Testing
- iv. State Transition Testing
- v. Use Case Testing

Equivalence partitioning is used to split date range and expense range into possible partitions for test. Boundary value analysis will test the boundary values of the date range and expense range. Decision table is used to cover major branches of decision. State transition testing will test for major states of EM.

2.2.5.1 Equivalence Partitioning

Test Condition ID	Test Condition	Test Coverage ID	Test Coverage	Example of Test Data
TCON-05-001	From Date <= To Date	TCOV-05-001	From Date <= To Date (Valid)	From Date = 01/01/2000 To Date = 01/02/2000
TCON-05-002	From Date > To Date	TCOV-05-002	From Date > To Date (Invalid)	From Date = 01/02/2000 To Date = 01/01/2000
TCON-05-003	One of the input in date range fields is null	TCOV-05-003	One of the input in date range fields is null (Invalid)	From Date = null To Date = 01/01/2000
TCON-05-004	Both input in date range fields is null	TCOV-05-004	Both input in date range fields is null (Invalid)	From Date = null To Date = null
TCON-05-005	0 < From Expense <= To Expense	TCOV-05-005	0 < From Expense <= To Expense (Valid)	From Expense = 1 To Expense = 2
TCON-05-006	From Expense > To Expense	TCOV-05-006	From Expense > To Expense (Invalid)	From Expense = 2 To Expense = 1
TCON-05-007	One of the input in expense range fields is of String type	TCOV-05-007	One of the input in expense range fields is of String type (Invalid)	From Expense = "abc" To Expense = 1
TCON-05-008	Both input in expense range fields is of String type	TCOV-05-008	Both input in expense range fields is of String type (Invalid)	From Expense = "abc" To Expense = "abc"
TCON-05-009	Category Dropdown = "Food"	TCOV-05-009	Category Dropdown = "Food" (Valid)	Category Dropdown = "Food"

Table 2.2.5.1.1 Filter Expense Equivalence Partitioning Test Condition & Coverage

2.2.5.2 Boundary Value Analysis

Test Condition ID	Test Condition	Test Coverage ID	Test Coverage	Example of Test Data
TCON-05-010	0 < From Expense <= To Expense	TCOV-05-010	0 < From Expense <= To Expense (Valid)	From Expense = 2 To Expense = 3
TCON-05-011	From Expense <= 0	TCOV-05-011	From Expense <= 0 (Invalid)	From Expense = -1 To Expense = 1
TCON-05-012	To Expense <= 0	TCOV-05-012	To Expense <= 0 (Invalid)	From Expense = 1 To Expense = -1
TCON-05-013	From Expense <= 0 & To Expense <= 0	TCOV-05-013	From Expense <= 0 & To Expense <= 0 (Invalid)	From Expense = -1 To Expense = -1

Table 2.2.5.2.1 Filter Expenses Boundary Value Analysis Test Condition & Coverage

2.2.5.3 Decision Table Testing

Test Condition ID	Test Condition
TCON-05-001	From Date <= To Date
TCON-05-005	0 < From Expense <= To Expense
TCON-05-014	Error Message
TCON-05-015	Filter Expenses

Table 2.2.5.3.1 Filter Expenses Decision Table Test Conditions

Condition	TCOV-05-014	TCOV-05-015	TCOV-05-016	TCOV-05-017
TCON-05-001	Y	N	Y	N
TCON-05-005	Y	N	N	Y
Output				
TCON-05-014	N	Y	Y	Y
TCON-05-015	Y	N	N	N

Table 2.2.5.3.2 Filter Expenses Decision Table

2.2.5.4 State Transition Testing

The test condition for state transition testing is as follow:

- i. TCON-05-015 Filter Expenses

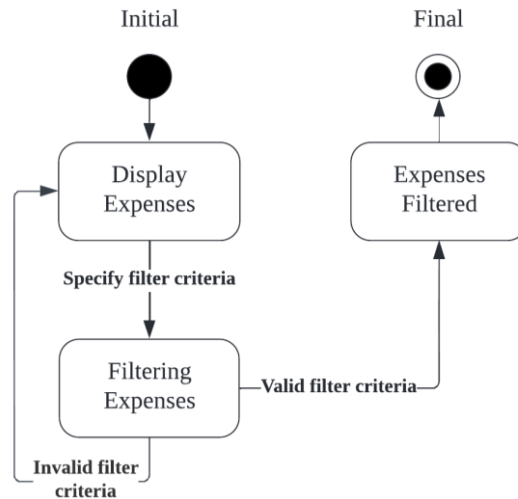


Figure 2.2.5.4.1 Filter Expenses State Transition Diagram

Event/State	Initial (S1)	Display Expenses (S2)	Filtering Expenses (S3)	Expenses Filtered (S4)
Constructor (C)	S2	N/A	N/A	N/A
Specify filter criteria (SFC)	N/A	S3	?	?
Invalid filter criteria (IFC)	N/A	?	S2	?
Valid filter criteria (VFC)	N/A	?	S4	?
Destructor (D)	N/A	?	?	Final

Table 2.2.5.4.1 Filter Expenses State Table

Test Coverage ID	Test Coverage
TCOV-05-018	S1 to S2 with C
TCOV-05-019	S2 to S3 with SFC
TCOV-05-020	S3 to S2 with IFC
TCOV-05-021	S3 to S4 with VFC
TCOV-05-022	S4 to Final with D

Table 2.2.5.4.2 Filter Expenses State Coverage

2.2.5.5 Use Case Testing

Use Case ID	UC-4	
Use Case	Filter Expenses	
Purpose	The system shall allow actor to filter expenses based on specified criteria	
Actor	User	
Trigger	Actor clicks the “filter” button in the top navigation bar	
Precondition	Expenses exists in the database	
Scenario Name	Step	Action
Main Flow	1	The system displays data table of expenses of current date
	2	Actor specifies filter criteria
	3	The system displays data table of filtered expenses
Alternative Flow – From Date is greater than To Date	2.1.1	Actor specify wrong date range
	2.1.2	The system displays “From Date must be less than or equals to To Date” error message
Alternative Flow – One or both date range input fields is null	2.2.1	Actor left one or both date range input fields empty
	2.2.2	The system displays “Date range inputs cannot be null”
Alternative Flow – From Expense is greater than To Expense	2.3.1	Actor specify wrong expense range
	2.3.2	The system displays “From Expense must be less than or equals to To Expense” error message
Alternative Flow – One or both expense range input fields is inputted a value less than or equal to zero	2.4.1	Actor provided a value less than or equal to zero to one or both expense range input fields
	2.4.2	The system displays “Expense range inputs must be greater than zero” error message
Alternative Flow – One or both expense range input fields is inputted a value of string type	2.5.1	Actor provided a value of string type into one or both expense range input fields
	2.5.2	The system displays “Expense range inputs cannot be of string type” error message

Table 2.2.5.5.1 Filter Expenses Use Case

Test Condition ID	Test Condition	Test Coverage ID	Test Coverage	Example of Test Data
TCON-05-016	Main Flow	TCOV-05-023	Main Flow	From Date = 01/01/2000 To Date = 01/02/2000 From Expense = 1 To Expense = 2 Category = "Food"
TCON-05-017	Alternative Flow – From Date is greater than To Date	TCOV-05-024	Alternative Flow – From Date is greater than To Date	From Date = 01/02/2000 To Date = 01/01/2000 From Expense = 1 To Expense = 2 Category = "Food"
TCON-05-018	Alternative Flow – One or both date range input fields is null	TCOV-05-025	Alternative Flow – One or both date range input fields is null	From Date = 01/01/2000 To Date = null From Expense = 1 To Expense = 2 Category = "Food"
TCON-05-019	Alternative Flow – From Expense is greater than To Expense	TCOV-05-026	Alternative Flow – From Expense is greater than To Expense	From Date = 01/01/2000 To Date = 01/02/2000 From Expense = 2 To Expense = 1 Category = "Food"
TCON-05-020	Alternative Flow – One or both expense range input fields is inputted a value less than or equal to zero	TCOV-05-027	Alternative Flow – One or both expense range input fields is inputted a value less than or equal to zero	From Date = 01/01/2000 To Date = 01/02/2000 From Expense = -3 To Expense = -5 Category = "Food"
TCON-05-021	Alternative Flow – One or both expense range input fields is inputted a value of string type	TCOV-05-028	Alternative Flow – One or both expense range input fields is inputted a value of string type	From Date = 01/01/2000 To Date = 01/02/2000 From Expense = "j"

				To Expense = "k" Category = "Food"
--	--	--	--	---------------------------------------

Table 2.2.5.5.2 Filter Expenses Test Condition & Coverage

2.2.6 F006 Manage Loan

Five techniques are identified for designing test for this feature, including:

- vi. Equivalence Partitioning
- vii. Boundary Value Analysis
- viii. Decision Table Testing
- ix. State Transition Testing
- x. Use Case Testing

Equivalence partitioning is used to split the date, loan amount, loan type, person name length, status and reason length into possible partitions for test. Boundary value analysis will test the boundary values of the loan amount and date input. Decision table is used to cover major branches of decision. State transition testing will test for major states of EM.

***Note:** Since Add Loan and Edit Loan both have same form fields, the testing for these functions have been combined in Equivalence Partitioning, Boundary Value Analysis and Decision Table Testing. However, they have defined specifically in State Transition Testing and Use Case Testing. Delete Loan is also tested in State Transition Testing and Use Case Testing only.

2.2.6.1 Equivalence Partitioning

Test Condition ID	Test Condition	Test Coverage ID	Test Coverage	Example of Test Data
TCON-06-001	Date <= 0	TCOV-06-001	Date <= 0 (Invalid)	Date = 0/06/2022
TCON-06-002	0 < Date <= 30	TCOV-06-002	0 < Date <= 30 (Valid)	Date = 06/06/2022
TCON-06-003	Date > 30	TCOV-06-003	Date > 30 (Invalid)	Date = 33/06/2022
TCON-06-004	Date != integer	TCOV-06-004	Date != integer (Invalid)	Date = "abc"
TCON-06-005	Amount <= 0	TCOV-06-005	Amount <= 0 (Invalid)	Amount = -5000
TCON-06-006	Amount > 100000	TCOV-06-006	Amount > 100000 (Invalid)	Amount = 200000
TCON-06-007	0 < Amount <= 100000	TCOV-06-007	0 < Amount <= 100000 (Valid)	Amount = 5000
TCON-06-008	Loan Type = "Given"	TCOV-06-008	Loan Type = "Given" (Valid)	Loan Type = "Given"
TCON-06-009	Person length = 0	TCOV-06-009	Person length = 0 (Invalid)	Person = ""
TCON-06-010	0 < Person length <= 80	TCOV-06-010	0 < Person length <= 80 (Valid)	Person = "Abdullah"
TCON-06-011	Person length > 80	TCOV-06-011	Person length > 80 (Invalid)	Person = "Siti Sarah Aqilah Rosaini Minah Binti Muhammad Abdullah Huhkum Kasim Karim Rahman"
TCON-06-012	Status = "Pending"	TCOV-06-012	Status = "Pending" (Valid)	Status = "Pending"

TCON-06-013	Reason length = 0	TCOV-06-013	Reason length = 0 (Invalid)	Reason = ""
TCON-06-014	0 < Reason length <= 300	TCOV-06-014	0 < Reason length <= 300 (Valid)	Reason = "Pay hospital fees"
TCON-06-015	Reason length > 300	TCOV-06-015	Reason length > 300 (Invalid)	Reason = "As a home owner I need to use personal loan to upgrade (renovate) my home and complete necessary repairs, like fixing the plumbing or redoing the electrical wiring. A personal loan is a good fit for me who don't have equity in my home and don't want to get a home equity line of credit or home equity loan."

Table 2.2.6.1.1 Manage Loan Equivalence Partitioning Test Conditions & Coverage

***Note:** We will be using the month of June so the date of month has a max value of 30

100000 has been used as the upper limit for loan amount according to Maybank Personal Loan limit.

80 characters has been used as the upper limit for person name length as we cannot fit more than 80 characters in Malaysian Identification Card (IC).

For the max length of the reason, we are assuming that to be 300 characters based on the text field size on the UI

2.2.6.2 Boundary Value Analysis

Test Condition ID	Test Condition	Test Coverage ID	Test Coverage	Example of Test Data
TCON-06-016	Date <=0	TCOV-06-016	Date <= 0 (Invalid)	Date = 0/06/2022
TCON-06-017	Date > 30	TCOV-06-017	Date > 30 (Invalid)	Date = 31/06/2022
TCON-06-018	0 < Date <= 30	TCOV-06-018	0 < Date <= 30 (Valid)	Date = 01/06/202
TCON-06-019	0 < Date <= 30	TCOV-06-019	0 < Date <= 30 (Valid)	Date = 02/06/202
TCON-06-020	0 < Date <= 30	TCOV-06-020	0 < Date <= 30 (Valid)	Date = 29/06/202
TCON-06-021	0 < Date <= 30	TCOV-06-021	0 < Date <= 30 (Valid)	Date = 30/06/2022
TCON-06-022	Amount <= 0	TCOV-06-022	Amount <= 0 (Invalid)	Amount = 0
TCON-06-023	Amount > 100000	TCOV-06-023	Amount > 100000 (Invalid)	Amount = 100001
TCON-06-024	0 < Amount <= 100000	TCOV-06-024	0 < Amount <= 100000 (Valid)	Amount = 1
TCON-06-025	0 < Amount <= 100000	TCOV-06-025	0 < Amount <= 100000 (Valid)	Amount = 2
TCON-06-026	0 < Amount <= 100000	TCOV-06-026	0 < Amount <= 100000 (Valid)	Amount = 99999
TCON-06-027	0 < Amount <= 100000	TCOV-06-027	0 < Amount <= 100000 (Valid)	Amount = 100000
TCON-06-028	Loan Type = "Given"	TCOV-06-028	Loan Type = "Given" (Valid)	Loan Type = "Given"
TCON-06-029	Loan Type = "Taken"	TCOV-06-029	Loan Type = "Taken" (Valid)	Loan Type = "Taken"
TCON-06-030	Person length = 0	TCOV-06-030	Person length = 0 (Invalid)	Person = ""
TCON-06-031	Person length > 80	TCOV-06-031	Person length > 80 (Invalid)	Person = "Siti Sarah Aqilah Rosaini Minah Binti Muhammad Abdullah Huhkum Kasim Karim Rahman"

TCON-06-032	0 < Person length <= 80	TCOV-06-032	0 < Person length <= 80 (Valid)	Person = "a"
TCON-06-033	0 < Person length <= 80	TCOV-06-033	0 < Person length <= 80 (Valid)	Person = "ab"
TCON-06-034	0 < Person length <= 80	TCOV-06-034	0 < Person length <= 80 (Valid)	Person = "Siti Sarah Aqilah Rosaini Minah Binti Muhammad Abdullah Huhkum Kasim Karim Rahm"
TCON-06-035	0 < Person length <= 80	TCOV-06-035	0 < Person length <= 80 (Valid)	Person = "Siti Sarah Aqilah Rosaini Minah Binti Muhammad Abdullah Huhkum Kasim Karim Rahma"
TCON-06-036	Status = "Pending"	TCOV-06-036	Status = "Pending" (Valid)	Status = "Pending"
TCON-06-037	Status = "Settled"	TCOV-06-037	Status = "Settled" (Valid)	Status = "Settled"
TCON-06-038	Reason length = 0	TCOV-06-038	Reason length = 0 (Invalid)	Reason = ""
TCON-06-039	Reason length > 300	TCOV-06-039	Reason length > 300 (Invalid)	Reason = ""As a homeowner I need to use personal loan to upgrade (renovate) my home and complete necessary repairs, like fixing the plumbing or redoing the electrical wiring. A personal loan is a good fit for me who don't have equity in my home and don't want to get a home equity line of credit or home equity loan."

TCON-06-040	0 < Reason length <= 300	TCOV-06-040	0 < Reason length <= 300 (Valid)	Reason = "a"
TCON-06-041	0 < Reason length <= 300	TCOV-06-041	0 < Reason length <= 300 (Valid)	Reason = "ab"
TCON-06-042	0 < Reason length <= 300	TCOV-06-042	0 < Reason length <= 300 (Valid)	Reason = "As a homeowner I need to use personal loan to upgrade (renovate) my home and complete necessary repairs, like fixing the plumbing or redoing the electrical wiring. A personal loan is a good fit for me who don't have equity in my home and don't want to get a home equity line of credit or home eq"
TCON-06-043	0 < Reason length <= 300	TCOV-06-043	0 < Reason length <= 300 (Valid)	"As a homeowner I need to use personal loan to upgrade (renovate) my home and complete necessary repairs, like fixing the plumbing or redoing the electrical wiring. A personal loan is a good fit for me who don't have equity in my home and don't want to get a home equity line of credit or home equ"

Table 2.2.6.2.1 Manage Loan Boundary Value Analysis Test Condition & Coverage

2.2.6.3 Decision Table Testing

Test Condition ID	Test Condition
TCON-06-044	0 < Date <= 30
TCON-06-045	0 < Amount <= 100000
TCON-06-046	0 < Person <= 80
TCON-06-047	0 < Reason <= 300
TCON-06-048	Error Dialog Message
TCON-06-049	Add new loan

Table 2.2.6.3.1 Manage Loan Decision Table Test Conditions

Condition	TCOV-06-044	TCOV-06-045	TCOV-06-046	TCOV-06-047	TCOV-06-048	TCOV-06-049
0 < Date <= 30	Y	N	N	N	N	Y
0 < Amount <= 8000	Y	N	N	N	Y	N
0 < Person <= 80	Y	N	N	Y	N	N
0 < Reason <= 300	Y	N	Y	N	N	N
Output						
Error Dialog Message	N	Y	Y	Y	Y	Y
Add new loan	Y	N	N	N	N	N

Table 2.2.6.3.2 Manage Loan Decision Table

2.2.6.4 State Transition Testing

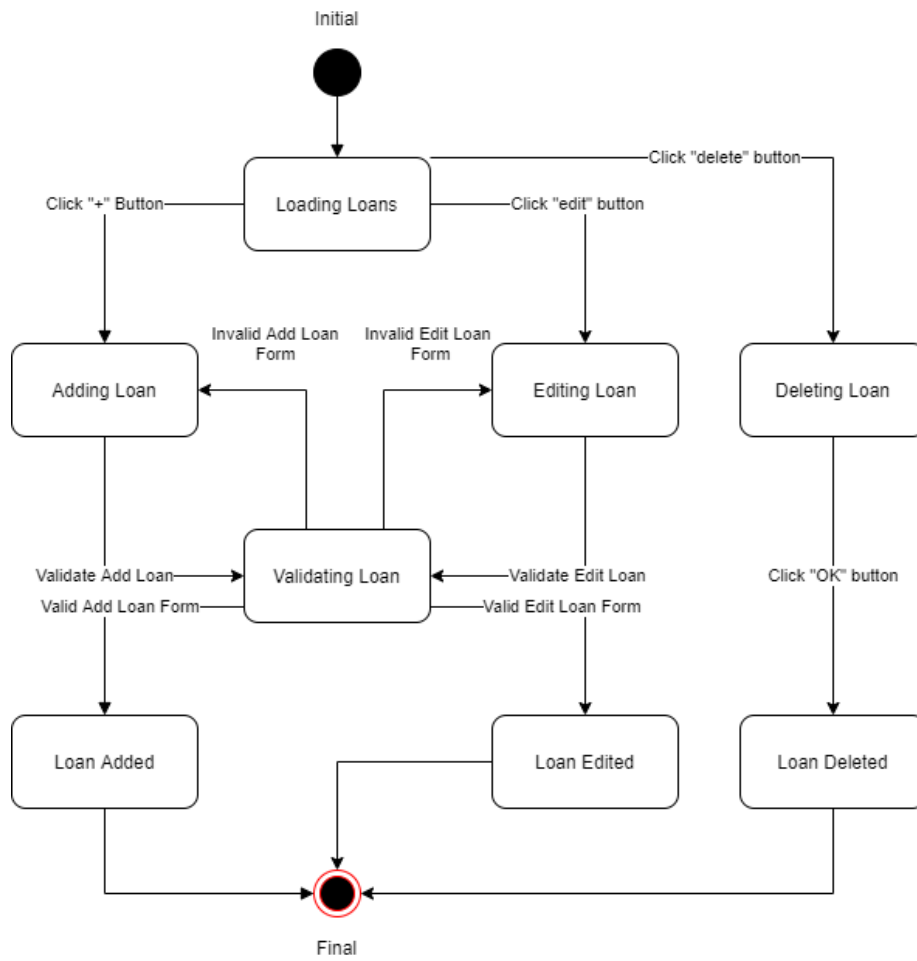


Figure 2.2.6.4.1 Manage Loan State Transition Diagram

The test condition for state transition testing is as follows:

- i. TCON-06-050 Manage Loan State Transition

Event/ State	Initial (S1)	Loading Loans (S2)	Adding Loan (S3)	Editing Loan (S4)	Deleting Loan (S5)	Validating Loan (S6)	Loan Added (S7)	Loan Edited (S8)	Loan Deleted (S9)
Constructor (C)	Loading Loans (TCOV-06-050)	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Click “+” Button (CAB)	N/A	Adding Loan (TCOV-06-051)	?	?	?	?	?	?	?
Click “edit” Button (CEB)	N/A	Editing Loan (TCOV-06-052)	?	?	?	?	?	?	?
Click “delete” Button (CDB)	N/A	Deleting Loan (TCOV-06-053)	?	?	?	?	?	?	?
Validate Add Loan (VAL)	N/A	?	Validating Loan (TCOV-06-054)	?	?	?	?	?	?
Validate Edit Loan (VEL)	N/A	?	?	Validating Loan (TCOV-06-055)	?	?	?	?	?

Click “OK” Button (COB)	N/A	?	?	?	Loan Deleted (TCOV-06-056)	?	?	?	?
Valid Add Loan Form (VALF)	N/A	?	?	?	?	Loan Added (TCOV-06-057)	?	?	?
Valid Edit Loan Form (VELF)	N/A	?	?	?	?	Loan Edited (TCOV-06-058)	?	?	?
Invalid Add Loan Form (IALF)	N/A	?	?	?	?	Adding Loan (TCOV-06-059)	?	?	?
Invalid Edit Loan Form (IELF)	N/A	?	?	?	?	Editing Loan (TCOV-06-060)	?	?	?
Destructor (D)	N/A	?	?	?	?	?	Final (TCOV-06-061)	Final (TCOV-06-062)	Final (TCOV-06-063)

Table 2.2.6.4.1 Manage Loan State Table

Test Coverage ID	Test Coverage
TCOV-06-050	S1 to S2 with input C
TCOV-06-051	S2 to S3 with input CAB
TCOV-06-052	S2 to S4 with input CEB
TCOV-06-053	S2 to S5 with input CDB
TCOV-06-054	S3 to S6 with input VAL
TCOV-06-055	S4 to S6 with input VEL
TCOV-06-056	S5 to S9 with input COB
TCOV-06-057	S6 to S7 with input VALF
TCOV-06-058	S6 to S8 with input VELF
TCOV-06-059	S6 to S3 with input IALF
TCOV-06-060	S6 to S4 with input IELF
TCOV-06-061	S7 to Final with input D
TCOV-06-062	S8 to Final with input D
TCOV-06-063	S9 to Final with input D

Table 2.2.6.4b Manage Loan State Coverage

2.2.6.5 Use Case Testing

Use Case ID	UC-6	
Use Case	Manage Loan	
Purpose	The system shall allow actor to manage (add, edit, delete) loan.	
Actor	User	
Trigger	Actor clicks “Loan” button in the top navigation bar.	
Precondition	Loan data exists in the database	
Scenario Name	Step	Action
Main Flow	1	The system displays total loan to repay, total loan to get and data table.
	2	User clicks on the “+” button.
	3	System displays Add Loan Form.
	4	User enters Date, Amount, Loan Type, Person, Status and Reason.
	5	User clicks on the “Save” button.
	7	System verifies the input fields and updates the loan in the database.
Alternative Flow – Edit an existing loan	2.1.1	User clicks “edit” button of a specific loan.
	2.1.2	System displays the loan into editable form.
	2.1.3	User edits the Date, Amount, Loan Type, Person, Status and Reason.
	2.1.4	User clicks the “Save” button.
	2.1.5	System verifies the input fields and updates the loan in the database.
Alternative Flow – Delete an existing loan	2.2.1	User clicks “delete” button from a specific loan.
	2.2.2	System prompts a confirmation message.
	2.2.3	User clicks “OK” button.
Alternative Flow – Invalid date	4.1.1	User enters an invalid date.
	4.1.2	Back to Main Flow 5.
Alternative Flow – Invalid amount	4.2.1	User enters invalid amount.
	4.2.2	Back to Main Flow 5.
Alternative Flow – Invalid person	4.3.1	User enters invalid person.
	4.3.2	Back to Main Flow 5.
Alternative Flow – Invalid reason	4.4.1	User enters invalid reason.
	4.4.2	Back to Main Flow 5.

Table 2.2.6.5.1 Manage Loan Use Case

Test Condition ID	Test Condition	Test Coverage ID	Test Coverage	Test Data
-------------------	----------------	------------------	---------------	-----------

TCON-06-051	Main Flow	TCOV-06-064	Main Flow	Date: 07/06/2022 Amount: 200 Loan Type: Taken Person: Ahmad Status: Pending Reason: Medical Fees
TCON-06-052	Alternative Flow – Edit an existing loan	TCOV-06-065	Alternative Flow – Edit an existing loan	Date: 07/06/2022 Amount: 300 Loan Type: Taken Person: Ahmad Status: Pending Reason: Medical Fees
TCON-06-053	Alternative Flow – Delete an existing loan	TCOV-06-066	Alternative Flow – Delete an existing loan	Press “delete” button. Press “OK” button.
TCON-06-054	Alternative Flow – Invalid date	TCOV-06-067	Alternative Flow – Invalid date	Date: null Amount: 200 Loan Type: Taken Person: Ahmad Status: Pending Reason: Medical Fees
TCON-06-055	Alternative Flow – Invalid amount	TCOV-06-068	Alternative Flow – Invalid amount	Date: 07/06/2022 Amount: null Loan Type: Taken Person: Ahmad Status: Pending Reason: Medical Fees
TCON-06-056	Alternative Flow – Invalid person	TCOV-06-069	Alternative Flow – Invalid person	Date: 07/06/2022 Amount: 200 Loan Type: Taken Person: null Status: Pending Reason: Medical Fees
TCON-06-057	Alternative Flow – Invalid reason	TCOV-06-070	Alternative Flow –	Date: 07/06/2022 Amount: 200 Loan Type: Taken

			Invalid reason	Person: Ahmad Status: Pending Reason: null
--	--	--	-------------------	--

Table 2.2.6.5.2 Manage Loan Test Conditions & Coverage

2.2.7 F007 Manage Savings

Five techniques are identified for designing test for this feature, including:

- i. Equivalence Partitioning
- ii. Boundary Value Analysis
- iii. Decision Table Testing
- iv. State Transition Testing
- v. Use Case Testing

Equivalence partitioning is used to split the saving goal amount, saving amount and target date into possible partitions for test. Boundary value analysis will test the boundary values of the saving goal amount, saving amount and date input. Decision table is used to cover major branches of decision. State transition testing will test for major states of EM.

2.2.7.1 Equivalence Partitioning

Test Condition ID	Test Condition	Test Coverage ID	Test Coverage	Example of Test Data
TCON-07-001	Goal Amount < 1	TCOV-07-001	Goal Amount < 1 (Invalid)	-10
TCON-07-002	Goal Amount > 80000	TCOV-07-002	Goal Amount > 80000 (Invalid)	90000
TCON-07-003	1 <= Goal Amount <= 80000	TCOV-07-003	1 <= Goal Amount <= 80000 (Valid)	4128
TCON-07-004	Goal Amount <= Saving Amount	TCOV-07-004	Goal Amount <= Saving Amount (Invalid)	Goal Amount: 2000 Saving Amount: 3000
TCON-07-005	Goal Amount > Saving Amount	TCOV-07-005	Goal Amount > Saving Amount (Valid)	Goal Amount: 100 Saving Amount: 0

TCON-07-006	Saving Amount < 0	TCOV-07-006	Saving Amount < 0 (Invalid)	-5
TCON-07-007	Saving Amount > 80000	TCOV-07-007	Saving Amount > 80000 (Invalid)	100000
TCON-07-008	0 <= Saving Amount <= 80000	TCOV-07-008	0 <= Saving Amount <= 80000 (Valid)	10
TCON-07-009	Date <= 0	TCOV-07-009	Date <= 0 (Invalid)	Date = 0/06/2022
TCON-07-010	0 < Date <= 30	TCOV-07-010	0 < Date <= 30 (Valid)	Date = 06/06/2022
TCON-07-011	Date > 30	TCOV-07-011	Date > 30 (Invalid)	Date = 33/06/2022
TCON-07-012	Date != integer	TCOV-07-012	Date != integer (invalid)	Date: "abc"
TCON-07-013	Comment length = 0	TCOV-07-013	Comment length = 0 (invalid)	Comment = ""
TCON-07-014	0 < Comment length <= 300	TCOV-07-014	0 < Comment length <= 300 (valid)	Comment = "Grocery"
TCON-07-015	Comment length > 300	TCOV-07-015	Comment length > 300 (invalid)	Comment = 350 chars
TCON-07-016	Category = "Food"	TCOV-07-016	Category = "Food" (valid)	Category = "Food"

Table 2.2.7.1.1 Manage Savings Equivalence Partitioning Test Conditions & Coverage

***Note:** 80000 has been used as the upper limit considering the average salary in Malaysia is 80,000 MYR annually

2.2.7.2 Boundary Value Analysis

Test Condition ID	Test Condition	Test Coverage ID	Test Coverage	Example of Test Data
TCON-07-017	Goal Amount < 1	TCOV-07-017	Goal Amount < 1 (Invalid)	0
TCON-07-018	Goal Amount > 80000	TCOV-07-018	Goal Amount > 80000 (Invalid)	80001
TCON-07-019	1 <= Goal Amount <= 80000	TCOV-07-019	1 <= Goal Amount <= 80000 (Valid)	1
TCON-07-020	1 <= Goal Amount <= 80000	TCOV-07-020	1 <= Goal Amount <= 80000 (Valid)	80000
TCON-07-021	Goal Amount <= Saving Amount	TCOV-07-021	Goal Amount <= Saving Amount (Invalid)	Goal Amount: 10 Saving Amount: 10
TCON-07-022	Goal Amount <= Saving Amount	TCOV-07-022	Goal Amount <= Saving Amount (Invalid)	Goal Amount: 9 Saving Amount: 10
TCON-07-023	Goal Amount > Saving Amount	TCOV-07-023	Goal Amount > Saving Amount (Valid)	Goal Amount: 10 Saving Amount: 9
TCON-07-024	Saving Amount < 0	TCOV-07-024	Saving Amount < 0 (Invalid)	-1
TCON-07-025	Saving Amount > 80000	TCOV-07-025	Saving Amount > 80000 (Invalid)	80001
TCON-07-026	0 <= Saving Amount <= 80000	TCOV-07-026	0 <= Saving Amount <= 80000 (Valid)	0

TCON-07-027	0 <= Saving Amount <= 80000	TCOV-07-027	0 <= Saving Amount <= 80000 (Valid)	80000
TCON-07-028	0 < Date <= 30	TCOV-07-028	0 < Date <= 30 (Valid)	Date = 01/06/2022
TCON-07-029	0 < Date <= 30	TCOV-07-029	0 < Date <= 30 (Valid)	Date = 02/06/2022
TCON-07-030	0 < Date <= 30	TCOV-07-030	0 < Date <= 30 (Valid)	Date = 29/06/2022
TCON-07-031	0 < Date <= 30	TCOV-07-031	0 < Date <= 30 (Valid)	Date = 30/06/2022
TCON-07-032	Category = "Food"	TCOV-07-032	Category = "Food"	Category = "Food"
TCON-07-033	Category = "Others"	TCOV-07-033	Category = "Others"	Category = "Others"
TCON-07-034	Comment length = 0	TCOV-07-034	Comment length = 0 (invalid)	Comment = ""
TCON-07-035	Comment length > 300	TCOV-07-035	Comment length > 300 (invalid)	Comment length = 301 chars
TCON-07-036	0 < Comment length <= 300	TCOV-07-036	0 < Comment length <= 300 (valid)	Comment = "a"
TCON-07-037	0 < Comment length <= 300	TCOV-07-037	0 < Comment length <= 300 (valid)	Comment = "ab"
TCON-07-038	0 < Comment length <= 300	TCOV-07-038	0 < Comment length <= 300 (valid)	Comment length = 299 chars

Table 2.2.7.2.1 Manage Savings Boundary Value Analysis Test Condition & Coverage

2.2.1.3 Decision Table Testing

Test Condition ID	Test Condition
TCON-07-039	Goal Amount > 0
TCON-07-040	Goal Amount > Saving Amount
TCON-07-041	Saving Amount > -1
TCON-07-042	0 < Date <= 31
TCON-07-043	Comment != null
TCON-07-044	Error Dialog Message
TCON-07-045	Create new Savings Goal

Table 2.2.1.3.1 Manage Savings Goal Decision Table Test Conditions

Condition	TCOV-07-039	TCOV-07-040	TCOV-07-041	TCOV-07-042	TCOV-07-043	TCOV-07-044	TCOV-07-045
Goal Amount > 0	Y	N	N	N	N	N	Y
Goal Amount > Saving Amount	Y	N	N	N	N	Y	N
Saving Amount > -1	Y	N	N	N	Y	N	N
0 < Date of Month <= 31	Y	N	N	Y	N	N	N
Comment != null	Y	N	Y	N	N	N	N
Output							
Error Dialog Message	N	Y	Y	Y	Y	Y	Y
Create new Savings Goal	Y	N	N	N	N	N	N

Table 2.2.7.3.2 Manage Savings Goal Decision Table

2.2.7.4 State Transition Testing

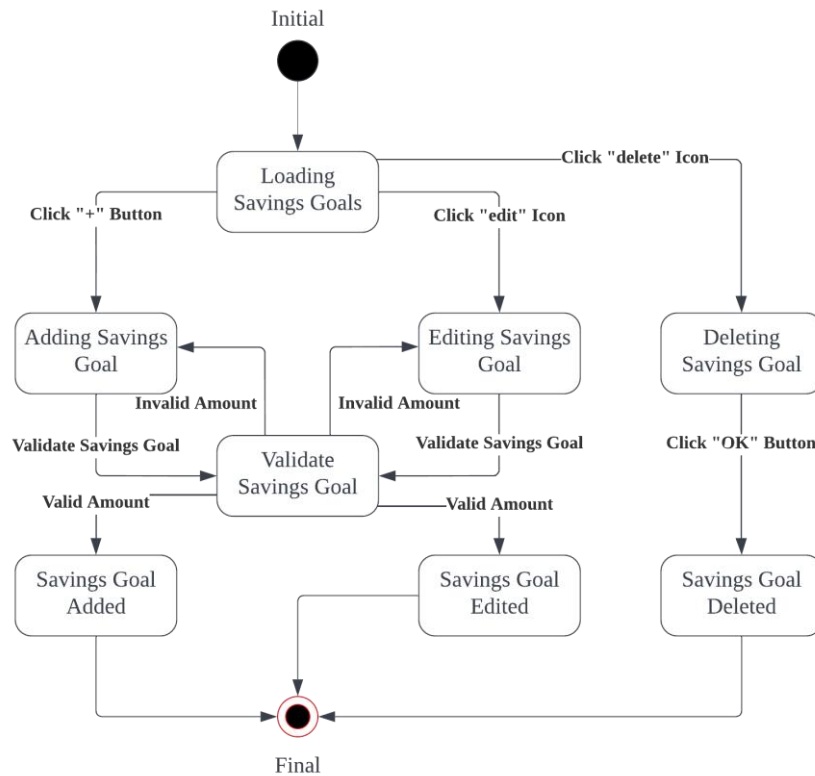


Figure 2.2.7.4.1 Manage Savings State Transition Diagram

The test condition for state transition testing is as follows:

- i. TCON-07-046 Manage Savings State Transition

Event/ State	Initial (S1)	Loading Savings Goals (S2)	Adding Savings Goal (S3)	Editing Savings Goal (S4)	Deleting Savings Goal (S5)	Validating Savings Goal (S6)	Savings Goal Added (S7)	Savings Goal Edited (S8)	Savings Goal Deleted (S9)
Constructor (C)	Loading Savings Goals (TCOV-07-046)	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Click “+” Button (CAB)	N/A	Adding Savings Goal (TCOV-07-047)	?	?	?	?	?	?	?
Click “edit” Button (CEB)	N/A	Editing Savings Goal (TCOV-07-048)	?	?	?	?	?	?	?
Click “delete” Button (CDB)	N/A	Deleting Savings Goal (TCOV-07-049)	?	?	?	?	?	?	?
Validate Savings Goal (VSG)	N/A	?	Validating Savings Goal (TCOV-07-050)	Validating Savings Goal (TCOV-07-051)	?	?	?	?	?

Click "OK" Button (COB)	N/A	?	?	?	Savings Goal Deleted (TCOV-07-052)	?	?	?	?
Valid Amount Add (VAA)	N/A	?	?	?	?	Savings Goal Added (TCOV-07-053)	?	?	?
Valid Amount Edit (VAE)						Savings Goal Edited (TCOV-07-054)			
Invalid Amount Add (IA)	N/A	?	?	?	?	Adding Savings Goal (TCOV-07-055)	?	?	?
Invalid Amount Edit (IAE)						Editing Savings Goal (TCOV-07-056)			
Destructor (D)	N/A	?	?	?	?	?	Final (TCOV-07-057)	Final (TCOV-07-058)	Final (TCOV-07-059)

Table 2.2.7.4.1 Manage Savings State Table

Test Coverage ID	Test Coverage
TCOV-07-046	S1 to S2 with input C
TCOV-07-047	S2 to S3 with input CAB
TCOV-07-048	S2 to S4 with input CEB
TCOV-07-049	S2 to S5 with input CDB
TCOV-07-050	S3 to S6 with input VSG
TCOV-07-051	S4 to S6 with input VSG
TCOV-07-052	S5 to S9 with input COB
TCOV-07-053	S6 to S7 with input VAA
TCOV-07-054	S6 to S8 with input VAE
TCOV-07-055	S6 to S3 with input IAA
TCOV-07-056	S6 to S4 with input IAE
TCOV-07-057	S7 to Final with input D
TCOV-07-058	S8 to Final with input D
TCOV-07-059	S9 to Final with input D

Table 2.2.7.4.2 Manage Savings State Coverage

2.2.7.5 Use Case Testing

Use Case ID	UC-7	
Use Case	Add Savings Goal	
Purpose	The system shall allow actor to add savings goal	
Actor	User	
Trigger	Actor clicks on the “+” button on the “Savings” page	
Precondition	-	
Scenario Name	Step	Action
Main Flow	1	The system displays saving’s goal(s) of the actor
	2	User clicks on the “+” button on the Savings page
	3	System displays an Add Savings modal
	4	User enters Date, Goal Amount, Savings Amount, Category and Comment
	5	User clicks on the “Save” button
	6	System saves the Savings Goal in the backend of the system
	7	Savings page updates to reflect the added Savings Goal
Alternative Flow – Goal Amount is less than or equal to 0	4.1.1	User enters invalid Goal Amount values
	4.1.2	User clicks on the “Save” button
	4.1.3	Error message displayed “Goal should be between 1 and 80000”
	4.1.4	Back to Step 4
Alternative Flow – Goal Amount more than 80000	4.2.1	User enters invalid Goal Amount values
	4.2.2	User clicks on the “Save” button
	4.2.3	Error message displayed “Goal should be between 1 and 80000”
	4.2.4	Back to Step 4
Alternative Flow – Empty Goal Amount	4.3.1	User does not enter Goal Amount value
	4.3.2	User clicks on the “Save” button
	4.3.3	Error message displayed “Goal should be between 1 and 80000”
	4.3.4	Back to Step 4
Alternative Flow – Goal Amount is less than Savings Amount	4.4.1	User enters invalid Savings Amount values
	4.4.2	User clicks on the “Save” button
	4.4.3	Error message displayed “Goal Amount must be higher than Savings Amount”
	4.4.4	Back to Step 4
	4.5.1	User enters invalid date value
	4.5.2	User clicks on “Save” button

Alternative Flow – Date of Month <= 0	4.5.3	Error message displayed “Date should be valid”
	4.5.4	Back to Step 4
Alternative Flow – Date of Month > 31	4.6.1	User enters invalid date value
	4.6.2	User clicks on “Save” button
	4.6.3	Error message displayed “Date should be valid”
	4.6.4	Back to Step 4
Alternative Flow – Empty Comment	4.7.1	User does not enter any comment
	4.7.2	User clicks on “Save” button
	4.7.3	Error message displayed “Comment cannot be empty”
	4.7.4	Back to Step 4

Table 2.2.7.5.1 Manage Savings Use Case

Test Condition ID	Test Condition	Test Coverage ID	Test Coverage	Test Data
TCON-07-047	Main Flow	TCOV-07-059	Main Flow	Date: 24/06/2022 Goal Amount: 3200 Savings Amount: 500 Category: Other Comment: PS4
TCON-07-048	Alternative Flow – Goal Amount is less than or equal to 0	TCOV-07-060	Alternative Flow – Goal Amount is less than or equal to 0	Date: 24/06/2022 Goal Amount: 0 Savings Amount: 0 Category: Other Comment: PS4
TCON-07-049	Alternative Flow – Goal Amount more than 80000	TCOV-07-061	Alternative Flow – Goal Amount more than 8000	Date: 24/06/2022 Goal Amount: 90000 Savings Amount: 500 Category: Car Comment: BMW
TCON-07-050	Alternative Flow – Empty Goal Amount	TCOV-07-062	Alternative Flow – Empty Goal Amount	Date: 24/06/2022 Goal Amount: - Savings Amount: 0 Category: Food Comment: Grocery
TCON-07-051	Alternative Flow – Goal Amount	TCOV-07-063	Alternative Flow –	Date: 24/06/2022 Goal Amount: 2000

	is less than Savings Amount		Goal Amount is less than Savings Amount	Savings Amount: 3000 Category: Other Comment: iPhone
TCON-07-052	Alternative Flow – Date <= 0	TCOV-07-064	Alternative Flow – Date <= 0	Date: 0/06/2022 Goal Amount: 5000 Savings Amount: 3000 Category: Other Comment: iPhone
TCON-07-053	Alternative Flow – Date > 31	TCOV-07-065	Alternative Flow – Date > 31	Date: 32/06/2022 Goal Amount: 10000 Savings Amount: 3000 Category: Other Comment: iPhone
TCON-07-054	Alternative Flow – Empty Comment	TCOV-07-066	Alternative Flow – Empty Comment	Date: 32/06/2022 Goal Amount: 10000 Savings Amount: 3000 Category: Other Comment: -
TCON-07-055	Alternative Flow – Edit an existing Savings Goal	TCOV-07-067	Alternative Flow – Edit an existing Savings Goal	Date: 24/06/2022 Goal Amount: 4000 Savings Amount: 500 Category: Other Comment: PS4
TCON-07-056	Alternative Flow – Delete an existing Savings Goal	TCOV-07-068	Alternative Flow – Delete an existing Savings Goal	Press “delete” icon. Press “OK” button.

Table 2.2.7.5.2 Manage Savings Test Conditions & Coverage

2.2.8 F011 Edit Currency Setting

For this feature, we will be testing the select currency drop down input for the edit currency setting as this is the only type of input for this feature.

Five techniques are identified for designing test for this feature, including:

- i. Equivalence Partitioning
- ii. Boundary Value Analysis
- iii. State Transition Testing
- iv. Use Case Testing

Equivalence partitioning is used to select currency drop down into possible partitions for test. Boundary value analysis will test the boundary values of the select currency drop down input. There will be no decision table testing as it cannot be performed for the drop-down input provided by the system. State transition testing will test for major states of EM.

2.2.8.1 Equivalence Partitioning

Test Condition ID	Test Condition	Test Coverage ID	Test Coverage	Example of Test Data
TCON-08-001	Select Currency dropdown = US Dollars	TCOV-08-001	Select Currency dropdown = US Dollars (Valid)	US Dollars

Table 2.2.8.1.1 Edit Currency Setting Equivalence Partitioning Test Condition & Coverage

***Note:** The Select Currency dropdown contains options of [Indian Rupees, US Dollars, Pounds, Yen, Euro]

2.2.8.2 Boundary Value Analysis

Test Condition ID	Test Condition	Test Coverage ID	Test Coverage	Example of Test Data
TCON-08-002	Select Currency dropdown = Indian Rupees	TCOV-08-002	Select Currency dropdown = Indian Rupees	Indian Rupees
TCON-08-003	Select Currency dropdown = Euro	TCOV-08-003	Select Currency dropdown = Euro	Euro

Table 2.2.8.2.1 Edit Currency Setting Boundary Value Analysis Test Condition & Coverage

***Note:** These options were selected as they are the topmost and bottom most option in the dropdown.

2.2.8.3 State Transition Testing

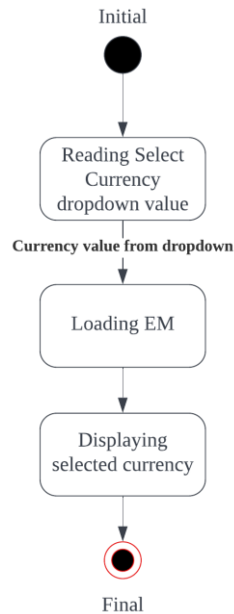


Figure 2.2.8.3.1 Edit Currency Setting Stats State Transition Diagram

The test condition for state transition testing is as following:

i.TCON-08-004 Edit Currency Setting state transition

Event/ State	Initial (S1)	Reading Select Currency dropdown value (S2)	Loading EM (S3)	Displaying selected currency (S4)
Constructor (C)	Reading Select Currency dropdown value (TCOV-08-004)	N/A	N/A	N/A
Currency value from dropdown (CV)	N/A	Loading EM (TCOV-08-005)		
Valid Currency (VC)	N/A		Displaying selected currency (TCOV-08-006)	
Destructor (D)				Final (TCOV-08-007)

Table 2.2.8.3.1 Edit Currency Setting State Table

Test Coverage ID	Test Coverage
TCOV-08-004	S1 to S2 with input C
TCOV-08-005	S2 to S3 with input CV
TCOV-08-006	S3 to S4 with input VC
TCOV-08-007	S4 to Final with input D

Table 2.2.8.3.2 Edit Currency Setting State Coverage

2.2.8.4 Use Case Testing

Use Case ID	UC-11	
Use Case	Edit Currency Setting	
Purpose	The system shall allow actor to edit their preferred currency setting	
Actor	User	
Trigger	Actor clicks the “Settings” button in the top navigation bar.	
Precondition	-	
Scenario Name	Step	Action
Main Flow	1	User clicks on the “Settings” button on the top navigation bar
	2	System navigates to the settings page
	3	User chooses their preferred currency settings from the Select Currency dropdown
	4	User clicks the “Save” button
	4	The system loads the Expense Manager for the new currency
	5	The system displays the selected currency to every page in the Expense Manager system.

Table 2.2.8.4.1 Edit Currency Setting Use Case

Test Condition ID	Test Condition	Test Coverage ID	Test Coverage	Test Data
TCON-08-005	Main Flow	TCOV-08-008	Main Flow	Pounds

Table 2.2.8.4.2 Edit Currency Setting Test Conditions & Coverage

2.3 Test Identification

Test Case ID	Description	Test Coverage ID	Test Condition ID
TC-01-001	Create new expense Main Flow	TCOV-01-003	TCON-01-003
		TCOV-01-005	TCON-01-005
		TCOV-01-005	TCON-01-009
		TCOV-01-009	TCON-01-011
		TCOV-01-011	TCON-01-014
		TCOV-01-014	TCON-01-015
		TCOV-01-015	TCON-01-016
		TCOV-01-016	TCON-01-017
		TCOV-01-017	TCON-01-020
		TCOV-01-020	TCON-01-021
		TCOV-01-021	TCON-01-022
		TCOV-01-022	TCON-01-023
		TCOV-01-023	TCON-01-024
		TCOV-01-024	TCON-01-025
		TCOV-01-025	TCON-01-028
		TCOV-01-028	TCON-01-029
		TCOV-01-029	TCON-01-030
		TCOV-01-030	TCON-01-031
		TCOV-01-031	TCON-01-032
		TCOV-01-032	TCON-01-033
		TCOV-01-033	TCON-01-034
		TCOV-01-038	TCON-01-036
TCOV-01-039	TCON-01-037		
TCOV-01-040	TCON-01-038		
TCOV-01-042			
TCOV-01-043			
TC-01-002	Create new expense Alternate Flow – Expense less than equal to 0	TCOV-01-001	TCON-01-001
		TCOV-01-012	TCON-01-012
		TCOV-01-033	TCON-01-035
		TCOV-01-034	TCON-01-037
		TCOV-01-035	TCON-01-039
		TCOV-01-039	
		TCOV-01-041	
TCOV-01-044			
TC-01-003	Create new expense Alternate Flow – Expense greater than 8000	TCOV-01-002	TCON-01-002
		TCOV-01-013	TCON-01-013
		TCOV-01-033	TCON-01-035
		TCOV-01-034	TCON-01-037

		TCOV-01-035	TCON-01-040
		TCOV-01-039	
		TCOV-01-041	
		TCOV-01-045	
TC-01-004	Create new expense Alternate Flow – Empty expens	TCOV-01-033	TCON-01-035
		TCOV-01-034	TCON-01-037
		TCOV-01-035	TCON-01-041
		TCOV-01-039	
		TCOV-01-041	
		TCOV-01-046	
TC-01-005	Create new expense Alternate Flow – Date of month <= 0	TCOV-01-004	TCON-01-004
		TCOV-01-018	TCON-01-018
		TCOV-01-033	TCON-01-035
		TCOV-01-034	TCON-01-037
		TCOV-01-036	TCON-01-042
		TCOV-01-039	
		TCOV-01-041	
		TCOV-01-047	
TC-01-006	Create new expense Alternate Flow – Date of month > 30	TCOV-01-006	TCON-01-006
		TCOV-01-019	TCON-01-019
		TCOV-01-033	TCON-01-035
		TCOV-01-034	TCON-01-037
		TCOV-01-036	TCON-01-043
		TCOV-01-039	
		TCOV-01-041	
		TCOV-01-048	
TC-01-007	Create new expense Alternate Flow – Invalid Comment	TCOV-01-008	TCON-01-008
		TCOV-01-010	TCON-01-010
		TCOV-01-026	TCON-01-026
		TCOV-01-027	TCON-01-028
		TCOV-01-033	TCON-01-035
		TCOV-01-035	TCON-01-037
		TCOV-01-036	TCON-01-044
		TCOV-01-039	
		TCOV-01-041	
		TCOV-01-049	

Table 2.3.1 Create New Expenses Test Identification

Test Case ID	Description	Test Coverage ID	Test Condition ID
TC-02-001	View Monthly Expenses Main Flow	TCOV-02-001	TCON-02-001
		TCOV-02-002	TCON-02-002
		TCOV-02-003	TCON-02-003
		TCOV-02-004	TCON-02-004
		TCOV-02-005	TCON-02-005
		TCOV-02-006	TCON-02-006
		TCOV-02-007	TCON-02-007
		TCOV-02-008	TCON-02-008
		TCOV-02-009	TCON-02-009
		TCOV-02-010	TCON-02-010
		TCOV-02-011	TCON-02-011
		TCOV-02-012	
		TCOV-02-013	
		TCOV-02-014	
		TCOV-02-015	TCON-02-012

Table 2.3.2 View Monthly Expenses Test Identification

Test Case ID	Description	Test Coverage ID	Test Condition ID
TC-03-001	View Daily Expenses Main Flow	TCOV-03-002	TCON-03-002
		TCOV-03-007	TCON-03-007
		TCOV-03-008	TCON-03-008
		TCOV-03-009	TCON-03-009
		TCOV-03-010	TCON-03-010
		TCOV-03-011	TCON-03-011
			TCON-03-013
		TCOV-03-013	TCON-03-014
		TCOV-03-014	
		TCOV-03-015	
		TCOV-03-017	
TCOV-03-018	TCON-03-015		
TC-03-002	View Daily Expenses Alternative Flow – Date != integer	TCOV-03-004	TCON-03-004
		TCOV-03-014	TCON-03-014
		TCOV-03-016	
		TCOV-03-019	TCON-03-016
TC-03-003	View Daily Expenses Alternative Flow – Date <= 0	TCOV-03-001	TCON-03-001
		TCOV-03-005	TCON-03-005
		TCOV-03-014	TCON-03-014
		TCOV-03-016	
		TCOV-03-020	TCON-03-017
TC-03-004	View Daily Expenses Alternative Flow – Date > 30	TCOV-03-003	TCON-03-003
		TCOV-03-006	TCON-03-006
		TCOV-03-014	TCON-03-014
		TCOV-03-016	
		TCOV-03-021	TCON-03-018
TC-03-005	Verify input date not null	TCOV-03-012	TCON-03-012

Table 2.3.3 View Daily Expenses Test Identification

Test Case ID	Description	Test Coverage ID	Test Condition ID
TC-04-001	View overall stats Main Flow	TCOV-04-001	TCON-04-001
		TCOV-04-004	TCON-04-004
		TCOV-04-005	TCON-04-005
		TCOV-04-010	TCON-04-010
		TCOV-04-011	TCON-04-011
		TCOV-04-012	
		TCOV-04-013	
		TCOV-04-014	
TC-04-002	View overall stats Alternative Flow – User enters year for the “bar chart” drop- down box	TCOV-04-002	TCON-04-002
		TCOV-04-006	TCON-04-006
		TCOV-04-007	TCON-04-007
		TCOV-04-010	TCON-04-010
		TCOV-04-011	TCON-04-012
		TCOV-04-012	
		TCOV-04-013	
		TCOV-04-015	
TC-04-003	View overall stats Alternative Flow – User enters year for the “each category” drop-down box	TCOV-04-003	TCON-04-003
		TCOV-04-008	TCON-04-008
		TCOV-04-009	TCON-04-009
		TCOV-04-010	TCON-04-010 TCON- 04-013
		TCOV-04-011	
		TCOV-04-012	
		TCOV-04-013	
		TCOV-04-016	

Table 2.3.4 View Overall Stats Test Identification

Test Case ID	Description	Test Coverage ID	Test Condition ID
TC-05-001	Filter Expenses Main Flow	TCOV-05-001	TCON-05-001
		TCOV-05-005	TCON-05-005
		TCOV-05-010	TCON-05-010
		TCOV-05-014	TCON-05-015
		TCOV-05-018	
		TCOV-05-019	
		TCOV-05-021	
		TCOV-05-022	
		TCOV-05-023	TCON-05-016
TC-05-002	Filter Expenses Alternative Flow – From Date is greater than To Date	TCOV-05-002	TCON-05-002
		TCOV-05-015	TCON-05-014
		TCOV-05-016	
		TCOV-05-017	
		TCOV-05-020	TCON-05-015
		TCOV-05-024	TCON-05-017
TC-05-003	Filter Expenses Alternative Flow – One or both date range input field is null	TCOV-05-003	TCON-05-003
		TCOV-05-004	TCON-05-004
		TCOV-05-020	TCON-05-015
		TCOV-05-025	TCON-05-018
TC-05-004	Filter Expenses Alternative Flow - From Expense is greater than To Expense	TCOV-05-006	TCON-05-006
		TCOV-05-015	
		TCOV-05-016	
		TCOV-05-017	
		TCOV-05-020	TCON-05-015
		TCOV-05-026	TCON-05-019
TC-05-005	Filter Expenses Alternative Flow - One or both expense range input fields are inputted a value less than or equal to zero	TCOV-05-011	TCON-05-011
		TCOV-05-012	TCON-05-012
		TCOV-05-013	TCON-05-013
		TCOV-05-015	TCON-05-014
		TCOV-05-016	
		TCOV-05-017	
		TCOV-05-020	TCON-05-015
		TCOV-05-027	TCON-05-020
TC-05-006	Filter Expenses Alternative Flow – One or both expense range input fields are inputted a value of string type	TCOV-05-007	TCON-05-007
		TCOV-05-008	TCON-05-008
		TCOV-05-020	TCON-05-015
		TCOV-05-028	TCON-05-021

Table 2.3.5 Filter Expenses Test Identification

Test Case ID	Description	Test Coverage ID	Test Condition ID
TC-06-001	Manage Loan Main Flow	TCOV-06-002	TCON-06-002
		TCOV-06-007	TCON-06-007
		TCOV-06-008	TCON-06-008
		TCOV-06-010	TCON-06-010
		TCOV-06-012	TCON-06-012
		TCOV-06-014	TCON-06-014
		TCOV-06-018	TCON-06-018
		TCOV-06-019	TCON-06-019
		TCOV-06-020	TCON-06-020
		TCOV-06-021	TCON-06-021
		TCOV-06-024	TCON-06-024
		TCOV-06-025	TCON-06-025
		TCOV-06-026	TCON-06-026
		TCOV-06-027	TCON-06-027
		TCOV-06-028	TCON-06-028
		TCOV-06-029	TCON-06-029
		TCOV-06-032	TCON-06-032
		TCOV-06-033	TCON-06-033
		TCOV-06-034	TCON-06-034
		TCOV-06-035	TCON-06-035
		TCOV-06-036	TCON-06-036
		TCOV-06-037	TCON-06-037
		TCOV-06-040	TCON-06-040
		TCOV-06-041	TCON-06-041
		TCOV-06-042	TCON-06-042
		TCOV-06-043	TCON-06-043
		TCOV-06-044	TCON-06-044
			TCON-06-045
			TCON-06-046
			TCON-06-047
			TCON-06-049
		TCOV-06-050	TCON-06-050
		TCOV-06-051	
TCOV-06-054			
TCOV-06-057			
TCOV-06-061			
TCOV-06-064	TCON-06-051		
TC-06-002	Manage Loan Alternative Flow – Edit an existing loan	TCOV-06-050	TCON-06-050
		TCOV-06-052	
		TCOV-06-055	

		TCOV-06-058	
		TCOV-06-062	
		TCOV-06-065	TCON-06-052
TC-06-003	Manage Loan Alternative Flow – Delete an existing loan	TCOV-06-050	TCON-06-050
		TCOV-06-053	
		TCOV-06-056	
		TCOV-06-063	
		TCOV-06-066	TCON-06-053
TC-06-004	Manage Loan Alternative Flow – Invalid date	TCOV-06-001	TCON-06-001
		TCOV-06-003	TCON-06-003
		TCOV-06-004	TCON-06-004
		TCOV-06-016	TCON-06-016
		TCOV-06-017	TCON-06-017
		TCOV-06-045	TCON-06-044
		TCOV-06-046	TCON-06-048
		TCOV-06-047	
		TCOV-06-048	
		TCOV-06-059	TCON-06-050
		TCOV-06-060	
		TCOV-06-067	TCON-06-054
		TC-06-005	Manage Loan Alternative Flow – Invalid amount
TCOV-06-006	TCON-06-006		
TCOV-06-022	TCON-06-022		
TCOV-06-023	TCON-06-023		
TCOV-06-045	TCON-06-045		
TCOV-06-046	TCON-06-048		
TCOV-06-047			
TCOV-06-049			
TCOV-06-059	TCON-06-050		
TCOV-06-060			
TCOV-06-068	TCON-06-055		
TC-06-006	Manage Loan Alternative Flow – Invalid person	TCOV-06-009	TCON-06-009
		TCOV-06-011	TCON-06-011
		TCOV-06-030	TCON-06-030
		TCOV-06-031	TCON-06-031
		TCOV-06-045	TCON-06-046
		TCOV-06-046	TCON-06-048
		TCOV-06-048	
		TCOV-06-049	
		TCOV-06-059	TCON-06-050
TCOV-06-060			

		TCOV-06-069	TCON-06-056
TC-06-007	Manage Loan Alternative Flow – Invalid reason	TCOV-06-013	TCON-06-013
		TCOV-06-015	TCON-06-015
		TCOV-06-038	TCON-06-038
		TCOV-06-039	TCON-06-039
		TCOV-06-045	TCON-06-047
		TCOV-06-047	TCON-06-048
		TCOV-06-048	
		TCOV-06-049	
		TCOV-06-059	TCON-06-050
		TCOV-06-060	
		TCOV-06-070	TCON-06-057

Table 2.3.6 Manage Loan Test Identification

Test Case ID	Description	Test Coverage ID	Test Condition ID
TC-07-001	Manage Savings Main Flow	TCOV-07-003	TCON-07-003
		TCOV-07-005	TCON-07-005
		TCOV-07-008	TCON-07-008
		TCOV-07-010	TCON-07-010
		TCOV-07-014	TCON-07-014
		TCOV-07-016	TCON-07-016
		TCOV-07-019	TCON-07-019
		TCOV-07-020	TCON-07-020
		TCOV-07-023	TCON-07-023
		TCOV-07-026	TCON-07-026
		TCOV-07-027	TCON-07-027
		TCOV-07-028	TCON-07-028
		TCOV-07-029	TCON-07-029
		TCOV-07-030	TCON-07-030
		TCOV-07-031	TCON-07-031
		TCOV-07-032	TCON-07-032
		TCOV-07-033	TCON-07-033
		TCOV-07-036	TCON-07-036
		TCOV-07-037	TCON-07-037
		TCOV-07-038	TCON-07-038
		TCOV-07-039	TCON-07-039
		TCOV-07-046	TCON-07-040
		TCOV-07-047	TCON-07-041
		TCOV-07-048	TCON-07-042
		TCOV-07-049	TCON-07-043
		TCOV-07-050	TCON-07-045
		TCOV-07-051	TCON-07-046
		TCOV-07-052	TCON-07-047
		TCOV-07-053	
TCOV-07-054			
TCOV-07-055			
TCOV-07-056			
TCOV-07-057			
TCOV-07-058			
TCOV-07-059			
TC-07-002	Alternative Flow – Goal Amount is less than or equal to 0	TCOV-07-001	TCON-07-001
		TCOV-07-017	TCON-07-017
		TCOV-07-040	TCON-07-044
		TCOV-07-041	TCON-07-046
		TCOV-07-042	TCON-07-048

		TCOV-07-043	
		TCOV-07-044	
		TCOV-07-055	
		TCOV-07-060	
TC-07-003	Alternative Flow – Goal Amount more than 80000	TCOV-07-002	TCON-07-002
		TCOV-07-018	TCON-07-018
		TCOV-07-040	
		TCOV-07-041	TCON-07-044
		TCOV-07-042	TCON-07-046
		TCOV-07-043	TCON-07-049
		TCOV-07-045	
		TCOV-07-055	
		TCOV-07-061	
TC-07-004	Alternative Flow – Empty Goal Amount	TCOV-07-040	TCON-07-044
		TCOV-07-041	TCON-07-046
		TCOV-07-042	TCON-07-050
		TCOV-07-043	
		TCOV-07-055	
		TCOV-07-062	
TC-07-005	Alternative Flow – Goal Amount is less than Savings Amount	TCOV-07-004	TCON-07-004
		TCOV-07-021	TCON-07-021
		TCOV-07-022	TCON-07-022
		TCOV-07-040	TCON-07-044
		TCOV-07-041	TCON-07-046
		TCOV-07-042	TCON-07-051
		TCOV-07-043	
		TCOV-07-045	
		TCOV-07-055	
		TCOV-07-063	
TC-07-006	Alternative Flow – Date <= 0	TCOV-07-007	TCON-07-007
		TCOV-07-040	TCON-07-044
		TCOV-07-041	TCON-07-046
		TCOV-07-043	TCON-07-051
		TCOV-07-044	TCON-07-052
		TCOV-07-045	
		TCOV-07-055	
		TCOV-07-064	
TC-07-007	Alternative Flow – Date > 31	TCOV-07-011	TCON-07-011
		TCOV-07-040	TCON-07-044
		TCOV-07-041	TCON-07-046
		TCOV-07-043	TCON-07-053
		TCOV-07-045	

		TCOV-07-055	
		TCOV-07-065	
TC-07-008	Alternative Flow – Empty Comment	TCOV-07-013	TCON-07-013
		TCOV-07-040	TCON-07-034
		TCOV-07-042	TCON-07-044
		TCOV-07-043	TCON-07-046
		TCOV-07-044	TCON-07-054
		TCOV-07-045	
		TCOV-07-055	
		TCOV-07-066	
TC-07-009	Alternative Flow – Edit an existing Savings Goal	TCOV-07-046	TCON-07-046
		TCOV-07-048	TCOV-07-067
		TCOV-07-051	
		TCOV-07-054	
		TCOV-07-056	
		TCOV-07-058	
		TCOV-07-067	
TC-07-010	Alternative Flow – Delete an existing Savings Goal	TCOV-07-046	TCON-07-046
		TCOV-07-049	TCOV-07-068
		TCOV-07-052	
		TCOV-07-059	

Table 2.3.7 Manage Savings Test Identification

Test Case ID	Description	Test Coverage ID	Test Condition ID
TC-08-001	Edit Currency Setting Main Flow	TCOV-08-001	TCON-08-001
		TCOV-08-002	TCON-08-002
		TCOV-08-003	TCON-08-003
		TCOV-08-004	TCON-08-004
		TCOV-08-005	TCON-08-005
		TCOV-08-006	
		TCOV-08-007	
		TCOV-08-008	

Table 2.3.8 Edit Currency Setting Test Identification

2.4 Feature Pass/Fail Criteria

Each feature must satisfy the following criteria in order to pass:

- i. Pass all its test cases
- ii. 0% calculation error if the feature involves monetary transaction
- iii. No deadlock
- iv. Fulfil its requirement

2.5 Test Deliverables

The following documents will be generated by the system test group and will be delivered after test completion.

Test documentation:

- i. Test Plan
- ii. Test Item Transmittal Report
- iii. Test Design Specifications
- iv. Test Case Specifications
- v. Test Procedure Specifications
- vi. Test Logs
- vii. Test Incident Reports
- viii. Test Summary Report
- ix. Test Completion Report

Test Case Specification

Expense Manager (EM)

Independent Verification and Validation

Version: 1.0.0

Date: 17/06/2022

1.0 Introduction

1.1 Purpose

This test case specification supports the following objective:

- i. To detail the test cases derived from test coverage in Test Design Specification.

1.2 References

The following document provide the test basis for this test design:

- i. EM Software Brief Description (SBD)

2.0 Test Cases

2.1 Environment

EM tests will be run using dummy accounts where the testing activities will not affect the production system. Tests are performed on the test team personal end devices with the following specification:

Operating System	Windows 10
Processor	Intel (64 bit)
Processor Core	2 or 4
RAM	8GB and above

Table 2.1.1 EM Testing Machine Requirements

2.2 Test Case

This section details all test cases designed separated according to feature.

2.2.1 F001 Add New Expenses

Test Case ID	TC-01-001		
Related Feature ID	F001		
Objective	Create new expense Main Flow		
Covered Test Coverage Items	TCOV-01-003, TCOV-01-005, TCOV-01-005, TCOV-01-009, TCOV-01-011, TCOV-01-014, TCOV-01-015, TCOV-01-016, TCOV-01-017, TCOV-01-020, TCOV-01-021, TCOV-01-022, TCOV-01-023, TCOV-01-024, TCOV-01-025, TCOV-01-028, TCOV-01-029, TCOV-01-030, TCOV-01-031, TCOV-01-032, TCOV-01-033, TCOV-01-038, TCOV-01-039, TCOV-01-040, TCOV-01-042, TCOV-01-043		
Input	Expected Result	Special Procedural Requirements	Intercase Dependency
Date = 24/05/2022	None	None	None
Expense = 500	None	None	None
Category = "Food"	None	None	None
Comment = "Grocery"	None	None	None
Click Save	Successfully saves expense into the system	None	None

Table 2.2.1.1 Create new expense Main Flow Test Case

Test Case ID	TC-01-002		
Related Feature ID	F001		
Objective	Create new expense Alternate Flow – Expense less than equal to 0		
Covered Test Coverage Items	TCOV-01-001, TCOV-01-012, TCOV-01-033, TCOV-01-034, TCOV-01-035, TCOV-01-039, TCOV-01-041, TCOV-01-044		
Input	Expected Result	Special Procedural Requirements	Intercase Dependency
Expense = -5	System displays error dialog “Expense must be between 0 and 8000”	None	None

Table 2.2.1.2 Create new expense Alternate Flow – Expense less than equal to 0 Test Case

Test Case ID	TC-01-003		
Related Feature ID	F001		
Objective	Create new expense Alternate Flow – Expense greater than 8000		
Covered Test Coverage Items	TCOV-01-002, TCOV-01-013, TCOV-01-033, TCOV-01-034, TCOV-01-035, TCOV-01-039, TCOV-01-041, TCOV-01-045		
Input	Expected Result	Special Procedural Requirements	Intercase Dependency
Expense = 10000	System displays error dialog “Expense must be between 0 and 8000”	None	None

Table 2.2.1.3 Create new expense Alternate Flow – Expense greater than 8000 Test Case

Test Case ID	TC-01-004		
Related Feature ID	F001		
Objective	Create new expense Alternate Flow – Empty expenses		
Covered Test Coverage Items	TCOV-01-033, TCOV-01-034, TCOV-01-035, TCOV-01-039, TCOV-01-041, TCOV-01-046		
Input	Expected Result	Special Procedural Requirements	Intercase Dependency
Expense = null	System displays error dialog “Expense must be between 0 and 8000”	None	None

Table 2.2.1.4 Create new expense Alternate Flow – Empty Expenses Test Case

Test Case ID	TC-01-005		
Related Feature ID	F001		
Objective	Create new expense Alternate Flow – Date of month <= 0		
Covered Test Coverage Items	TCOV-01-004, TCOV-01-018, TCOV-01-033, TCOV-01-034, TCOV-01-036, TCOV-01-039, TCOV-01-041, TCOV-01-047		
Input	Expected Result	Special Procedural Requirements	Intercase Dependency
Date = 0/06/2022	System displays error dialog “Date should be valid”	None	None

Table 2.2.1.5 Create new expense Alternate Flow – Date of month <= 0 Test Case

Test Case ID	TC-01-006		
Related Feature ID	F001		
Objective	Create new expense Alternate Flow – Date of month > 30		
Covered Test Coverage Items	TCOV-01-006, TCOV-01-019, TCOV-01-033, TCOV-01-034, TCOV-01-036, TCOV-01-039, TCOV-01-041, TCOV-01-048		
Input	Expected Result	Special Procedural Requirements	Intercase Dependency
Date = 31/06/2022	System displays error dialog “Date should be valid”	None	None

Table 2.2.1.6 Create new expense Alternate Flow – Date of month > 30 Test Case

Test Case ID	TC-01-007		
Related Feature ID	F001		
Objective	Create new expense Alternate Flow – Invalid Comment		
Covered Test Coverage Items	TCOV-01-008, TCOV-01-010, TCOV-01-026, TCOV-01-027, TCOV-01-033, TCOV-01-035, TCOV-01-036, TCOV-01-039, TCOV-01-041, TCOV-01-049		
Input	Expected Result	Special Procedural Requirements	Intercase Dependency
Comment = ""	System displays error dialog “Comment cannot be empty & should be less than 300 characters”	None	None

Table 2.2.1.7 Create new expense Alternate Flow – Invalid Comment Test Case

2.2.2 F002 View Monthly Expenses

Test Case ID	TC-02-001		
Related Feature ID	F002		
Objective	View Monthly Expenses Main Flow		
Covered Test Coverage Items	TCOV-02-001, TCOV-02-002, TCOV-02-003, TCOV-02-004, TCOV-02-005, TCOV-02-006, TCOV-02-007, TCOV-02-008, TCOV-02-009, TCOV-02-010, TCOV-02-011, TCOV-02-012, TCOV-02-013, TCOV-02-014, TCOV-02-015		
Input	Expected Result	Special Procedural Requirements	Intercase Dependency
Month = "June" Year = 2022	System displays the monthly expenses charts	None	None

Table 2.2.2.1 View Monthly Expenses Main Flow Test Case

2.2.3 F003 View Daily Expenses

Test Case ID	TC-03-001		
Related Feature ID	F003		
Objective	View Daily Expenses Main Flow		
Covered Test Coverage Items	TCOV-03-002, TCOV-03-007, TCOV-03-008, TCOV-03-009, TCOV-03-010, TCOV-03-011, TCOV-03-013, TCOV-03-014, TCOV-03-015, TCOV-03-017, TCOV-03-018		
Input	Expected Result	Special Procedural Requirements	Intercase Dependency
Date = 06/06/2022	System displays the doughnut chart for that date	None	None

Table 2.2.3.1 View Daily Expenses Main Flow Test Case

Test Case ID	TC-03-002		
Related Feature ID	F003		
Objective	View Daily Expenses Alternative Flow – Date != integer		
Covered Test Coverage Items	TCOV-03-004, TCOV-03-014, TCOV-03-016, TCOV-03-019		
Input	Expected Result	Special Procedural Requirements	Intercase Dependency
Date = “abc”	System displays error message “Date should be valid”	None	None

Table 2.2.3.2 View Daily Expenses Alternative Flow – Date != integer Test Case

Test Case ID	TC-03-003		
Related Feature ID	F003		
Objective	View Daily Expenses Alternative Flow – Date <= 0		
Covered Test Coverage Items	TCOV-03-001, TCOV-03-005, TCOV-03-014, TCOV-03-016, TCOV-03-020		
Input	Expected Result	Special Procedural Requirements	Intercase Dependency
Date = 0/06/2022	System displays error message “Date should be valid”	None	None

Table 2.2.3.3 View Daily Expenses Alternative Flow – Date <= 0 Test Case

Test Case ID	TC-03-004		
Related Feature ID	F003		
Objective	View Daily Expenses Alternative Flow – Date > 30		
Covered Test Coverage Items	TCOV-03-003, TCOV-03-006, TCOV-03-014, TCOV-03-016, TCOV-03-021		
Input	Expected Result	Special Procedural Requirements	Intercase Dependency
Date = 33/06/2022	System displays error message “Date should be valid”	None	None

Table 2.2.3.4 View Daily Expenses Alternative Flow – Date > 30 Test Case

Test Case ID	TC-03-005		
Related Feature ID	F003		
Objective	Verify input date not null		
Covered Test Coverage Items	TCOV-03-012		
Input	Expected Result	Special Procedural Requirements	Intercase Dependency
Date = null	System displays error message “Date should be valid”	None	None

Table 2.2.3.5 Verify input date not null Test Case

2.2.4 F004 View Overall Stats

Test Case ID	TC-04-001		
Related Feature ID	F004		
Objective	View overall stats Main Flow		
Covered Test Coverage Items	TCOV-04-001, TCOV-04-004, TCOV-04-005, TCOV-04-010, TCOV-04-011, TCOV-04-012, TCOV-04-013, TCOV-04-014		
Input	Expected Result	Special Procedural Requirements	Intercase Dependency
Year = 2022	System displays overall stats for the “doughnut chart” expense statistics	None	None

Table 2.2.4.1 View overall stats Main Flow Test Case

Test Case ID	TC-04-002		
Related Feature ID	F004		
Objective	View overall stats Alternative Flow – User enters year for the “bar chart” drop-down box		
Covered Test Coverage Items	TCOV-04-002, TCOV-04-006, TCOV-04-007, TCOV-04-010, TCOV-04-011, TCOV-04-012, TCOV-04-013, TCOV-04-015		
Input	Expected Result	Special Procedural Requirements	Intercase Dependency
Year = 2019	System displays overall stats for the “bar chart” expense statistics	None	None

Table 2.2.4.2 View overall stats Alternative Flow – User enters year for the “bar chart” drop-down box Test Case

Test Case ID	TC-04-003		
Related Feature ID	F004		
Objective	View overall stats Alternative Flow – User enters year for the “each category” drop-down box		
Covered Test Coverage Items	TCOV-04-003, TCOV-04-008, TCOV-04-009, TCOV-04-010, TCOV-04-011, TCOV-04-012, TCOV-04-013, TCOV-04-016		
Input	Expected Result	Special Procedural Requirements	Intercase Dependency
Year = 2020	System displays overall stats for the “each category” expense statistics	None	None

Table 2.2.4.3 View overall stats Alternative Flow – User enters year for the “each category” drop-down box Test Case

2.2.5 F005 Filter Expenses

Test Case ID	TC-05-001		
Related Feature ID	F005		
Objective	Filter Expenses Main Flow		
Covered Test Coverage Items	TCOV-05-001, TCOV-05-005, TCOV-05-010, TCOV-05-014, TCOV-05-018, TCOV-05-019, TCOV-05-021, TCOV-05-022, TCOV-05-023		
Input	Expected Result	Special Procedural Requirements	Intercase Dependency
From Date = 01/01/2000 To Date = 01/02/2000 From Expense = 1 To Expense = 2 Category = “Food”	System filters the expenses	None	None

Table 2.2.5.1 Filter Expenses Main Flow Test Case

Test Case ID	TC-05-002		
Related Feature ID	F005		
Objective	Filter Expenses Alternative Flow - From Date is greater than To Date		
Covered Test Coverage Items	TCOV-05-002, TCOV-05-015, TCOV-05-016, TCOV-05-017, TCOV-05-020, TCOV-05-024		
Input	Expected Result	Special Procedural Requirements	Intercase Dependency
From Date = 01/02/2000 To Date = 01/01/2000 From Expense = 1 To Expense = 2 Category = "Food"	System displays "From Date must be less than or equals to To Date" error message	None	None

Table 2.2.5.2 Filter Expenses Alternative Flow - From Date is greater than To Date Test Case

Test Case ID	TC-05-003		
Related Feature ID	F005		
Objective	Filter Expenses Alternative Flow – One or both date range input field is null		
Covered Test Coverage Items	TCOV-05-003, TCOV-05-004, TCOV-05-020, TCOV-05-025		
Input	Expected Result	Special Procedural Requirements	Intercase Dependency
From Date = 01/01/2000 To Date = null From Expense = 1 To Expense = 2 Category = "Food"	System displays "Date range inputs cannot be null" error message	None	None

Table 2.2.5.3 Filter Expenses Alternative Flow – One or both date range input field is null Test Case

Test Case ID	TC-05-004		
Related Feature ID	F005		
Objective	Filter Expenses Alternative Flow - From Expense is greater than To Expense		
Covered Test Coverage Items	TCOV-05-006, TCOV-05-015, TCOV-05-016, TCOV-05-017, TCOV-05-020, TCOV-05-026		
Input	Expected Result	Special Procedural Requirements	Intercase Dependency
From Date = 01/01/2000 To Date = 01/02/2000 From Expense = 2 To Expense = 1 Category = "Food"	System displays "From Expense must be less than or equals to To Expense" error message	None	None

Table 2.2.5.4 Filter Expenses Alternative Flow - From Expense is greater than To Expense Test Case

Test Case ID	TC-05-005		
Related Feature ID	F005		
Objective	Filter Expenses Alternative Flow - One or both expense range input fields are inputted a value less than or equal to zero		
Covered Test Coverage Items	TCOV-05-011, TCOV-05-012, TCOV-05-013, TCOV-05-015, TCOV-05-016, TCOV-05-017, TCOV-05-020, TCOV-05-027		
Input	Expected Result	Special Procedural Requirements	Intercase Dependency
From Date = 01/01/2000 To Date = 01/02/2000 From Expense = -3 To Expense = -5 Category = "Food"	System displays "Expense range inputs must be greater than zero" error message	None	None

Table 2.2.5.5 Filter Expenses Alternative Flow - One or both expense range input fields are inputted a value less than or equal to zero Test Case

Test Case ID	TC-05-006		
Related Feature ID	F005		
Objective	Filter Expenses Alternative Flow – One or both expense range input fields are inputted a value of string type		
Covered Test Coverage Items	TCOV-05-007, TCOV-05-008, TCOV-05-020, TCOV-05-028		
Input	Expected Result	Special Procedural Requirements	Intercase Dependency
From Date = 01/01/2000 To Date = 01/02/2000 From Expense = "j" To Expense = "k" Category = "Food"	System displays "Expense range inputs cannot be of string type" error message	None	None

Table 2.2.5.6 Filter Expenses Alternative Flow – One or both expense range input fields are inputted a value of string type Test Case

2.2.6 F006 Manage Loan

Test Case ID	TC-06-001		
Related Feature ID	F006		
Objective	Manage Loan Main Flow		
Covered Test Coverage Items	TCOV-06-002, TCOV-06-007, TCOV-06-008, TCOV-06-010, TCOV-06-012, TCOV-06-014, TCOV-06-018, TCOV-06-019, TCOV-06-020, TCOV-06-021, TCOV-06-024, TCOV-06-025, TCOV-06-026, TCOV-06-027, TCOV-06-028, TCOV-06-029, TCOV-06-032, TCOV-06-033, TCOV-06-034, TCOV-06-035, TCOV-06-036, TCOV-06-037, TCOV-06-040, TCOV-06-041, TCOV-06-042, TCOV-06-043, TCOV-06-044, TCOV-06-050, TCOV-06-051, TCOV-06-054, TCOV-06-057, TCOV-06-061, TCOV-06-064		
Input	Expected Result	Special Procedural Requirements	Intercase Dependency
Date: 07/06/2022 Amount: 200 Loan Type: Taken Person: Ahmad Status: Pending Reason: Medical Fees	The system saves the loan data into database and displays the data in the data table.	None	None

Table 2.2.6.1 Manage Loan Main Flow Test Case

Test Case ID	TC-06-002		
Related Feature ID	F006		
Objective	Manage Loan Alternative Flow – Edit an existing loan		
Covered Test Coverage Items	TCOV-06-050, TCOV-06-052, TCOV-06-055, TCOV-06-058, TCOV-06-062, TCOV-06-065		
Input	Expected Result	Special Procedural Requirements	Intercase Dependency
Date: 07/06/2022 Amount: 300 Loan Type: Taken Person: Ahmad Status: Pending Reason: Medical Fees	The system updates the loan data in the database and displays the data in the data table.	None	None

Table 2.2.6.2 Manage Loan Alternative Flow – Edit an existing loan Test Case

Test Case ID	TC-06-003		
Related Feature ID	F006		
Objective	Manage Loan Alternative Flow – Delete an existing loan		
Covered Test Coverage Items	TCOV-06-050, TCOV-06-053, TCOV-06-056, TCOV-06-063, TCOV-06-066		
Input	Expected Result	Special Procedural Requirements	Intercase Dependency
Press “delete” button. Press “OK” button.	The system deletes the loan data from the database and removes the data from the data table.	None	None

Table 2.2.6.3 Manage Loan Alternative Flow – Delete an existing loan Test Case

Test Case ID	TC-06-004		
Related Feature ID	F006		
Objective	Manage Loan Alternative Flow – Invalid date		
Covered Test Coverage Items	TCOV-06-001, TCOV-06-003, TCOV-06-004, TCOV-06-016, TCOV-06-017, TCOV-06-045, TCOV-06-046, TCOV-06-047, TCOV-06-048, TCOV-06-059, TCOV-06-060, TCOV-06-067		
Input	Expected Result	Special Procedural Requirements	Intercase Dependency
Date = 0/06/2022	System displays error message “Date should be valid”	None	None
Date = 33/06/2022	System displays error message “Date should be valid”	None	None
Date = “abc”	System displays error message “Date should be valid”	None	None
Date = null	System displays error message “Date should be valid”	None	None

Table 2.2.6.4 Manage Loan Alternative Flow – Invalid date Test Case

Test Case ID	TC-06-005		
Related Feature ID	F006		
Objective	Manage Loan Alternative Flow – Invalid amount		
Covered Test Coverage Items	TCOV-06-005, TCOV-06-006, TCOV-06-022, TCOV-06-023, TCOV-06-045, TCOV-06-046, TCOV-06-047, TCOV-06-049, TCOV-06-059, TCOV-06-060, TCOV-06-068		
Input	Expected Result	Special Procedural Requirements	Intercase Dependency
Amount = -5000	System displays error message “Amount must be between 0 and 100000”	None	None
Amount = 200000	System displays error message “Amount must be between 0 and 100000”	None	None
Amount = “abc”	System displays error message “Amount must be between 0 and 100000”	None	None
Amount = null	System displays error message “Amount must be between 0 and 100000”	None	None

Table 2.2.6.5 Manage Loan Alternative Flow – Invalid amount Test Case

Test Case ID	TC-06-006		
Related Feature ID	F006		
Objective	Manage Loan Alternative Flow – Invalid person		
Covered Test Coverage Items	TCOV-06-009, TCOV-06-011, TCOV-06-030, TCOV-06-031, TCOV-06-045, TCOV-06-046, TCOV-06-048, TCOV-06-049, TCOV-06-059, TCOV-06-060, TCOV-06-069		
Input	Expected Result	Special Procedural Requirements	Intercase Dependency
Person = null	System displays error message “Person name must be between 0 and 80 characters”	None	None
Person = “Siti Sarah Aqilah Rosaini Minah Binti Muhammad Abdullah Huhkum Kasim Karim Rahman”	System displays error message “Person name must be between 0 and 80 characters”	None	None
Person = “123”	System displays error message “Invalid person name”	None	None

Table 2.2.6.6 Manage Loan Alternative Flow – Invalid person Test Case

Test Case ID	TC-06-007		
Related Feature ID	F006		
Objective	Manage Loan Alternative Flow – Invalid reason		
Covered Test Coverage Items	TCOV-06-013, TCOV-06-015, TCOV-06-038, TCOV-06-039, TCOV-06-045, TCOV-06-047, TCOV-06-048, TCOV-06-049, TCOV-06-059, TCOV-06-060, TCOV-06-070		
Input	Expected Result	Special Procedural Requirements	Intercase Dependency
Reason = null	System displays error message “Reason must be between 0 and 300 characters”	None	None
Reason = “As a home owner I need to use personal loan to upgrade (renovate) my home and complete necessary repairs, like fixing the plumbing or redoing the electrical wiring. A personal loan is a good fit for me who don’t have equity in my home and don’t want to get a home equity line of credit or home equity loan”	System displays error message “Reason must be between 0 and 300 characters”	None	None

Table 2.2.6.7 Manage Loan Alternative Flow – Invalid reason Test Case

2.2.7 F007 Manage Savings

Test Case ID	TC-07-001		
Related Feature ID	F007		
Objective	Manage Savings Main Flow		
Covered Test Coverage Items	TCOV-07-003, TCOV-07-005, TCOV-07-008, TCOV-07-010, TCOV-07-014, TCOV-07-016, TCOV-07-019, TCOV-07-020, TCOV-07-023, TCOV-07-026, TCOV-07-027, TCOV-07-028, TCOV-07-029, TCOV-07-030, TCOV-07-031, TCOV-07-032, TCOV-07-033, TCOV-07-036, TCOV-07-037, TCOV-07-038, TCOV-07-039, TCOV-07-046, TCOV-07-047, TCOV-07-048, TCOV-07-049, TCOV-07-050, TCOV-07-051, TCOV-07-052, TCOV-07-053, TCOV-07-054, TCOV-07-055, TCOV-07-056, TCOV-07-057, TCOV-07-058, TCOV-07-059		
Input	Expected Result	Special Procedural Requirements	Intercase Dependency
Date: 24/06/2022 Goal Amount: 3200 Savings Amount: 500 Category: Other Comment: PS4	The system saves the savings goal data into database and displays the data in the data cards.	None	None

Table 2.2.7.1 Manage Savings Main Flow Test Case

Test Case ID	TC-07-002		
Related Feature ID	F007		
Objective	Manage Savings Alternative Flow – Goal Amount is less than or equal to 0		
Covered Test Coverage Items	TCOV-07-001, TCOV-07-017, TCOV-07-040, TCOV-07-041, TCOV-07-042, TCOV-07-043, TCOV-07-044, TCOV-07-055, TCOV-07-060		
Input	Expected Result	Special Procedural Requirements	Intercase Dependency
Date: 24/06/2022 Goal Amount: 0 Savings Amount: 0 Category: Other Comment: PS4	System displays error message “Goal Amount must be between 0 and 80000”	None	None

Table 2.2.7.2 Manage Savings Alternative Flow – Goal Amount is less than or equal to 0 Test Case

Test Case ID	TC-07-003		
Related Feature ID	F007		
Objective	Manage Savings Alternative Flow – Goal Amount more than 80000		
Covered Test Coverage Items	TCOV-07-002, TCOV-07-018, TCOV-07-040, TCOV-07-041, TCOV-07-042, TCOV-07-043, TCOV-07-045, TCOV-07-055, TCOV-07-061		
Input	Expected Result	Special Procedural Requirements	Intercase Dependency
Date: 24/06/2022 Goal Amount: 90000 Savings Amount: 500 Category: Car Comment: BMW	System displays error message “Goal Amount must be between 0 and 80000”	None	None

Table 2.2.7.3 Manage Savings Alternative Flow – Goal Amount more than 80000 Test Case

Test Case ID	TC-07-004		
Related Feature ID	F007		
Objective	Manage Savings Alternative Flow – Empty Goal Amount		
Covered Test Coverage Items	TCOV-07-040, TCOV-07-041, TCOV-07-042, TCOV-07-043, TCOV-07-055, TCOV-07-062		
Input	Expected Result	Special Procedural Requirements	Intercase Dependency
Date: 24/06/2022 Goal Amount: - Savings Amount: 0 Category: Food Comment: Grocery	System displays error message “Goal Amount is invalid”	None	None

Table 2.2.7.4 Manage Savings Alternative Flow – Empty Goal Amount Test Case

Test Case ID	TC-07-005		
Related Feature ID	F007		
Objective	Manage Savings Alternative Flow – Goal Amount is less than Savings Amount		
Covered Test Coverage Items	TCOV-07-004, TCOV-07-021, TCOV-07-022, TCOV-07-040, TCOV-07-041, TCOV-07-042, TCOV-07-043, TCOV-07-045, TCOV-07-055, TCOV-07-063		
Input	Expected Result	Special Procedural Requirements	Intercase Dependency
Date: 24/06/2022 Goal Amount: 2000 Savings Amount: 3000 Category: Other Comment: iPhone	System displays error message “Goal Amount must be more than Savings Amount”	None	None

Table 2.2.7.5 Manage Savings Alternative Flow – Goal Amount is less than Savings Amount Test Case

Test Case ID	TC-07-006		
Related Feature ID	F007		
Objective	Manage Savings Alternative Flow – Date <= 0		
Covered Test Coverage Items	TCOV-07-007, TCOV-07-040, TCOV-07-041, TCOV-07-043, TCOV-07-044, TCOV-07-045, TCOV-07-055, TCOV-07-064		
Input	Expected Result	Special Procedural Requirements	Intercase Dependency
Date: 0/06/2022 Goal Amount: 5000 Savings Amount: 3000 Category: Other Comment: iPhone	System displays error message “Date should be valid”	None	None

Table 2.2.7.6 Manage Savings Alternative Flow – Date <= 0 Test Case

Test Case ID	TC-07-007		
Related Feature ID	F007		
Objective	Manage Savings Alternative Flow – Date > 31		
Covered Test Coverage Items	TCOV-07-011, TCOV-07-040 TCOV-07-041, TCOV-07-043, TCOV-07-045, TCOV-07-055, TCOV-07-065		
Input	Expected Result	Special Procedural Requirements	Intercase Dependency
Date: 32/06/2022 Goal Amount: 10000 Savings Amount: 3000 Category: Other Comment: iPhone	System displays error message “Date should be valid”	None	None

Table 2.2.7.7 Manage Savings Alternative Flow – Date > 31 Test Case

Test Case ID	TC-07-008		
Related Feature ID	F007		
Objective	Manage Savings Alternative Flow – Empty Comment		
Covered Test Coverage Items	TCOV-07-013, TCOV-07-040, TCOV-07-042, TCOV-07-043, TCOV-07-044, TCOV-07-045, TCOV-07-055, TCOV-07-066		
Input	Expected Result	Special Procedural Requirements	Intercase Dependency
Date: 32/06/2022 Goal Amount: 10000 Savings Amount: 3000 Category: Other Comment: -	System displays error message “Comment must be between 0 and 300 characters”	None	None

Table 2.2.7.8 Manage Savings Alternative Flow – Empty Comment Test Case

Test Case ID	TC-07-009		
Related Feature ID	F007		
Objective	Manage Savings Alternative Flow – Edit an existing Savings Goal		
Covered Test Coverage Items	TCOV-07-046, TCOV-07-048, TCOV-07-051, TCOV-07-054, TCOV-07-056, TCOV-07-058, TCOV-07-067		
Input	Expected Result	Special Procedural Requirements	Intercase Dependency
Date: 24/06/2022 Goal Amount: 4000 Savings Amount: 500 Category: Other Comment: PS4	The system updates the savings goal data in the database and displays the data in the card table.	None	None

Table 2.2.7.9 Manage Savings Alternative Flow – Edit an existing Savings Goal Test Case

Test Case ID	TC-07-010		
Related Feature ID	F007		
Objective	Manage Savings Alternative Flow – Delete an existing Savings Goal		
Covered Test Coverage Items	TCOV-07-046, TCOV-07-049, TCOV-07-052, TCOV-07-059		
Input	Expected Result	Special Procedural Requirements	Intercase Dependency
Press “delete” icon. Press “OK” button.	The system deletes the savings goal data from the database and removes the data from the data cards.	None	None

Table 2.2.7.10 Manage Savings Alternative Flow – Delete an existing Savings Goal Test Case

2.2.8 F011 Edit Currency Setting

Test Case ID	TC-08-001		
Related Feature ID	F011		
Objective	Edit Currency Setting Main Flow		
Covered Test Coverage Items	TCOV-08-001, TCOV-08-002, TCOV-08-003, TCOV-08-004, TCOV-08-005, TCOV-08-006, TCOV-08-007, TCOV-08-008		
Input	Expected Result	Special Procedural Requirements	Intercase Dependency
Select currency drop down: Pound	The system displays the selected currency to every page in the Expense Manager system.	None	None

Table 2.2.8.1 Edit Currency Setting Main Flow Test Case

Test Procedure Specification

Expense Manager (EM)

Independent Verification and Validation

Version: 1.0.0

Date: 17/06/2022

1.0 Introduction

1.1 Purpose

This procedure describes the steps necessary to perform the test specified in the test design specification for EM. The procedure describes the execution of the test case described in Test Case Specification EM.

For test design specification and test cases, please refer to the following documents:

- i. EM Test Design Specification
- ii. EM Test Case Specification

1.2 Scope

The following features are covered in this procedure:

Function ID	Function	Function Description	Accessible Role
F001	Add new expenses	To allow user to perform add expenses for a particular date and saves it	User
F002	View monthly expenses	To allow the user to view a statistical view of their monthly expenditure	User
F003	View daily expenses	To allow the user to view a statistical view of their daily expenditure	User
F004	View overall stats	To allow the user to view a statistical view of their overall expenditure	User
F005	Filter expenses	To allow the user to perform filter expenses records by their specified criteria	User
F006	Manage loan	To allow the user to perform adding, editing and deleting of their loans.	User
F007	Manage savings	To allow the user to perform adding, editing and deleting of their saving's goals	User
F011	Edit currency setting	To allow the user to perform changes to the currency used	User

Table 1.2 EM Features

1.3 References

The following document provide the test basis for this test design:

- i. EM Software Brief Description (SBD)

2.0 Test Procedures

2.1 Create New Expense Test Procedure

Prior to the execution of the following test procedures, these special requirements must be prepared:

- i. User has the expense manager system on their computer
- ii. User has logged into the expense manager system

Test Procedure ID	TP-01-001
Objective	Create new expense Main Flow
Test Cases to be Executed	TC-01-001
Set up	<ol style="list-style-type: none">1. Click on the “+” button on the “Home” page.2. Input a valid date.3. Input “500” for expense.4. Input category “Food”.5. Input comment “Grocery”.6. Click save button.
Wrap Up	None

Table 2.1.1 Create new expense Main Flow Test Procedure

Test Procedure ID	TP-01-002
Objective	Create new expense Alternate Flow – Expense less than equal to 0, Expense greater than 8000, Empty expense
Test Cases to be Executed	TC-01-002, TC-01-003, TC-01-004
Set up	<ol style="list-style-type: none"> 1. Click on the “+” button on the “Home” page. 2. Input a valid date. 3. Input category “Food”. 4. Input comment “Grocery”. 5. Input “-50” for expense. [TC-01-002] 6. Click save button. 7. Repeat 1-4. 8. Input “10000” for expense. [TC-01-003] 9. Click save button. 10. Repeat 1-4. 11. Keep the expense text field empty. [TC-01-004] 12. Click save button
Wrap Up	None

Table 2.1.2 Create new expense Alternate Flow – Expense less than equal to 0, Expense greater than 8000, Empty expense Test Procedure

Test Procedure ID	TP-01-003
Objective	Create new expense Alternate Flow – Date of month <= 0, Date of month > 30
Test Cases to be Executed	TC-01-005, TC-01-006
Set up	<ol style="list-style-type: none"> 1. Click on the “+” button on the “Home” page 2. Input “500” for expense. 3. Input category “Food”. 4. Input comment “Grocery”. 5. Input date 0/06/2022. [TC-01-005] 6. Click save button. 7. Repeat 1-4. 8. Input date 31/06/2022. [TC-01-006] 9. Click save button
Wrap Up	None

Table 2.1.3 Create new expense Alternate Flow – Date of month <= 0, Date of month greater than 30 Test Procedure

Test Procedure ID	TP-01-004
Objective	Create new expense Alternate Flow – Invalid comment
Test Cases to be Executed	TC-01-007
Set up	<ol style="list-style-type: none"> 1. Click on the “+” button on the “Home” page. 2. Input date 22/06/2022. 3. Input “500” for expense. 4. Input category “Food”. 5. Keep the comment text field empty. [TC-01-007] 6. Click save button. 7. Repeat 1-4. 8. Enter a comment of length more than 300 characters. [TC-01-007] 9. Click save button
Wrap Up	None

Table 2.1.4 Create new expense Alternate Flow – Invalid comment Test Procedure

2.2 View Monthly Expenses Test Procedure

Test Procedure ID	TP-02-001
Objective	View Monthly Expenses Main Flow
Test Cases to be Executed	TC-02-001
Set up	<ol style="list-style-type: none"> 1. Select from the Month dropdown field the value: “June”, and select from the Year dropdown field the value: 2022 [TC-02-001]
Wrap Up	None

Table 2.2.1 View Monthly Expenses Main Flow Test Procedure

2.3 View Daily Expenses Test Procedure

Test Procedure ID	TP-03-001
Objective	View Daily Expenses Main Flow, View Daily Expenses Alternative Flow – Date != integer, View Daily Expenses Alternative Flow – Date <= 0, T View Daily Expenses Alternative Flow – Date > 30, Verify input date not null
Test Cases to be Executed	TC-03-001, TC-03-002, TC-03-003, TC-03-004, TC-03-005
Set up	<ol style="list-style-type: none"> 1. Input date 06/06/2022. [TC-03-001] 2. Input date “abc”. [TC-03-002] 3. Input date 0/06/2022. [TC-03-003] 4. Input date 33/06/2022. [TC-03-004] 5. Input nothing to input null date. [TC-03-005]
Wrap Up	None

Table 2.3.1 View Daily Expenses Main Flow, View Daily Expenses Alternative Flow – Date != integer, View Daily Expenses Alternative Flow – Date <= 0, T View Daily Expenses Alternative Flow – Date > 30, Verify input date not null Test Procedure

2.4 View Overall Stats Test Procedure

Prior to the execution of the following test procedures, these special requirements must be prepared:

- i. User has the expense manager system on their computer
- ii. User has logged into the expense manager system

Test Procedure ID	TP-04-001
Objective	View overall stats Main Flow, Alternate Flow - User enters year for the “bar chart” drop-down box, User enters year for the “each category” drop-down box
Test Cases to be Executed	TC-04-001, TC-04-002, TC-04-003
Set up	<ol style="list-style-type: none"> 1. Click on “Stat” from the top navigation bar. 2. Input 2022 from the year drop-down box beside in the “doughnut chart” section. [TC-04-001] 3. Input 2019 for the year drop-down box beside the “bar chart” section. [TC-04-002] 4. Input 2020 for the year drop-down box beside the “each category” section [TC-04-003]
Wrap Up	None

Table 2.4.1 View overall stats Main Flow, Alternate Flow - User enters year for the “bar chart” drop-down box, User enters year for the “each category” drop-down box

2.5 Filter Expenses Test Procedure

Test Procedure ID	TP-05-001
Objective	Filter Expenses Main Flow, Filter Expenses Alternative Flow – From Date is greater than To Date, Filter Expenses Alternative Flow – One or both date range input field is null, Filter Expenses Alternative Flow - From Expense is greater than To Expense, Filter Expenses Alternative Flow - One or both expense range input fields are inputted a value less than or equal to zero, Filter Expenses Alternative Flow – One or both expense range input fields are inputted a value of string type
Test Cases to be Executed	TC-05-001, TC-05-002, TC-05-03, TC-05-004, TC-05-005, TC-05-006
Set up	<ol style="list-style-type: none"> 1. Input into From Date field the value: “01/01/2000” 2. Input into To Date field the value: “01/02/2000” 3. Input into From Expense field the value: 1 4. Input into To Expense field the value: 2 5. Select from the Category dropdown list the value: “Food”
Remark	Repeat step 1, 2, 3, 4, 5, for different test cases with the input mentioned in respective test cases
Wrap Up	None

Table 2.5.1 Filter Expenses Main Flow, Filter Expenses Alternative Flow – From Date is greater than To Date, Filter Expenses Alternative Flow – One or both date range input field is null, Filter Expenses Alternative Flow - From Expense is greater than To Expense, Filter Expenses Alternative Flow - One or both expense range input fields are inputted a value less than or equal to zero, Filter Expenses Alternative Flow – One or both expense range input fields are inputted a value of string type Test Procedure

2.6 Manage Loan Test Procedure

Test Procedure ID	TP-06-001
Objective	Manage Loan Main Flow
Test Cases to be Executed	TC-06-001
Set up	<ol style="list-style-type: none"> 1. Click on the “+” button on the “Loan” page. 2. Input 07/06/2022 for Date. 3. Input 200 for Amount. 4. Choose Taken for Loan Type. 5. Input “Ahmad” for Person. 6. Choose Pending for Status. 7. Input “Medical Fees” for Reason. 8. Click save button.
Wrap Up	None

Table 2.6.1 Manage Loan Main Flow Test Procedure

Test Procedure ID	TP-06-002
Objective	Manage Loan Alternative Flow – Edit an existing loan
Test Cases to be Executed	TC-06-002
Set up	<ol style="list-style-type: none"> 1. Click on the “edit” button from specific loan on the “Loan” page. 2. Change the amount to 300. [TC-06-002] 3. Click save button.
Wrap Up	None

Table 2.6.2 Manage Loan Alternative Flow – Edit an existing loan Test Procedure

Test Procedure ID	TP-06-003
Objective	Manage Loan Alternative Flow – Delete an existing loan
Test Cases to be Executed	TC-06-003
Set up	<ol style="list-style-type: none"> 1. Click on the “delete” button from specific loan on the “Loan” page. 2. Click “OK” button. [TC-06-003]
Wrap Up	None

Table 2.6.3 Manage Loan Alternative Flow – Delete an existing loan Test Procedure

Test Procedure ID	TP-06-004
Objective	Manage Loan Alternative Flow – Invalid date, Manage Loan Alternative Flow – Invalid amount, Manage Loan Alternative Flow – Invalid person, Manage Loan Alternative Flow – Invalid reason
Test Cases to be Executed	TC-06-004, TC-06-005, TC-06-006, TC-06-007
Set up	<ol style="list-style-type: none"> 1. Click on the “+” button on the “Loan” page. 2. Input 0/06/2022 for invalid Date. [TC-06-004] 3. Input -5000 for invalid Amount. [TC-06-005] 4. Choose Taken for Loan Type. 5. Input “123” for invalid Person. [TC-06-006] 6. Choose Pending for Status. 7. Input nothing (null) for invalid Reason. [TC-06-007] 8. Click save button.
Remark	Repeat step 2, 3, 5 and 7 for different inputs as mentioned in the respective test cases.
Wrap Up	None

Table 2.6.4 Manage Loan Alternative Flow – Invalid date, Manage Loan Alternative Flow – Invalid amount, Manage Loan Alternative Flow – Invalid person, Manage Loan Alternative Flow – Invalid reason Test Procedure

2.7 Manage Savings Test Procedure

Test Procedure ID	TP-07-001
Objective	Manage Savings Main Flow
Test Cases to be Executed	TC-07-001
Set up	<ol style="list-style-type: none"> 1. Click on the “+” button on the “Savings” page. 2. Input 24/06/2022 for Date. 3. Input 3200 for Goal Amount. 4. Input 500 for Savings Amount. 5. Choose “Other” for Category 6. Input “PS4” in Comment 7. Click save button.
Wrap Up	None

Table 2.7.1 Manage Savings Main Flow Test Procedure

Test Procedure ID	TP-07-002
Objective	Manage Savings Alternative Flow – Goal Amount is less than or equal to 0 Manage Savings Alternative Flow – Goal Amount is less than or equal to 0, Manage Savings Alternative Flow – Goal Amount more than 80000, Manage Savings Alternative Flow – Empty Goal Amount
Test Cases to be Executed	TC-07-002, TC-07-003, TC-07-004
Set up	<ol style="list-style-type: none"> 1. Input 0 in Goal Amount for Invalid Amount [TC-07-002] 2. Input 90000 in Goal Amount for Invalid Amount [TC-07-003] 3. Input nothing(null) in Goal Amount for Invalid Amount [TC-07-004]
Wrap Up	None

Table 2.7.2 Manage Savings Alternative Flow – Goal Amount is less than or equal to 0, Manage Savings Alternative Flow – Goal Amount more than 80000, Manage Savings Alternative Flow – Empty Goal Amount Procedure

Test Procedure ID	TP-07-003
Objective	Manage Savings Alternative Flow – Goal Amount is less than Savings Amount
Test Cases to be Executed	TC-07-005
Set up	1. Input 1000 in Goal Amount 2. Input 3000 in Savings Amount for Invalid Amount [TC-07-005]
Wrap Up	None

Table 2.7.3 Manage Savings Alternative Flow – Goal Amount is less than Savings Amount Procedure

Test Procedure ID	TP-07-004
Objective	Manage Savings Alternative Flow – Date <= 0, Manage Savings Alternative Flow – Date > 31
Test Cases to be Executed	TC-07-006, TC-07-007
Set up	1. Input Date 0/06/2022. [TC-07-006] 2. Input Date 33/06/2022. [TC-07-007]
Wrap Up	None

Table 2.7.4 Manage Savings Alternative Flow – Date <= 0, Manage Savings Alternative Flow – Date > 31 Test Procedure

Test Procedure ID	TP-07-005
Objective	Manage Savings Alternative Flow – Empty Comment
Test Cases to be Executed	TC-07-008
Set up	1. Input nothing (null) in Comment [TC-07-008]
Wrap Up	None

Table 2.7.5 Manage Savings Manage Savings Alternative Flow – Empty Comment Procedure

Test Procedure ID	TP-07-006
Objective	Manage Savings Alternative Flow – Edit an existing Savings Goal
Test Cases to be Executed	TC-07-009
Set up	<ol style="list-style-type: none"> 2. Click on the “edit” button from specific Savings Goal on the “Savings” page. 3. Change the Goal Amount to 4000. [TC-07-009] 4. Click save button.
Wrap Up	None

Table 2.7.6 Manage Savings Alternative Flow – Edit an existing Savings Goal Procedure

Test Procedure ID	TP-07-007
Objective	Manage Loan Alternative Flow – Delete an existing Savings Goal
Test Cases to be Executed	TC-07-010
Set up	<ol style="list-style-type: none"> 3. Click on the “delete” icon from specific Savings Goal on the “Savings” page. 4. Click “OK” button. [TC-07-010]
Wrap Up	None

Table 2.7.7 Manage Savings Alternative Flow – Delete an existing Savings Goal Test Procedure

2.8 Edit Currency Setting Test Procedure

Test Procedure ID	TP-08-001
Objective	Edit Currency Setting Main Flow
Test Cases to be Executed	TC-08-001
Set up	<ol style="list-style-type: none"> 1. Click on the “+” button on the “Loan” page. 2. Input Pound for Select Currency dropdown [TC-08-001] 3. Click save button.
Wrap Up	None

Table 2.8.1 Edit Currency Setting Main Flow Test Procedure

Test Log

Expense Manager (EM)

Independent Verification and Validation

Version: 1.0.0

Date: 17/06/2022

General Information			
Test Log Scope	This test log covers Add New Expense (F001), View Monthly Expenses (F002), View Daily Expenses (F003), View Overall Stats (F004), Filter Expenses (F005), Manage Loan (F006), Manage Savings (F007), Edit Currency Setting (F011) as described in Test Plan, EM_TP_1.0.		
Test Log Description	The item tested was the add new expense feature, viewing monthly expenses feature, viewing daily expenses feature, viewing overall stats feature, filtering expenses feature, managing loan feature, managing savings feature, and editing currency settings feature in EM 1.0. The test log records the execution of the test procedure (EM_TPS_1.0.0).		
Revision Version	1.0	People Responsible	<ol style="list-style-type: none"> 1. Kisyok Indran A/L Veerasamy 2. Muhammad Irfan bin Nasruddin 3. Muhammad Hafiz bin Baharudin 4. Fahad Aqeel Thakur
Activities Execution Information			
Execution Start Date	04/06/2022	End Date	11/06/2022

Procedure Result							
Requirement ID	Test Cases ID	Test Procedure ID	Type of Testing	Tool	Pass / Fail	Test Incident Report ID / Test Incident ID	Remark
F001 Create new expense	TC-01-001	TP-01-001	Functional	Manual	Pass	-	-
	TC-01-002	TP-01-002			Pass	-	-
	TC-01-003	TP-01-003			Fail	TIR-01-001	No error message "Expense should be between 0 & 8000"
	TC-01-004				Pass	-	-
	TC-01-005	TP-01-003			Fail	TIR-01-002	No error message "Date should be valid"
	TC-01-006				Fail	TIR-01-003	No error message "Date should be valid"
	TC-01-007	TP-01-004			Pass	-	-
F002 View Monthly Expenses	TC-02-001	TP-02-001	Functional	Manual	Pass	-	-
F003 View Daily Expenses	TC-03-001	TP-03-001	Functional	Manual	Pass	-	-
	TC-03-002				Fail	TIR-03-001	No error message "Date should be valid".
	TC-03-003				Fail	TIR-03-002	No error message "Date should be valid".

	TC-03-004				Fail	TIR-03-003	No error message "Date should be valid".
	TC-03-005				Fail	TIR-03-004	No error message "Date should be valid". Page becomes blank.
F004 View overall stats	TC-04-001	TP-04-001	Functional	Manual	Pass	-	-
	TC-04-002				Pass	-	-
	TC-04-003				Pass	-	-
F005 Filter Expenses	TC-05-001	TP-05-001	Functional	Manual	Pass	-	-
	TC-05-002				Fail	TIR-05-001	No error message "From Date must be less than or equals to To Date".
	TC-05-003				Fail	TIR-05-002	No error message "Date range inputs cannot be null". Page becomes blank.
	TC-05-004				Fail	TIR-05-003	No error message "From Expense must be less than or equals to To Expense".
	TC-05-005				Fail	TIR-05-004	No error message "Expense range inputs must be greater than zero".
	TC-05-006				Fail	TIR-05-005	No error message "Expense range inputs cannot be of string type". Page becomes blank.

F006 Manage Loan	TC-06-001	TP-06-001	Functional	Manual	Pass	-	-
	TC-06-002	TP-06-002	Functional	Manual	Pass	-	-
	TC-06-003	TP-06-003	Functional	Manual	Pass	-	-
	TC-06-004	TP-06-004	Functional	Manual	Fail	TIR-06-001	No error message "Date should be valid". System saves the current date.
	TC-06-005				Fail	TIR-06-002	No error message "Amount must be between 0 and 100000". System saves the amount.
	TC-06-006				Fail	TIR-06-003	No error message "Person name must be between 0 and 80 characters" or "Invalid person name". System saves the person's name.
	TC-06-007				Fail	TIR-06-004	No error message "Reason must be between 0 and 300 characters". System saves the reason.
F007 Manage Savings	TC-07-001	TP-07-001	Functional	Manual	Pass	-	-
	TC-07-002	TP-07-002	Functional	Manual	Pass	-	-
	TC-07-003		Functional	Manual	Fail	TIR-07-001	No error message "Goal Amount should be between 0 & 80000"

	TC-07-004		Functional	Manual	Pass	-	-
	TC-07-005	TP-07-003	Functional	Manual	Fail	TIR-007-002	No error message "Goal Amount should be greater than Savings Amount"
	TC-07-006	TP-07-004	Functional	Manual	Fail	TIR-007-003	No error message "Date should be valid".
	TC-07-007		Functional	Manual	Fail	TIR-007-004	No error message "Date should be valid".
	TC-07-008	TP-07-005	Functional	Manual	Pass	-	-
	TC-07-009	TP-07-006	Functional	Manual	Pass	-	-
	TC-07-010	TP-07-007	Functional	Manual	Pass	-	-
F011 Edit Currency Setting	TC-08-001	TP-08-001	Functional	Manual	Pass	-	-
Environment Information							
Requested Test Environment				Test Environment After Changes			
Software: Expense Manager Version 1.0, Windows 11, Microsoft Edge, NodeJS 18.3.0 Hardware: Intel 64-bit Processor Reference: Test Plan				No Change			
Anomalous Event							
Unexpected event occurred		Test Procedure ID		Anomaly Reporter Name			

No error message "Expense should be between 0 & 8000"	TP-01-002	Fahad Aqeel Thakur
No error message "Date should be valid"	TP-01-003	Fahad Aqeel Thakur
No error message "Date should be valid"	TP-01-003	Fahad Aqeel Thakur
No display message "Date should be valid".	TP-03-001	Kisyok Indran A/L Veerasamy
No display message "Date should be valid".	TP-03-001	Kisyok Indran A/L Veerasamy
No display message "Date should be valid".	TP-03-001	Kisyok Indran A/L Veerasamy
No display message "Date should be valid". Page becomes blank	TP-03-001	Kisyok Indran A/L Veerasamy
No error message "From Date must be less than or equals to To Date".	TP-05-001	Muhammad Irfan bin Nasruddin
No error message "Date range inputs cannot be null". Page becomes blank.	TP-05-001	Muhammad Irfan bin Nasruddin
No error message "From Expense must be less than or equals to To Expense".	TP-05-001	Muhammad Irfan bin Nasruddin
No error message "Expense range inputs must be greater than zero".	TP-05-001	Muhammad Irfan bin Nasruddin

No error message "Expense range inputs cannot be of string type". Page becomes blank.	TP-05-001	Muhammad Irfan bin Nasruddin
No error message "Date should be valid". System saves the current date.	TP-06-004	Kisyok Indran A/L Veerasamy
No error message "Amount must be between 0 and 100000". System saves the amount.	TP-06-004	Kisyok Indran A/L Veerasamy
No error message "Person name must be between 0 and 80 characters" or "Invalid person name". System saves the person's name.	TP-06-004	Kisyok Indran A/L Veerasamy
No error message "Reason must be between 0 and 300 characters". System saves the reason.	TP-06-004	Kisyok Indran A/L Veerasamy
No error message "Goal Amount should be between 0 & 80000"	TP-07-002	Muhammad Hafiz bin Baharudin
No error message "Goal Amount should be greater than Savings Amount"	TP-07-003	Muhammad Hafiz bin Baharudin
No error message "Date should be valid".	TP-07-004	Muhammad Hafiz bin Baharudin

No error message "Date should be valid".	TP-07-004	Muhammad Hafiz bin Baharudin
--	-----------	------------------------------

Test Incident Report

Expense Manager (EM)



Independent Verification and Validation

Version: 1.0.0

Date: 17/06/2022

Document Control

Document Name	EM Test Incident Report (Iteration 1)
Reference Number	EM_TIR_1
Version	1.0.0
Project Code	EM
Status	In-use
Date Released	17 June 2022

Name	Position	Signature
Prepared by: Muhammad Irfan bin Nasruddin Muhammad Hafiz bin Baharudin Fahad Aqeel Thakur	Test Analyst EM Test Team	
Reviewed by: Kisyok Indran A/L Veerasamy	Test Lead EM Test Team	
Verified by: Dr. Siti Hafizah Binti Ab Hamid	Test Manager EM Test Team	

Version History

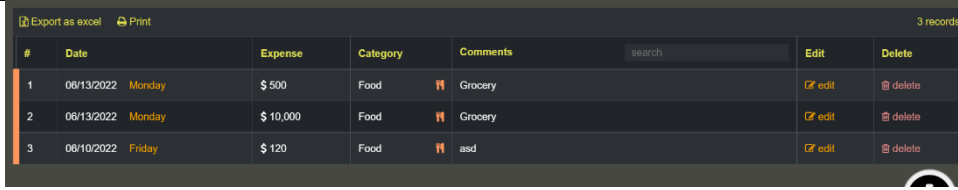
Version	Release Date	Section	Amendments
1.0	17/6/2022	All	Original Document

Distribution List

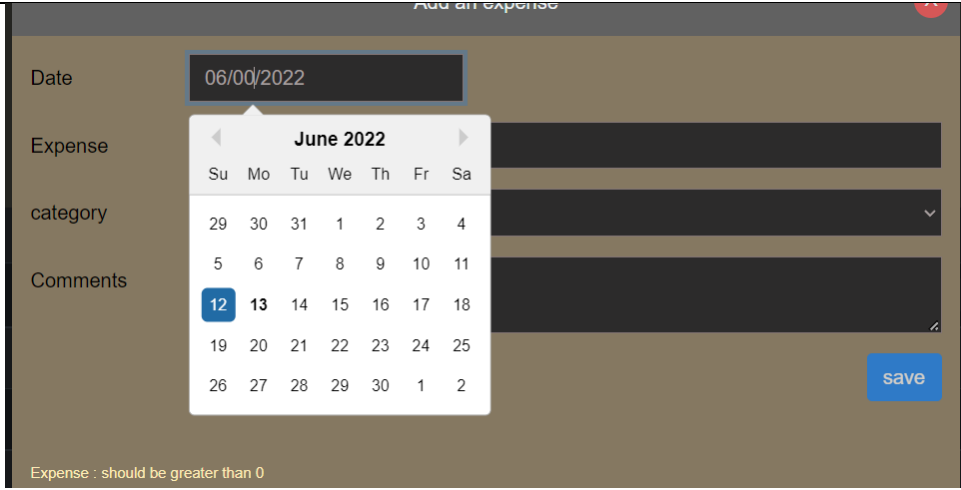
Version	Release Date	Controlled Copy No	Recipient Name	Department	Issue Date	Return Date
1.0.0	17/6/2022	01		EM Test Team	17/6/2022	

Test Incident Report

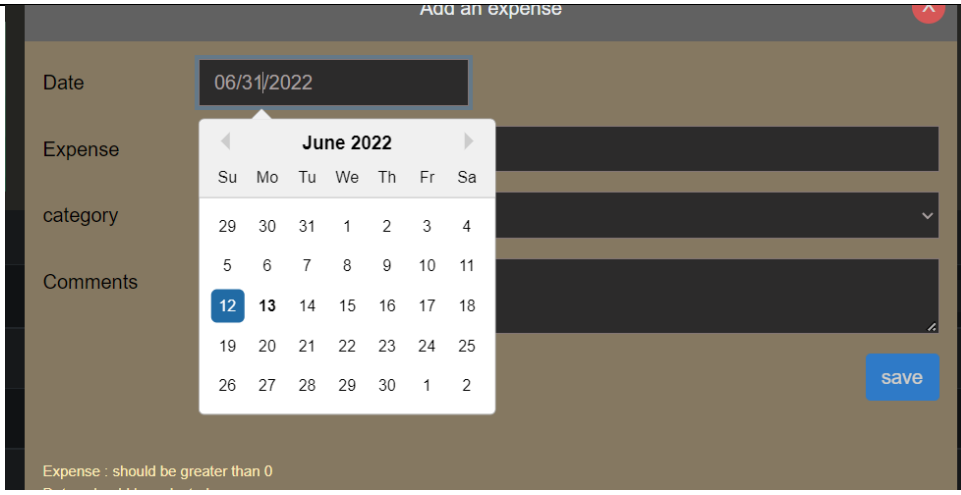
Incident Report Identifier (Table 1)	
Scope	The scope of this Test Incident Report is on any incident occurred based on Test Procedure executed on Expense Manager (ST_EM_1.0) v1.0 which took place during Test Execution only.
References	EM_TPS_1_1.0.0 EM_TL_1_1.0.0

Test Incident Number	TIR-01-001
Summary	
The system does not display error message when expense is greater than 8000	
Date and Time Incident	12/06/2022 (4:53 PM)
Context	Expense Manager (EM_1.1)
Description of Incident	
Test Procedure	TP-01-002
Test Data	Expense = 10000
Expected Result	Display error message "Expense should be between 0 & 8000"
Actual Result	System displays the expense amount. System does not display error message.
Unexpected Outcome	
Procedure to reproduce the incident	Input expense = 10000 [TC-01-003]
Test Environment	Hardware: Windows 10, Intel (64 bit), 8GB RAM Reference: EM_TP_1. O
Attempt to repeat	Procedure done 3 times after incident first emerged. Same result each time.
Tester's Name	Fahad Aqeel Thakur
Observer's Name (witness)	Kisyok Indran A/L Veerasamy
Status of Incident	

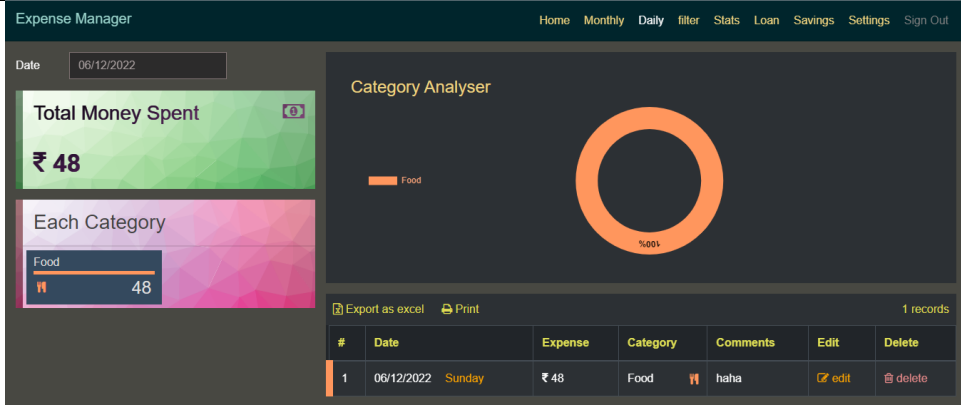
Open
Impact
Major – Severe problems but possible to work around
Priority
Immediate – Must be fixed as soon as possible
Description of the corrective action
To discuss with the developer on fixing the Create New Expenses feature. Developer shall fix the system to display error message as mentioned.

Test Incident Number	TIR-01-002
Summary	
The system does not display error message date should be valid	
Date and Time Incident	12/06/2022 (5:02 PM)
Context	Expense Manager (EM_1.1)
Description of Incident	
Test Procedure	TP-01-003
Test Data	date = 0/06/2022
Expected Result	Display error message "Date should be valid"
Actual Result	No error message shown "Date should be valid"
Unexpected Outcome	
Procedure to reproduce the incident	Input date = 0/06/2022 [TC-01-005]
Test Environment	Hardware: Windows 10, Intel (64 bit), 8GB RAM Reference: EM_TP_1. O
Attempt to repeat	Procedure done 3 times after incident first emerged. Same result each time.
Tester's Name	Fahad Aqeel Thakur
Observer's Name (witness)	Muhammad Hafiz bin Baharudin
Status of Incident	
Open	
Impact	
Major – Severe problems but possible to work around	

Priority
Immediate – Must be fixed as soon as possible
Description of the corrective action
To discuss with the developer on fixing the Create New Expenses feature. Developer shall fix the system to display error message as mentioned.

Test Incident Number	TIR-01-003
Summary	
The system does not display error message date should be valid	
Date and Time Incident	12/06/2022 (5:07 PM)
Context	Expense Manager (EM_1.1)
Description of Incident	
Test Procedure	TP-01-003
Test Data	date = 31/06/2022
Expected Result	Display error message "Date should be valid"
Actual Result	No error message shown "Date should be valid"
Unexpected Outcome	
Procedure to reproduce the incident	Input date = 31/06/2022 [TC-01-006]
Test Environment	Hardware: Windows 10, Intel (64 bit), 8GB RAM Reference: EM_TP_1. O
Attempt to repeat	Procedure done 3 times after incident first emerged. Same result each time.
Tester's Name	Fahad Aqeel Thakur
Observer's Name (witness)	Muhammad Irfan bin Nasruddin
Status of Incident	
Open	
Impact	
Major – Severe problems but possible to work around	

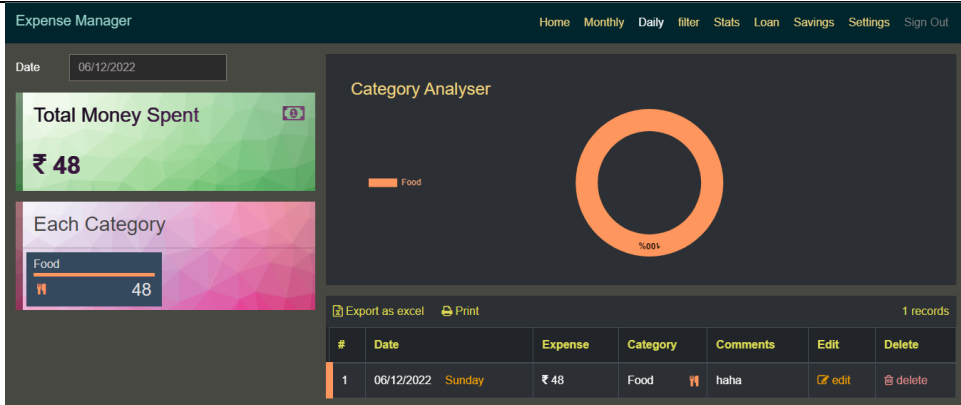
Priority
Immediate – Must be fixed as soon as possible
Description of the corrective action
To discuss with the developer on fixing the Create New Expenses feature. Developer shall fix the system to display error message as mentioned.

Test Incident Number	TIR-03-001
Summary	
The system does not display “Date should be valid” when the date is not an integer.	
Date and Time Incident	12/06/2022
Context	Expense Manager (EM_1.1)
Description of Incident	
Test Procedure	TP-03-001
Test Data	Input date “abc”
Expected Result	System displays error dialog “Date should be valid”
Actual Result	System displays the current date. System does not display error message.
Unexpected Outcome	
Procedure to reproduce the incident	Input date “abc”. [TC-03-002]
Test Environment	Hardware: Windows 10, Intel (64 bit), 8GB RAM Reference: EM_TP_1.0
Attempt to repeat	Procedure done 3 times continuously after incident 1 st discovered. Same result emerged.
Tester’s Name	Kisyok Indran A/L Veerasamy
Observer’s Name (witness)	Fahad Aqeel Thakur
Status of Incident	
Open	
Impact	
Major – Sever problems but possible to work around	
Priority	

Delayed – System is unstable but incident must be fixed prior to next level of test or shipment

Description of the corrective action

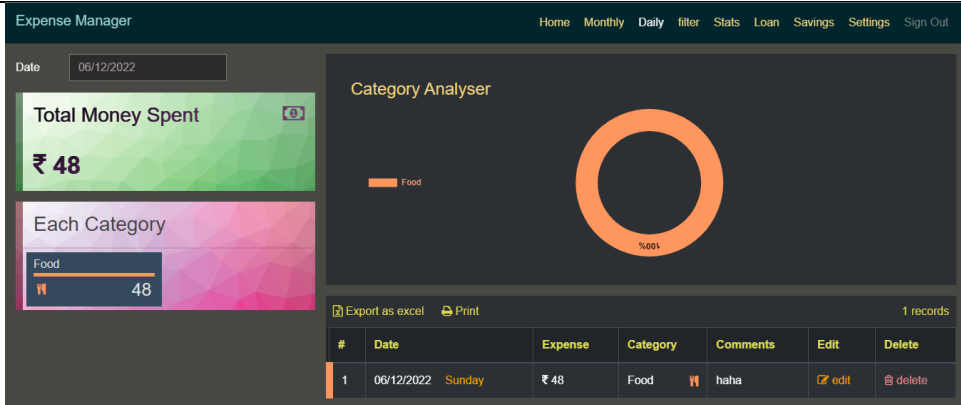
To discuss with the developer on fixing the View Daily Expenses feature.
Developer shall fix the system to display error message as mentioned.

Test Incident Number	TIR-03-002
Summary	
The system does not display “Date should be valid” when the date is less than or equals to zero.	
Date and Time Incident	12/06/2022
Context	Expense Manager (EM_1.1)
Description of Incident	
Test Procedure	TP-03-001
Test Data	Input date 0/06/2022
Expected Result	System displays error dialog “Date should be valid”
Actual Result	System displays the current date. System does not display error message.
Unexpected Outcome	
Procedure to reproduce the incident	Input date 0/06/2022. [TC-03-003]
Test Environment	Hardware: Windows 10, Intel (64 bit), 8GB RAM Reference: EM_TP_1.0
Attempt to repeat	Procedure done 3 times continuously after incident 1 st discovered. Same result emerged.
Tester's Name	Kisyok Indran A/L Veerasamy
Observer's Name (witness)	Fahad Aqeel Thakur
Status of Incident	
Open	
Impact	
Major – Sever problems but possible to work around	
Priority	

Delayed – System is unstable but incident must be fixed prior to next level of test or shipment

Description of the corrective action

To discuss with the developer on fixing the View Daily Expenses feature.
Developer shall fix the system to display error message as mentioned.

Test Incident Number	TIR-03-003
Summary	
The system does not display “Date should be valid” when the date is more than 30.	
Date and Time Incident	12/06/2022
Context	Expense Manager (EM_1.1)
Description of Incident	
Test Procedure	TP-03-001
Test Data	Input date 33/06/2022
Expected Result	System displays error dialog “Date should be valid”
Actual Result	System displays the current date. System does not display error message.
Unexpected Outcome	
Procedure to reproduce the incident	Input date 33/06.2022. [TC-03-004]
Test Environment	Hardware: Windows 10, Intel (64 bit), 8GB RAM Reference: EM_TP_1.0
Attempt to repeat	Procedure done 3 times continuously after incident 1 st discovered. Same result emerged.
Tester's Name	Kisyok Indran A/L Veerasamy
Observer's Name (witness)	Fahad Aqeel Thakur
Status of Incident	
Open	
Impact	
Major – Sever problems but possible to work around	
Priority	

Delayed – System is unstable but incident must be fixed prior to next level of test or shipment

Description of the corrective action

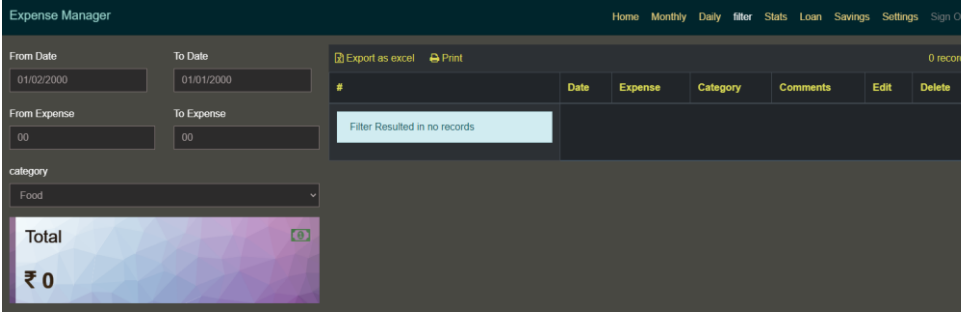
To discuss with the developer on fixing the View Daily Expenses feature.
Developer shall fix the system to display error message as mentioned.

Test Incident Number	TIR-03-004
Summary	
The system does not display "Date should be valid" when the date is null.	
Date and Time Incident	12/06/2022
Context	Expense Manager (EM_1.1)
Description of Incident	
Test Procedure	TP-03-001
Test Data	Input nothing to input null date.
Expected Result	System displays error dialog "Date should be valid"
Actual Result	Page becomes blank System does not display error message.
Unexpected Outcome	
Procedure to reproduce the incident	Input nothing to input null date. [TC-03-005].
Test Environment	Hardware: Windows 10, Intel (64 bit), 8GB RAM Reference: EM_TP_1.O
Attempt to repeat	Procedure done 3 times continuously after incident 1 st discovered. Same result emerged.
Tester's Name	Kisyok Indran A/L Veerasamy
Observer's Name (witness)	Fahad Aqeel Thakur
Status of Incident	
Open	
Impact	
Major – Sever problems but possible to work around	
Priority	

Delayed – System is unstable but incident must be fixed prior to next level of test or shipment

Description of the corrective action

To discuss with the developer on fixing the View Daily Expenses feature.
Developer shall fix the system to display error message as mentioned.

Test Incident Number	TIR-05-001
Summary	
The system does not display “From Date must be less than or equals to To Date” error message when the From Date input value is less than To Date input value.	
Date and Time Incident	12/06/2022
Context	Expense Manager (EM_1.1)
Description of Incident	
Test Procedure	TP-05-001
Test Data	Input “01/02/2000” into From Date field and “01/01/2000” into To Date field
Expected Result	The system display “From Date must be less than or equals to To Date” error message
Actual Result	System does not display error message.
Unexpected Outcome	
Procedure to reproduce the incident	<ol style="list-style-type: none"> 1. Input into From Date field the value: “01/02/2000” [TC-05-002]. 2. Input into To Date field the value: “01/01/2000” [TC-05-002].
Test Environment	Hardware: Windows 10, Intel (64 bit), 8GB RAM Reference: EM_TP_1.0
Attempt to repeat	Procedure done 3 times continuously after incident 1 st discovered. Same result emerged.
Tester’s Name	Muhammad Irfan bin Nasruddin
Observer’s Name (witness)	Fahad Aqeel Thakur
Status of Incident	
Open	
Impact	
Minor – Does not impact the functionality or usability of the process is not according to requirements/design specifications	
Priority	

Delayed – System is unstable, but incident must be fixed prior to next level of test or shipment

Description of the corrective action

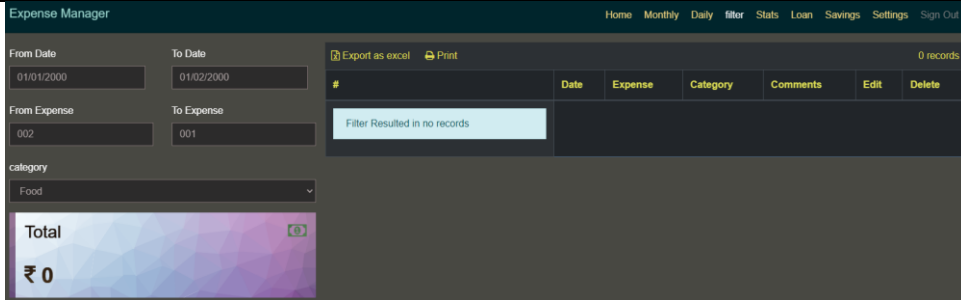
To discuss with the developer on fixing the Filter Expenses feature. Developer shall fix the system to display error message as mentioned.

Test Incident Number	TIR-05-002
Summary	
The system does not display "Date range inputs cannot be null" error message when the From Date input value is null.	
Date and Time Incident	12/06/2022
Context	Expense Manager (EM_1.1)
Description of Incident	
Test Procedure	TP-05-001
Test Data	Input null into From Date field by making it empty
Expected Result	The system display "Date range inputs cannot be null" error message
Actual Result	System does not display error message. Page becomes blank.
Unexpected Outcome	
Procedure to reproduce the incident	1. Input into From Date field the value: null [TC-05-003].
Test Environment	Hardware: Windows 10, Intel (64 bit), 8GB RAM Reference: EM_TP_1.O
Attempt to repeat	Procedure done 3 times continuously after incident 1 st discovered. Same result emerged.
Tester's Name	Muhammad Irfan bin Nasruddin
Observer's Name (witness)	Fahad Aqeel Thakur
Status of Incident	
Open	
Impact	
Major – Severe problems but possible to work around	
Priority	

Delayed – System is unstable, but incident must be fixed prior to next level of test or shipment

Description of the corrective action

To discuss with the developer on fixing the Filter Expenses feature. Developer shall fix the system to display error message as mentioned and remedy blank page from appearing.

Test Incident Number	TIR-05-003
Summary	
The system does not display “From Expense must be less than or equals to To Expense” error message when the From Expense input value is greater than To Expense input value.	
Date and Time Incident	12/06/2022
Context	Expense Manager (EM_1.1)
Description of Incident	
Test Procedure	TP-05-001
Test Data	From Date: “01/01/2000” To Date: “01/02/2000” From Expense: 2 To Expense: 1
Expected Result	The system display “From Expense must be less than or equals to To Expense” error message
Actual Result	System does not display error message.
Unexpected Outcome	 <p>The screenshot shows the Expense Manager interface. The search filters are set to: From Date: 01/01/2000, To Date: 01/02/2000, From Expense: 002, To Expense: 001, and category: Food. The table below the filters is empty, displaying "Filter Resulted in no records". The total amount shown is ₹ 0.</p>
Procedure to reproduce the incident	<ol style="list-style-type: none"> 1. Input into From Date field the value: “01/01/2000” 2. Input into To Date field the value: “01/02/2000” 3. Input into From Expense field the value: 2 4. Input into To Expense field the value: 1
Test Environment	Hardware: Windows 10, Intel (64 bit), 8GB RAM Reference: EM_TP_1.0
Attempt to repeat	Procedure done 3 times continuously after incident 1 st discovered. Same result emerged.
Tester’s Name	Muhammad Irfan bin Nasruddin
Observer’s Name (witness)	Fahad Aqeel Thakur
Status of Incident	
Open	
Impact	

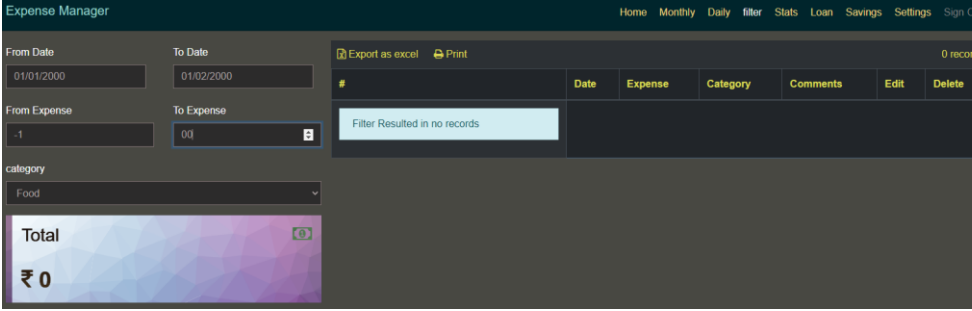
Minor – Does not impact the functionality or usability of the process is not according to requirements/design specifications

Priority

Delayed – System is unstable, but incident must be fixed prior to next level of test or shipment

Description of the corrective action

To discuss with the developer on fixing the Filter Expenses feature. Developer shall fix the system to display error message as mentioned.

Test Incident Number	TIR-05-004
Summary	
The system does not display “Expense range inputs must be greater than zero” error message when the From Expense input value is less than zero.	
Date and Time Incident	12/06/2022
Context	Expense Manager (EM_1.1)
Description of Incident	
Test Procedure	TP-05-001
Test Data	From Date: “01/01/2000” To Date: “01/02/2000” From Expense: -1
Expected Result	The system display “Expense range inputs must be greater than zero” error message
Actual Result	System does not display error message.
Unexpected Outcome	
Procedure to reproduce the incident	<ol style="list-style-type: none"> 1. Input into From Date field the value: “01/01/2000” 2. Input into To Date field the value: “01/02/2000” 3. Input into From Expense field the value: -1
Test Environment	Hardware: Windows 10, Intel (64 bit), 8GB RAM Reference: EM_TP_1.O
Attempt to repeat	Procedure done 3 times continuously after incident 1 st discovered. Same result emerged.
Tester’s Name	Muhammad Irfan bin Nasruddin
Observer’s Name (witness)	Fahad Aqeel Thakur
Status of Incident	
Open	
Impact	
Minor – Does not impact the functionality or usability of the process is not according to requirements/design specifications	

Priority
Delayed – System is unstable, but incident must be fixed prior to next level of test or shipment
Description of the corrective action
To discuss with the developer on fixing the Filter Expenses feature. Developer shall fix the system to display error message as mentioned.

Test Incident Number	TIR-05-005
Summary	
The system does not display “Expense range inputs cannot be of string type” error message when the From Expense input value is of string type.	
Date and Time Incident	12/06/2022
Context	Expense Manager (EM_1.1)
Description of Incident	
Test Procedure	TP-05-001
Test Data	From Date: “01/01/2000” To Date: “01/02/2000” From Expense: “j”
Expected Result	The system display “Expense range inputs cannot be of string type” error message
Actual Result	System does not display error message. Page becomes blank.
Unexpected Outcome	
Procedure to reproduce the incident	<ol style="list-style-type: none"> 1. Input into From Date field the value: “01/01/2000” 2. Input into To Date field the value: “01/02/2000” 3. Input into From Expense field the value: “j”
Test Environment	Hardware: Windows 10, Intel (64 bit), 8GB RAM Reference: EM_TP_1.O
Attempt to repeat	Procedure done 3 times continuously after incident 1 st discovered. Same result emerged.
Tester’s Name	Muhammad Irfan bin Nasruddin
Observer’s Name (witness)	Fahad Aqeel Thakur
Status of Incident	
Open	
Impact	

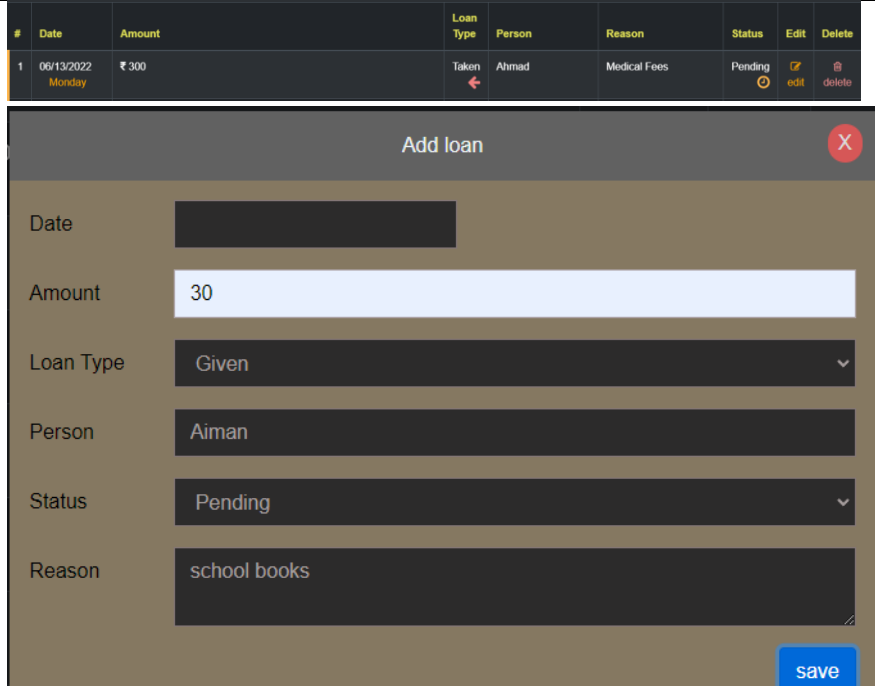
Major – Severe problems but possible to work around

Priority

Delayed – System is unstable, but incident must be fixed prior to next level of test or shipment

Description of the corrective action

To discuss with the developer on fixing the Filter Expenses feature. Developer shall fix the system to display error message as mentioned and remedy blank page from appearing.

Test Incident Number	TIR-06-001
Summary	
The system does not display error message "Date should be valid" when the date is invalid.	
Date and Time Incident	12/06/2022
Context	Expense Manager (EM_1.1)
Description of Incident	
Test Procedure	TP-06-004
Test Data	Date = 0/06/2022. Date = 33/06/2022. Date = "abc". Date = null.
Expected Result	System displays error message "Date should be valid"
Actual Result	Input date 0/06/2022: System does not display error message. System saves the current date. Input date 33/06/2022: System does not display error message. System saves the current date. Input date "abc": System does not display error message. System saves the current date. Input null date: System does not display error message. System froze.
Unexpected Outcome	

Procedure to reproduce the incident	Input date 0/06/2022. [TC-06-004] Input date 33/06/2022. [TC-06-004] Input date "abc". [TC-06-004] Input null date. [TC-06-004]
Test Environment	Hardware: Windows 10, Intel (64 bit), 8GB RAM Reference: EM_TP_1.O
Attempt to repeat	Procedure done 3 times continuously after incident 1 st discovered. Same result emerged.
Tester's Name	Kisyok Indran A/L Veerasamy
Observer's Name (witness)	Muhammad Hafiz bin Baharudin
Status of Incident	
Open	
Impact	
Major – Severe problems but possible to work around	
Priority	
Immediate: Must be fixed as soon as possible	
Description of the corrective action	
To discuss with the developer on fixing the Manage Loan feature. Developer shall fix the system to display error message as mentioned.	

Test Incident Number	TIR-06-002																				
Summary																					
The system does not display error message “Amount must be between 0 and 100000” when the amount is invalid.																					
Date and Time Incident	12/06/2022																				
Context	Expense Manager (EM_1.1)																				
Description of Incident																					
Test Procedure	TP-06-004																				
Test Data	Amount = -5000. Amount = 200000. Amount = “abc”.																				
Expected Result	System displays error message “Amount must be between 0 and 100000”																				
Actual Result	Input amount -5000: System does not display error message. System saves the amount. Input amount 200000: System does not display error message. System saves the amount. Input amount “abc”: System does not display error message. System does not allow user to input alphabets in amount field.																				
Unexpected Outcome	<table border="1"> <tr> <td>3</td> <td>06/07/2022</td> <td>Tuesday</td> <td>₹ -5,000</td> <td>Taken</td> <td>Ahmad</td> <td>Medical Fees</td> <td>Pending</td> <td>edit</td> <td>delete</td> </tr> <tr> <td>2</td> <td>06/13/2022</td> <td>Monday</td> <td>₹ 2,00,000</td> <td>Given</td> <td>Ahmad</td> <td>Medical Fees</td> <td>Pending</td> <td>edit</td> <td>delete</td> </tr> </table>	3	06/07/2022	Tuesday	₹ -5,000	Taken	Ahmad	Medical Fees	Pending	edit	delete	2	06/13/2022	Monday	₹ 2,00,000	Given	Ahmad	Medical Fees	Pending	edit	delete
3	06/07/2022	Tuesday	₹ -5,000	Taken	Ahmad	Medical Fees	Pending	edit	delete												
2	06/13/2022	Monday	₹ 2,00,000	Given	Ahmad	Medical Fees	Pending	edit	delete												
Procedure to reproduce the incident	Input amount -5000. [TC-06-005] Input amount 200000. [TC-06-005] Input amount “abc”. [TC-06-005]																				
Test Environment	Hardware: Windows 10, Intel (64 bit), 8GB RAM Reference: EM_TP_1.0																				
Attempt to repeat	Procedure done 3 times continuously after incident 1 st discovered. Same result emerged.																				
Tester's Name	Kisyok Indran A/L Veerasamy																				
Observer's Name (witness)	Muhammad Hafiz bin Baharudin																				
Status of Incident																					
Open																					
Impact																					

Major – Severe problems but possible to work around

Priority

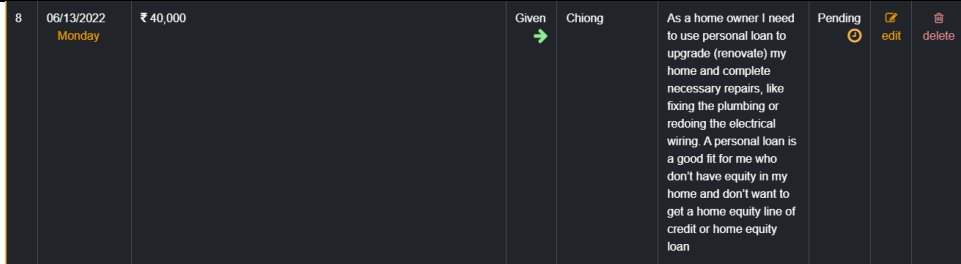
Immediate: Must be fixed as soon as possible
--

Description of the corrective action

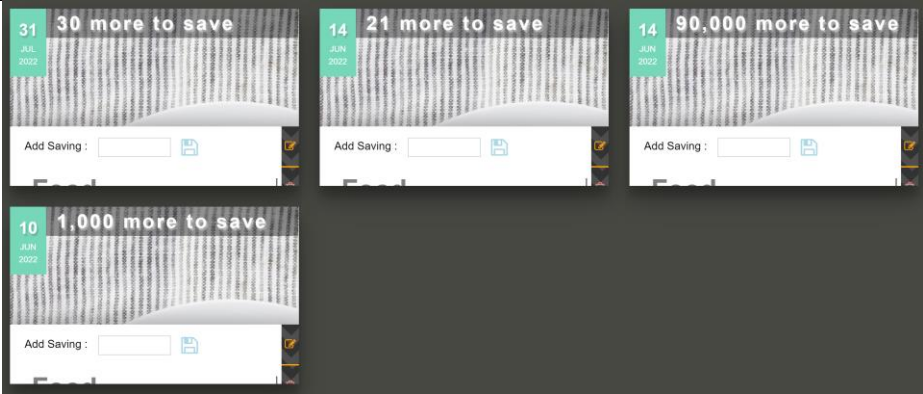
To discuss with the developer on fixing the Manage Loan feature. Developer shall fix the system to display error message as mentioned.
--

Test Incident Number	TIR-06-003																		
Summary																			
The system does not display error message “Person name must be between 0 and 80 characters” when the person’s name length is invalid. The system does not display error message “Invalid person name” when the person’ name contains number.																			
Date and Time Incident	12/06/2022																		
Context	Expense Manager (EM_1.1)																		
Description of Incident																			
Test Procedure	TP-06-004																		
Test Data	Person = “Siti Sarah Aqilah Rosaini Minah Binti Muhammad Abdullah Huhkum Kasim Karim Rahman” Person = “123”																		
Expected Result	System displays error message “Person name must be between 0 and 80 characters” or “Invalid person name”.																		
Actual Result	Input person “Siti Sarah Aqilah Rosaini Minah Binti Muhammad Abdullah Huhkum Kasim Karim Rahman”: System does not display error message. System saves the person’s name. Input person “123”: System does not display error message. System saves the person’s name.																		
Unexpected Outcome	<table border="1"> <tr> <td>6</td> <td>06/13/2022 Monday</td> <td>₹ 30</td> <td>Given →</td> <td>Siti Sarah Aqilah Rosaini Minah Binti Muhammad Abdullah Huhkum Kasim Karim Rahman</td> <td>school books</td> <td>Pending</td> <td>edit</td> <td>delete</td> </tr> <tr> <td>7</td> <td>06/13/2022 Monday</td> <td>₹ 40</td> <td>Given →</td> <td>123</td> <td>party with friends</td> <td>Pending</td> <td>edit</td> <td>delete</td> </tr> </table>	6	06/13/2022 Monday	₹ 30	Given →	Siti Sarah Aqilah Rosaini Minah Binti Muhammad Abdullah Huhkum Kasim Karim Rahman	school books	Pending	edit	delete	7	06/13/2022 Monday	₹ 40	Given →	123	party with friends	Pending	edit	delete
6	06/13/2022 Monday	₹ 30	Given →	Siti Sarah Aqilah Rosaini Minah Binti Muhammad Abdullah Huhkum Kasim Karim Rahman	school books	Pending	edit	delete											
7	06/13/2022 Monday	₹ 40	Given →	123	party with friends	Pending	edit	delete											
Procedure to reproduce the incident	Input person “Siti Sarah Aqilah Rosaini Minah Binti Muhammad Abdullah Huhkum Kasim Karim Rahman”. [TC-06-006] Input person “123”. [TC-06-006]																		
Test Environment	Hardware: Windows 10, Intel (64 bit), 8GB RAM Reference: EM_TP_1.0																		
Attempt to repeat	Procedure done 3 times continuously after incident 1 st discovered. Same result emerged.																		
Tester’s Name	Kisyok Indran A/L Veerasamy																		
Observer’s Name (witness)	Muhammad Hafiz bin Baharudin																		
Status of Incident																			
Open																			

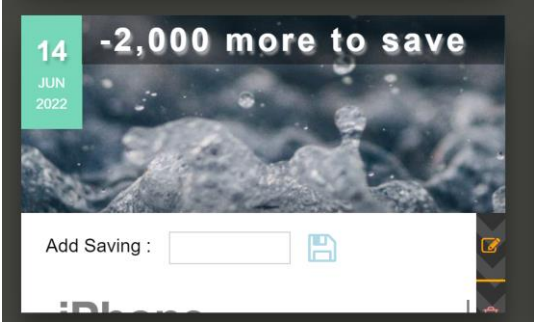
Impact
Major – Severe problems but possible to work around
Priority
Immediate: Must be fixed as soon as possible
Description of the corrective action
To discuss with the developer on fixing the Manage Loan feature. Developer shall fix the system to display error message as mentioned.

Test Incident Number	TIR-06-004
Summary	
The system does not display error message "Reason must be between 0 and 300 characters".	
Date and Time Incident	12/06/2022
Context	Expense Manager (EM_1.1)
Description of Incident	
Test Procedure	TP-06-004
Test Data	Reason = "As a home owner I need to use personal loan to upgrade (renovate) my home and complete necessary repairs, like fixing the plumbing or redoing the electrical wiring. A personal loan is a good fit for me who don't have equity in my home and don't want to get a home equity line of credit or home equity loan"
Expected Result	System displays error message "Reason must be between 0 and 300 characters".
Actual Result	System does not display error message. System saves the reason.
Unexpected Outcome	
Procedure to reproduce the incident	Input reason "As a home owner I need to use personal loan to upgrade (renovate) my home and complete necessary repairs, like fixing the plumbing or redoing the electrical wiring. A personal loan is a good fit for me who don't have equity in my home and don't want to get a home equity line of credit or home equity loan". [TC-06-007]
Test Environment	Hardware: Windows 10, Intel (64 bit), 8GB RAM Reference: EM_TP_1.0
Attempt to repeat	Procedure done 3 times continuously after incident 1 st discovered. Same result emerged.
Tester's Name	Kisyok Indran A/L Veerasamy
Observer's Name (witness)	Muhammad Hafiz bin Baharudin
Status of Incident	
Open	

Impact
Major – Severe problems but possible to work around
Priority
Immediate: Must be fixed as soon as possible
Description of the corrective action
To discuss with the developer on fixing the Manage Loan feature. Developer shall fix the system to display error message as mentioned.

Test Incident Number	TIR-07-001
Summary	
The system does not display error message when Goal Amount is greater than 80000	
Date and Time Incident	14/06/2022 (4:40 PM)
Context	Expense Manager (EM_1.1)
Description of Incident	
Test Procedure	TP-07-002
Test Data	Goal Amount = 90000
Expected Result	Display error message "Goal Amount should be between 0 & 80000"
Actual Result	System displays the Savings Goal with the invalid Goal Amount. System does not display error message.
Unexpected Outcome	
Procedure to reproduce the incident	Date: 24/06/2022 Goal Amount: 90000 Savings Amount: 500 Category: Car Comment: BMW [TC-07-003]
Test Environment	Hardware: Windows 10, Intel (64 bit), 8GB RAM Reference: EM_TP_1. O
Attempt to repeat	Procedure done 3 times after incident first emerged. Same result each time.
Tester's Name	Muhammad Hafiz bin Baharudin
Observer's Name (witness)	Kisyok Indran A/L Veerasamy
Status of Incident	
Open	
Impact	
Major – Severe problems but possible to work around	

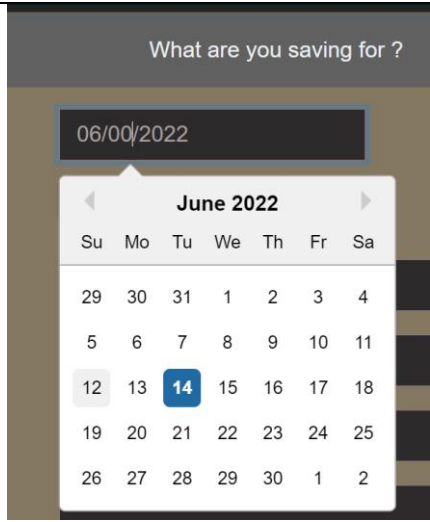
Priority
Immediate – Must be fixed as soon as possible
Description of the corrective action
To discuss with the developer on fixing the Manage Savings feature. Developer shall fix the system to display error message as mentioned.

Test Incident Number	TIR-07-002
Summary	
The system does not display error message when Goal Amount less or equal to Saving Amount	
Date and Time Incident	14/06/2022 (4:40 PM)
Context	Expense Manager (EM_1.1)
Description of Incident	
Test Procedure	TP-07-003
Test Data	Goal Amount = 1000 Saving Amount = 3000
Expected Result	Display error message "Goal Amount should be greater than Savings Amount"
Actual Result	System displays the Savings Goal with the invalid Goal Amount. System does not display error message.
Unexpected Outcome	
Procedure to reproduce the incident	Date: 24/06/2022 Goal Amount: 2000 Savings Amount: 3000 Category: Other Comment: iPhone [TC-07-005]
Test Environment	Hardware: Windows 10, Intel (64 bit), 8GB RAM Reference: EM_TP_1. O
Attempt to repeat	Procedure done 3 times after incident first emerged. Same result each time.
Tester's Name	Muhammad Hafiz bin Baharudin
Observer's Name (witness)	Kisyok Indran A/L Veerasamy
Status of Incident	
Open	
Impact	
Major – Severe problems but possible to work around	
Priority	

Immediate – Must be fixed as soon as possible

Description of the corrective action

To discuss with the developer on fixing the Manage Savings feature. Developer shall fix the system to display error message as mentioned.

Test Incident Number	TIR-07-003
Summary	
The system does not display error message date should be valid	
Date and Time Incident	14/06/2022 (5:00 PM)
Context	Expense Manager (EM_1.1)
Description of Incident	
Test Procedure	TP-07-004
Test Data	date = 0/06/2022
Expected Result	Display error message "Date should be valid"
Actual Result	No error message shown "Date should be valid"
Unexpected Outcome	
Procedure to reproduce the incident	Date: 0/06/2022 Goal Amount: 5000 Savings Amount: 3000 Category: Other Comment: iPhone [TC-07-006]
Test Environment	Hardware: Windows 10, Intel (64 bit), 8GB RAM Reference: EM_TP_1. O
Attempt to repeat	Procedure done 3 times after incident first emerged. Same result each time.
Tester's Name	Muhammad Hafiz bin Baharudin
Observer's Name (witness)	Kisyok Indran A/L Veerasamy
Status of Incident	
Open	
Impact	

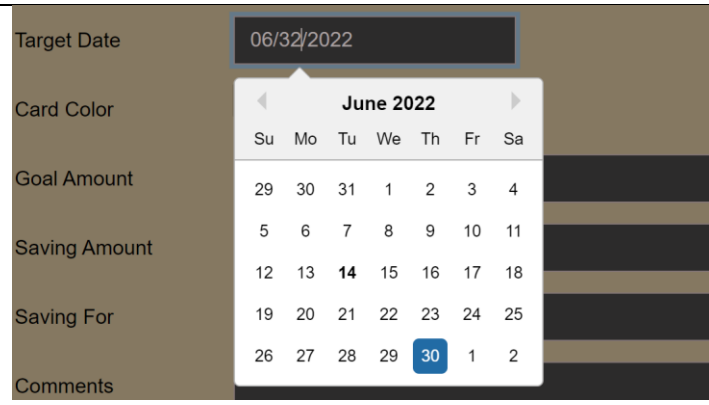
Major – Severe problems but possible to work around

Priority

Immediate – Must be fixed as soon as possible

Description of the corrective action

To discuss with the developer on fixing the Manage Savings feature. Developer shall fix the system to display error message as mentioned.

Test Incident Number	TIR-007-004
Summary	
The system does not display error message date should be valid	
Date and Time Incident	14/06/2022 (5:04 PM)
Context	Expense Manager (EM_1.1)
Description of Incident	
Test Procedure	TP-07-004
Test Data	date = 0/06/2022
Expected Result	Display error message "Date should be valid"
Actual Result	No error message shown "Date should be valid"
Unexpected Outcome	
Procedure to reproduce the incident	Date: 32/06/2022 Goal Amount: 10000 Savings Amount: 3000 Category: Other Comment: iPhone [TC-07-007]
Test Environment	Hardware: Windows 10, Intel (64 bit), 8GB RAM Reference: EM_TP_1. O
Attempt to repeat	Procedure done 3 times after incident first emerged. Same result each time.
Tester's Name	Muhammad Hafiz bin Baharudin
Observer's Name (witness)	Kisyok Indran A/L Veerasamy
Status of Incident	
Open	
Impact	
Major – Severe problems but possible to work around	
Priority	

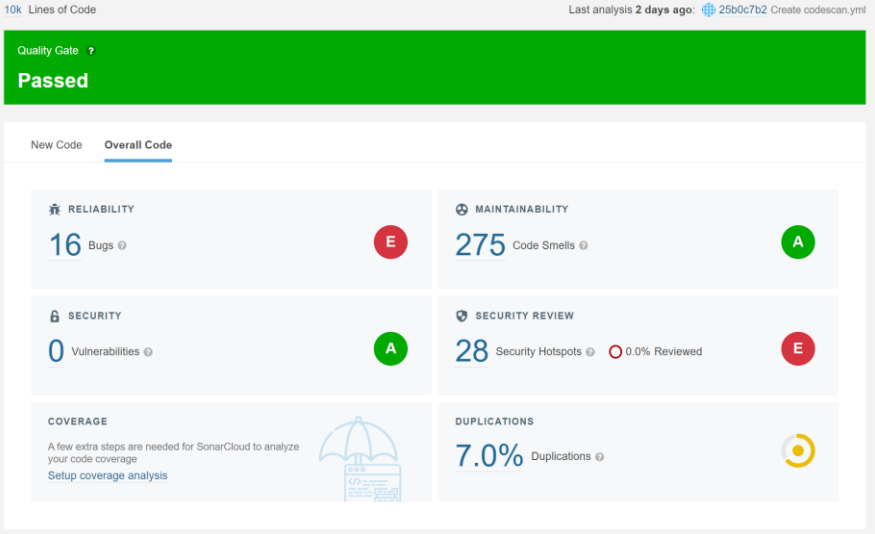
Immediate – Must be fixed as soon as possible

Description of the corrective action

To discuss with the developer on fixing the Manage Savings feature. Developer shall fix the system to display error message as mentioned.

Static Testing

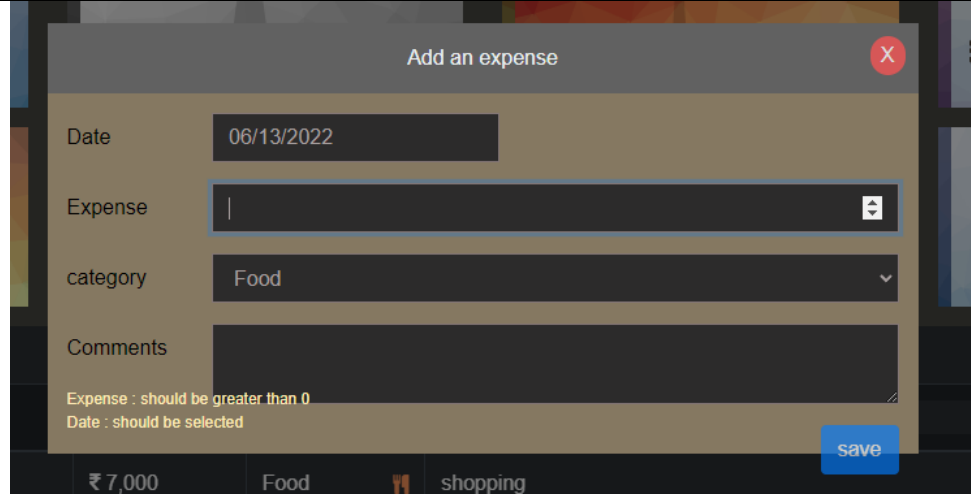
Note: The following test incidents were discovered via static testing.

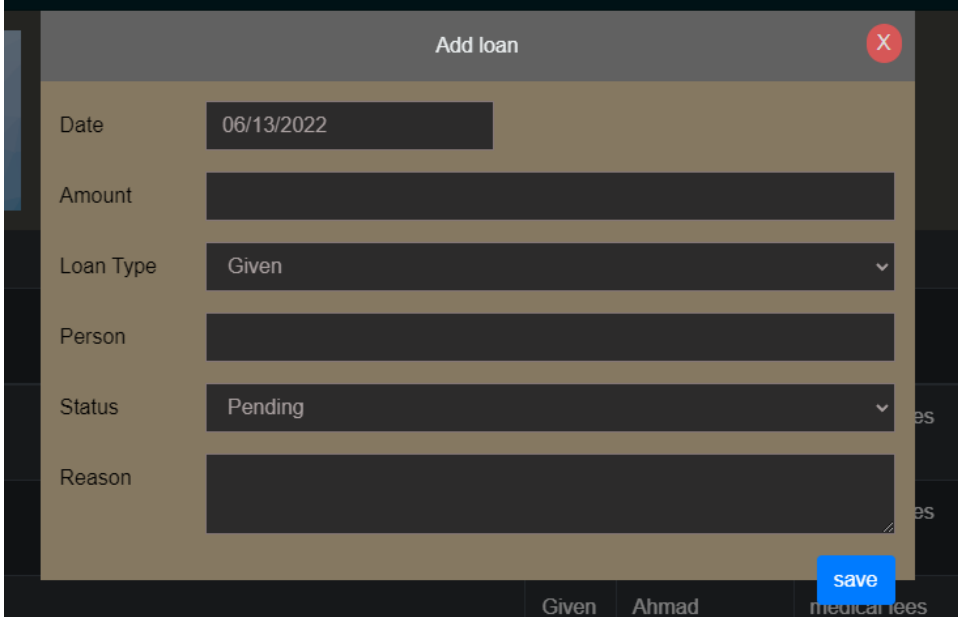
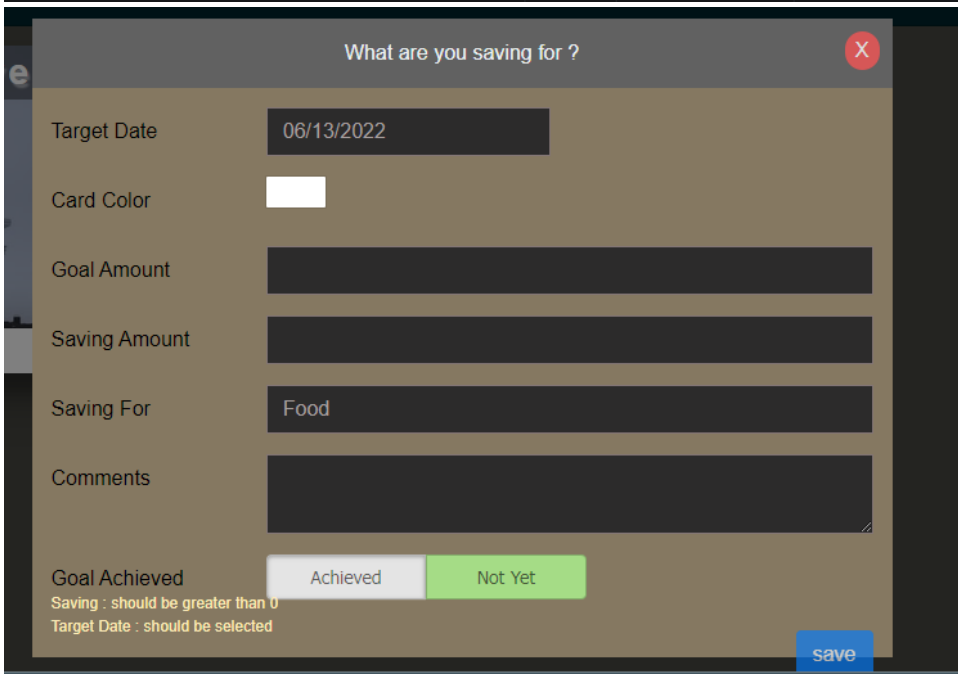
Test Incident Number	TIR-12-001
Summary	
Sonar Cloud Static Testing is applied to the EM	
Date and Time Incident	12/06/2022 (8:45 PM)
Context	Expense Manager (EM_1.1)
Description of Incident	
Test Procedure	
Test Data	Overall code of EM
Expected Result	No issues regarding reliability, maintainability, security, and duplications of the code
Actual Result	Several issues appear after the process of testing with Sonar Cloud that pertains to reliability, maintainability and security issues of the system that has not been reviewed
Unexpected Outcome	 <p>The screenshot shows a SonarCloud dashboard for a project with 10k Lines of Code. The Quality Gate is 'Passed'. Key metrics include: 16 Bugs (Error), 275 Code Smells (Alert), 0 Vulnerabilities (Alert), 28 Security Hotspots (Error) with 0.0% reviewed, and 7.0% Duplications (Alert). Reliability is marked with an Error (E) and Maintainability with an Alert (A). A note indicates that coverage analysis is not yet set up.</p>
Procedure to reproduce the incident	1. Run the repository in sonarcloud.
Test Environment	Sonar Cloud Free Testing : sonarcloud.io
Attempt to repeat	
Tester's Name	Muhammad Hafiz bin Baharudin
Observer's Name (witness)	Kisyok Indran A/L Veerasamy, Fahad Aqeel Thakur, Muhammad Irfan bin Nasruddin

Status of Incident
Open
Impact
Major – Severe problems but possible to work around
Priority
Immediate – Must be fixed as soon as possible
Description of the corrective action
To discuss with the developer on fixing the issues of reliability, maintainability and security review of the system, ensuring the quality, safety and performance of the system. Developer shall fix the issues of reliability and maintainability of the system and also review the security issues contain within the codes.

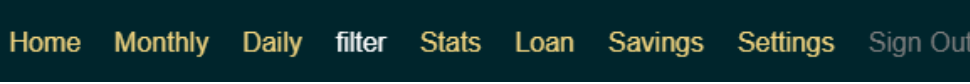
GUI Testing

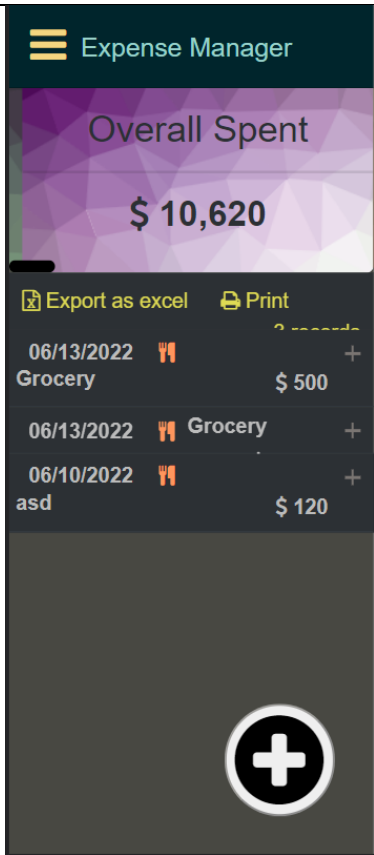
Note: The following test incidents were discovered via GUI testing.

Test Incident Number	TIR-13-001
Summary	
The “save” button is not aligned properly inside the Add Expense, Add Loan and Add Savings form modal.	
Date and Time Incident	12/06/2022
Context	Expense Manager (EM_1.1)
Description of Incident	
Test Procedure	
Test Data	“save” button
Expected Result	The “save” button is placed inside the form modal.
Actual Result	The “save” button overflows the form modal.
Unexpected Outcome	

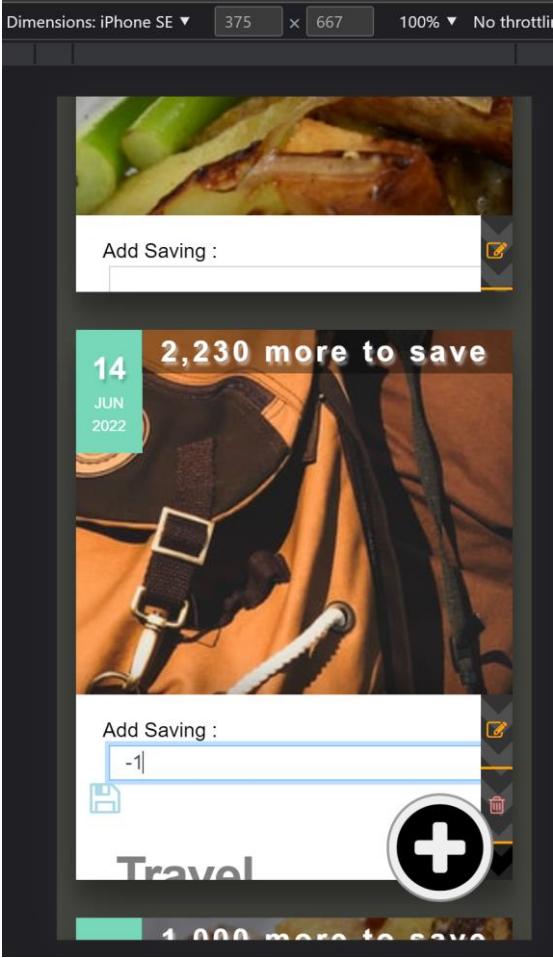
	 <p>The screenshot shows a mobile application form titled "Add loan". The form fields are: Date (06/13/2022), Amount (empty), Loan Type (Given), Person (empty), Status (Pending), and Reason (empty). A blue "save" button is at the bottom right. Below the form, there are labels "Given", "Ahmad", and "medical fees".</p>  <p>The screenshot shows a mobile application form titled "What are you saving for?". The form fields are: Target Date (06/13/2022), Card Color (white), Goal Amount (empty), Saving Amount (empty), Saving For (Food), and Comments (empty). At the bottom, there is a "Goal Achieved" section with "Achieved" and "Not Yet" buttons. A blue "save" button is at the bottom right. Below the form, there are labels "Given", "Ahmad", and "medical fees".</p>
Procedure to reproduce the incident	<ol style="list-style-type: none"> 1. Click on the "+" button on the "Home" page. 2. Click on the "+" button on the "Loan" page. 3. Click on the "+" button on the "Savings" page.
Test Environment	<p>Hardware: Windows 10, Intel (64 bit), 8GB RAM Reference: EM_TP_1.O</p>
Attempt to repeat	<p>Procedure done 3 times continuously after incident 1st discovered. Same result emerged.</p>
Tester's Name	<p>Kisyok Indran A/L Veerasamy</p>

Observer's Name (witness)	Muhammad Irfan bin Nasruddin
Status of Incident	
Open	
Impact	
Major – Severe problems but possible to work around	
Priority	
Immediate: Must be fixed as soon as possible	
Description of the corrective action	
To discuss with the developer on fixing the “save” button. Developer shall fix the system to place the button inside the form modal.	

Test Incident Number	TIR-13-002
Summary	
The “filter” navigation link in the top navigation bar is not properly cased.	
Date and Time Incident	12/06/2022
Context	Expense Manager (EM_1.1)
Description of Incident	
Test Procedure	
Test Data	“filter” navigation link
Expected Result	All the navigation link in the top navigation bar has their first character uppercased.
Actual Result	The “filter” navigation link doesn’t have its first character uppercased.
Unexpected Outcome	
Procedure to reproduce the incident	1. Login into the system.
Test Environment	Hardware: Windows 10, Intel (64 bit), 8GB RAM Reference: EM_TP_1.O
Attempt to repeat	Procedure done 3 times continuously after incident 1 st discovered. Same result emerged.
Tester’s Name	Muhammad Irfan bin Nasruddin
Observer’s Name (witness)	Kisyok Indran A/L Veerasamy
Status of Incident	
Open	
Impact	
Minor – Does not impact the functionality or usability of the process is not according to requirements/design specifications	
Priority	
Delayed – System is unstable, but incident must be fixed prior to next level of test or shipment	
Description of the corrective action	
To discuss with the developer on fixing the top navigation bar. Developer shall fix the system to properly case the navigation link in the top navigation bar, under all circumstances.	

Test Incident Number	TIR-13-003
Summary	
The table header of the expenses in the home page is not aligned correctly and part of it is not visible when viewed in Galaxy Fold Device	
Date and Time Incident	14/06/2022 (5:00 PM)
Context	Expense Manager (EM_1.1)
Description of Incident	
Test Procedure	
Test Data	Expense table header
Expected Result	The table is completely responsive to different device types and resolutions.
Actual Result	The expense table header is not aligned correctly and part of it is not visible when viewed in Galaxy Fold Device
Unexpected Outcome	
Procedure to reproduce the incident	<ol style="list-style-type: none"> 1. Login to the system on google chrome. 2. Click Ctrl + Shift + I key to open the chrome dev tools. 3. Toggle device emulation by clicking CTRL + Shift + M. 4. Change the dimension to Galaxy Fold.
Test Environment	Hardware: Windows 10, Intel (64 bit), 8GB RAM Reference: EM_TP_1.O
Attempt to repeat	Procedure done 3 times continuously after incident 1 st discovered. Same result emerged.

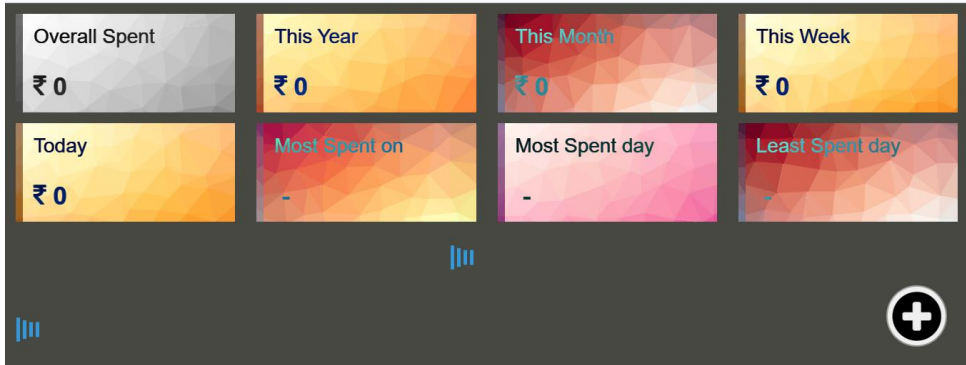
Tester's Name	Fahad Aqeel Thakur
Observer's Name (witness)	Muhammad Irfan bin Nasruddin
Status of Incident	
Open	
Impact	
Minor – Does not impact the functionality or usability of the process is not according to requirements/design specifications	
Priority	
Delayed – System is unstable, but incident must be fixed prior to next level of test or shipment	
Description of the corrective action	
To discuss with the developer on fixing the responsiveness of the website. Developer shall fix the system to make the application components responsive on different devices.	

Test Incident Number	TIR-13-004
Summary	
The Savings Goal Card is not aligned correctly, and some part of the card is not visible within mobile view such as when viewed in iPhone SE Device size.	
Date and Time Incident	14/06/2022 (11:27 PM)
Context	Expense Manager (EM_1.1)
Description of Incident	
Test Procedure	
Test Data	Savings Goal Card
Expected Result	The Savings Goal Card is completely responsive to different device types and resolutions.
Actual Result	The Savings Goal Card is not aligned correctly and part of it is not visible when viewed in iPhone SE Device size
Unexpected Outcome	
Procedure to reproduce the incident	<ol style="list-style-type: none"> 1. Login to the system on google chrome. 2. Go to Savings Page 3. Click Ctrl + Shift + I key to open the chrome dev tools. 4. Toggle device emulation by clicking CTRL + Shift + M. 5. Change the dimension to iPhone SE.

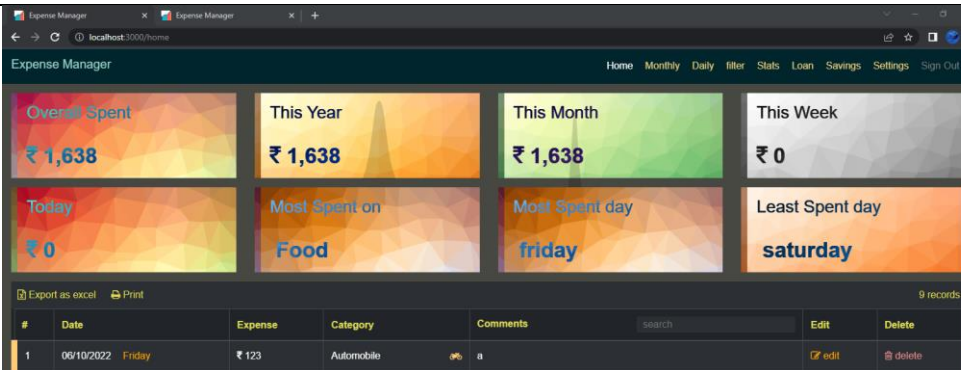
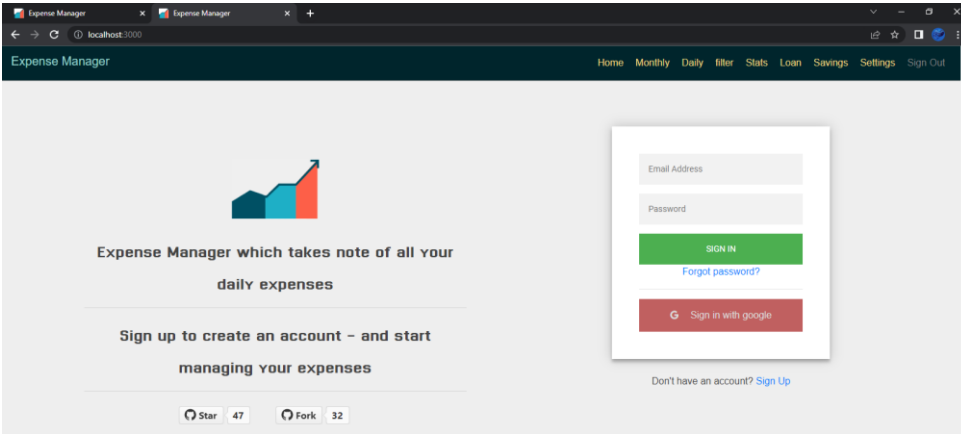
Test Environment	Hardware: Windows 10, Intel (64 bit), 8GB RAM Reference: EM_TP_1.O
Attempt to repeat	Procedure done 3 times continuously after incident 1 st discovered. Same result emerged.
Tester's Name	Muhammad Hafiz bin Baharudin
Observer's Name (witness)	Fahad Aqeel Thakur
Status of Incident	
Open	
Impact	
Minor – Does not impact the functionality or usability of the process is not according to requirements/design specifications	
Priority	
Delayed – System is unstable, but incident must be fixed prior to next level of test or shipment	
Description of the corrective action	
To discuss with the developer on fixing the responsiveness of the website. Developer shall fix the system to make the application components responsive on different devices.	

Error Guessing

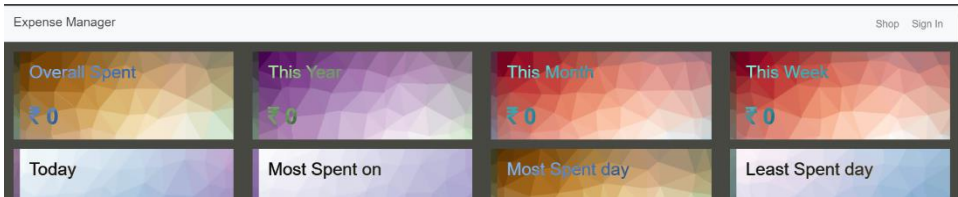
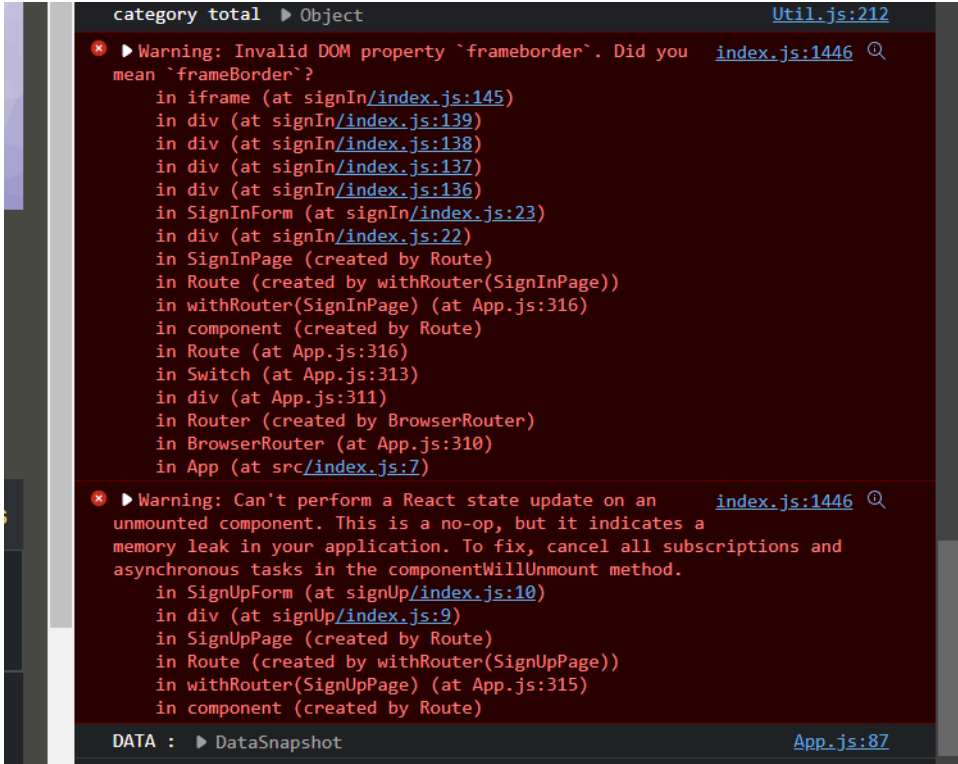
Note: The following test incidents were discovered via error guessing.

Test Incident Number	TIR-14-001
Summary	
The system crashes when pressing “+” button after signing out.	
Date and Time Incident	12/06/2022
Context	Expense Manager (EM_1.1)
Description of Incident	
Test Procedure	
Test Data	Click “+” button.
Expected Result	The system displays error message “Please sign in first”.
Actual Result	The system crashes and displays blank page.
Unexpected Outcome	<p>After signing out and clicking browser’s back button:</p>  <p>After clicking “+” button:</p> <div style="background-color: #cccccc; height: 150px; width: 100%;"></div>
Procedure to reproduce the incident	<ol style="list-style-type: none"> 1. Click “Sign Out” button. 2. Click browser’s back button. 3. Click “+” button.


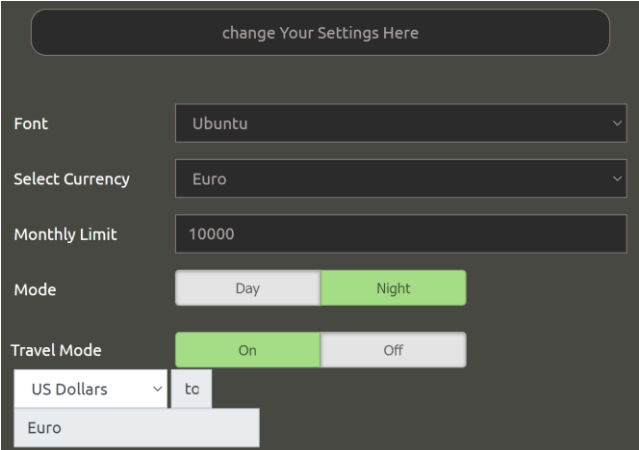
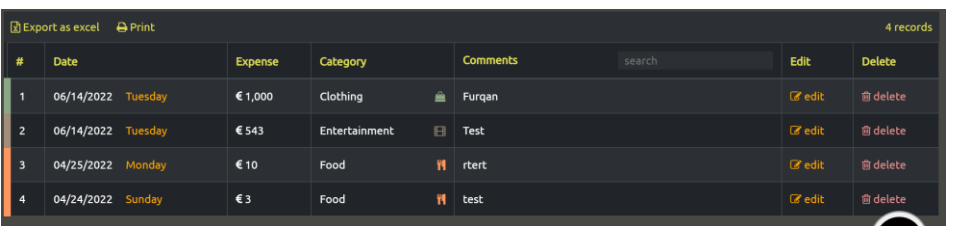
Test Environment	Hardware: Windows 10, Intel (64 bit), 8GB RAM Reference: EM_TP_1.O
Attempt to repeat	Procedure done 3 times continuously after incident 1 st discovered. Same result emerged.
Tester's Name	Kisyok Indran A/L Veerasamy
Observer's Name (witness)	Muhammad Irfan bin Nasruddin
Status of Incident	
Open	
Impact	
Major – Severe problems but possible to work around	
Priority	
Immediate: Must be fixed as soon as possible	
Description of the corrective action	
To discuss with the developer on fixing the sign out function. Developer shall fix the system to return the user to Sign In page after signing out.	

Test Incident Number	TIR-14-002
Summary	
The system display login page when the user is still logged in but opens another browser tab of the website	
Date and Time Incident	12/06/2022
Context	Expense Manager (EM_1.1)
Description of Incident	
Test Procedure	
Test Data	Website link: localhost:3000
Expected Result	The system displays the Home page when user is already logged in and opens another browser tab of the website.
Actual Result	The system displays login page while the user login session is still valid.
Unexpected Outcome	 <p>After opening a new tab of the website despite the login session is still valid.</p> 
Procedure to reproduce the incident	<ol style="list-style-type: none"> 1. Login in into the system 2. Open another browser tab 3. Type in the link of the website: localhost:3000 into the browser link bar, then press enter
Test Environment	Hardware: Windows 10, Intel (64 bit), 8GB RAM Reference: EM_TP_1.0

Attempt to repeat	Procedure done 3 times continuously after incident 1 st discovered. Same result emerged.
Tester's Name	Muhammad Irfan bin Nasruddin
Observer's Name (witness)	Kisyok Indran A/L Veerasamy
Status of Incident	
Open	
Impact	
Minor – Does not impact the functionality or usability of the process is not according to requirements/design specifications	
Priority	
Delayed – System is unstable, but incident must be fixed prior to next level of test or shipment	
Description of the corrective action	
To discuss with the developer on fixing the user experience problem. Developer shall fix the system to automatically display Home page when user is already logged in and opens the website.	

Test Incident Number	TIR-14-003
Summary	
After creating a new account in the system with email and being redirected to the home page, it displays the top-navigation bar as though the user is not logged into the system.	
Date and Time Incident	14/06/2022 (5:30 PM)
Context	Expense Manager (EM_1.1)
Description of Incident	
Test Procedure	
Test Data	New account creation using email at http://localhost:3000/signup
Expected Result	The system creates the account and logs the user into the system without issues.
Actual Result	The system displays a wrong top navigation bar to the user after creating a new account with email, as though the user was not signed in.
Unexpected Outcome	 <p>Top navigation bar error even though we can access the system.</p>  <p>Errors thrown by the system after performing these steps.</p>

Procedure to reproduce the incident	<ol style="list-style-type: none"> 1. On the sign up link http://localhost:3000/signup create a new account with your credentials. 2. The system will log in and redirect to the home page. 3. The navigation bar is similar to the one when not signed in.
Test Environment	Hardware: Windows 10, Intel (64 bit), 8GB RAM Reference: EM_TP_1.O
Attempt to repeat	Procedure done 3 times continuously after incident 1 st discovered. Same result emerged.
Tester's Name	Fahad Aqeel Thakur
Observer's Name (witness)	Kisyok Indran A/L Veerasamy
Status of Incident	
Open	
Impact	
Major – Severe problems but possible to work around	
Priority	
Immediate: Must be fixed as soon as possible	
Description of the corrective action	
To discuss with the developer on fixing the error with the navigation bar which hinders the user experience. Developer shall fix the system to properly display the correct navigation bar after the user has created an account.	

Test Incident Number	TIR-14-004
Summary	
Travel Mode Currency conversion needs to convert already saved expense to the selected currency	
Date and Time Incident	14/06/2022 (5:30 PM)
Context	Expense Manager (EM_1.1)
Description of Incident	
Test Procedure	
Test Data	Select Currency: US Dollar Travel Mode: Euro to US Dollar
Expected Result	The expenses is converted following the conversion from Euro to USD Dollar
Actual Result	The system displays a wrong expense conversion and only change the currency tag
Unexpected Outcome	 <p>Before conversion</p>  <p>Change currency from Euro to USD Dollar</p>  <p>Wrong conversion for the currency and only the tag changed</p>

Procedure to reproduce the incident	<ol style="list-style-type: none"> 1. Create an expense in US Dollar 2. Activate Travel Mode from US Dollar to Euro 3. Expense does not convert the currency
Test Environment	Hardware: Windows 10, Intel (64 bit), 8GB RAM Reference: EM_TP_1.O
Attempt to repeat	Procedure done 3 times continuously after incident 1 st discovered. Same result emerged.
Tester's Name	Muhammad Hafiz
Observer's Name (witness)	Kisyok Indran A/L Veerasamy
Status of Incident	
Open	
Impact	
Major – Severe problems but possible to work around	
Priority	
Immediate: Must be fixed as soon as possible	
Description of the corrective action	
To discuss with the developer on fixing the error with the currency conversion. Developer shall fix the system to properly display the correct currency conversion after they activated the travel mode.	

Test Summary Report

Expense Manager (EM)

Independent Verification and Validation

Version: 1.0.0

Date: 17/06/2022

General Information			
Test Report Description	This Test Summary Report will provide a summary of the findings based on the Test Log, the Test Incident Report.		
Version Author	Kisyok Indran A/L Veerasley	Contact Number	-
Associated Test Plan Reference	EM_TPS_1.0.0		

Summary				
Test Status Report Summary				
The status of the test are as follow:				
Number of Test Cases Planned to be Completed	Number of Test Cases Remaining to be Executed	Number of Test Cases Completed		
40	0	40		
Current status of the Test Incident Report are as follow:				
New (new incident introduced in current iteration)	Open (Incident discovered from previous test iteration that still pending to be resolved)	Reject (Incidents that were rejected after investigation because it cannot be categorised as incident)	Resolved (Incidents that have been confirmed resolved)	Deferred (Incidents that have been deferred. Incidents are confirmed to be fixed for the next release version)
Number of new incidents = 29	Number of Open Incident = 29	Number of Reject Incidents = 0	Number of Resolved Incidents = 0	Number of Deferred Incident = 0
Note: The number of Incident is counted based on the number of Test Incident ID and not based from the Test Incident Report ID				
0 Incidents have been deferred temporarily and are planned to be fixed in the next version, prior to product release are as below:				
Incident ID	Short Description	Status		

Document Reference

- EM_TP_1.0.0
- EM_TDS_1.0.0
- EM_TCS_1.0.0
- EM_TPS_1.0.0
- EM_TTR_1.0.0
- EM_TL_1.0.0
- EM_TIR_1.0.0

Changes from Plans

There has been no change from plan and the test has completed successfully. Refer to the schedule attached below.

Comprehensive Assessment

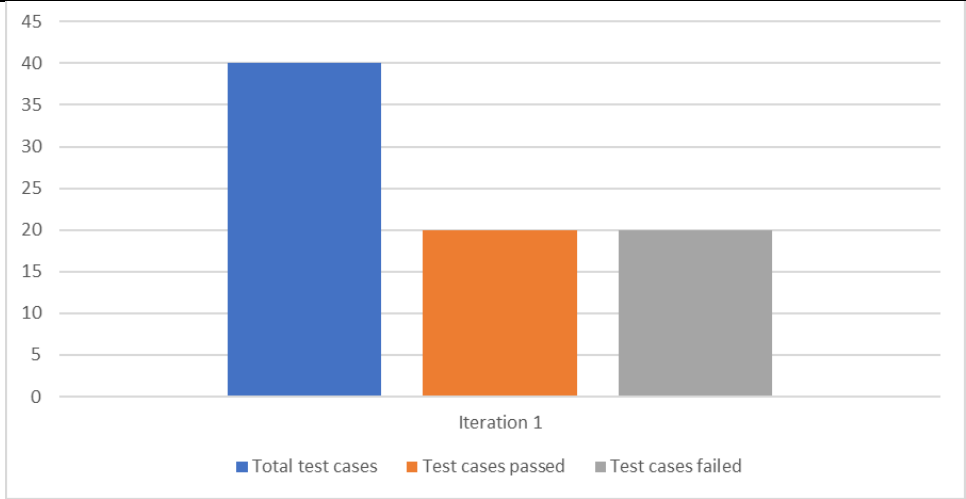
All 29 incidents have been opened. The incidents were caused by logic error and are planned to be corrected by the development team. For the schedule adherence, refer to the diagram below:

Testing Schedule	
	Actual
	Planned
16 May 2022	16 June 2022

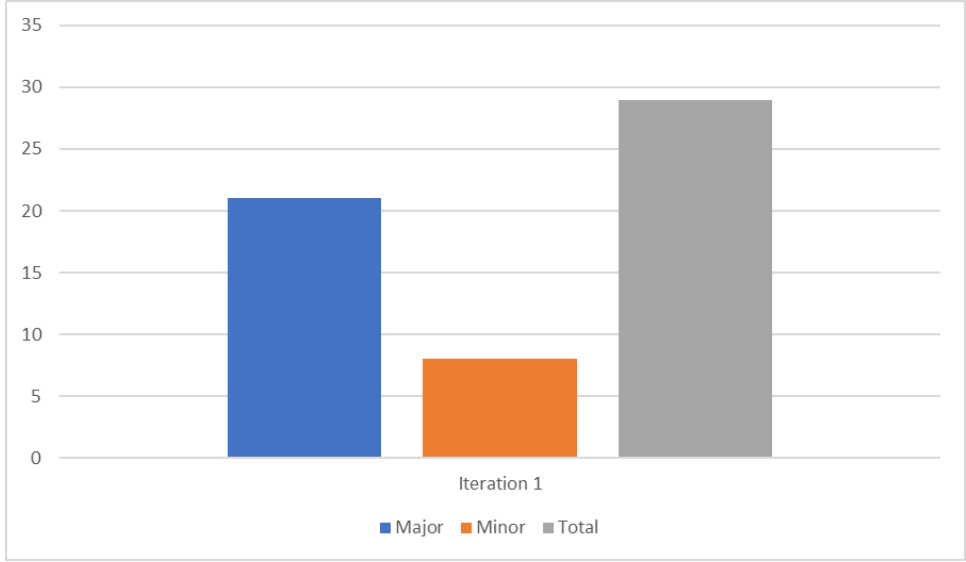
For the Test Coverage completeness refer to the table below:

Test Documents	% Completed	Date of Completion
Test Plan	100	17/6/2022
Test Design Specification	100	17/6/2022
Test Case Specification	100	17/6/2022
Test Procedure Specification	100	17/6/2022
Test Log	100	17/6/2022
Test Incident Report	100	17/6/2022

The Test Case Coverage is shown below:



Below are the lists of defects found through testing, by severity.



NOTE:

- Incident having “Mission Critical” impact = Fatal defect.
- Incident having “Major” impact = Major defect.
- Incident having “Minor” impact = Minor defect.

Result Summary

The items tested were Add New Expenses, View Monthly Expenses, View Daily Expenses, View Overall Stats, Filter Expenses, Manage Loan, Manage Savings and Edit Currency Setting features in EM. The tests were conducted by using dummy accounts.

40 test cases have been planned and executed. 50% of the test cases were recorded with incident.

100% of the incidents are opened to be fixed by the developers. The fatal defects found are considered to be high risk and must be fixed as soon as possible.

The product will be retested after defects have been found to ensure that all defects have been addressed. If there is no new incident raised, it is expected that the product has conformed to the SBD document, version 1.0.

Rationale for Decisions



Due to existence of open defects, at the end of the test iteration, the test team highly suggest that those defects are fixed prior to release. Existence of fatal defects signifies high product risk.

Conclusion and Recommendation Based on Test Result

Conclusion: The current version of product is not fit for release until (fatal) defects are fixed.

Recommendation: A more thorough logic check on the system may lessen similar defects in the future.

Approvals

Name	Job Title	Signature
Prepared by: Kisyok Indran A/L Veerasley	Test Lead	
Reviewed by: Muhammad Irfan bin Nasruddin	Test Analyst	
Approved by: Dr. Siti Hafizah Binti Ab Hamid	Test Manager	

Test Completion Report

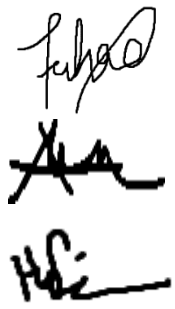

Expense Manager (EM)

Independent Verification and Validation

Version: 1.0.0

Date: 17/06/2022

Document Name	EM Test Completion Report (Iteration 1)
Reference Number	EM_TCR_1
Version	1.0.0
Project Code	EM
Status	In-use
Date Released	17 June 2022

Name	Position	Signature
Prepared by: Muhammad Irfan bin Nasruddin Muhammad Hafiz bin Baharudin Fahad Aqeel Thakur	Test Analyst EM Test Team	
Reviewed by: Kisyok Indran A/L Veerasamy	Test Lead EM Test Team	
Verified by: Dr. Siti Hafizah Binti Ab Hamid	Test Manager EM Test Team	

Version History

Version	Release Date	Section	Amendments
1.0	17/6/2022	All	Original Document

Distribution List

Version	Release Date	Controlled Copy No	Recipient Name	Department	Issue Date	Return Date
1.0.0	17/6/2022	01		EM Test Team	17/6/2022	

1.0 Summary of Testing Performed

We tested 8 features of the Expense Manager application. The features are as follows:

- i. F001 Add New Expense
- ii. F002 View Monthly Expenses
- iii. F003 View Daily expenses
- iv. F004 View Overall Stats
- v. F005 Filter Expenses
- vi. F006 Manage Loan
- vii. F007 Manage Savings
- viii. F011 Edit Currency Settings

Main focus was given to the black-box testing procedures where we tested the features using equivalence partitioning, boundary value analysis, decision table testing, state transition testing and use case testing. In addition, we also performed GUI and error guessing testing for some of the features in the Expense Manager and later Static Testing was done using Sonar Cloud Testing.

Equivalence partition testing was done to test certain valid and invalid classes that could occur with the inputs into the feature. This was similar to the Boundary Value Analysis testing where we would test the boundaries to determine the valid and invalid classes. Decision table testing allowed us to test the various decision points that would occur in the particular feature and the output related to that decision. State Transition testing allowed us to test various states of the feature. Lastly, Use Case testing allowed us to test the use case as described in the EM_SBD.

Test cases were created for each test that was designed in the Test Design Specification. Based on these test cases, test procedures were written down to allow the tester to perform the tests and log their findings in the Test Log Report.

The testing was performed using a chromium-based web browser with JavaScript enabled.

2.0 Deviations from Planned Testing

There were no deviations from the planned testing, and all the test activities were performed based on the Test Plan specified.

3.0 Test Completion Evaluation

In our test plan, the criteria to pass were as follows:

All test cases must be passed.

0% calculation error in any function involving expenditures, loans, and savings.

No deadlock during operation.

Based on these criteria, we can say that this system did not pass our tests as there were quite a few test cases which failed to produce the expected result. In total there were 20 defects with the functionality of the system. Some of these were major issues causing the entire system to crash and these issues need to be fixed as soon as possible by the developer.

The errors occurring in the Create new expenses, Manage loan & Manage savings features are of high-risk impact as specified in Table 3.7.3 of the Test Plan. These issues would need to be focused on by the developer as they would cause degraded user experience due to system crashes or invalid inputs.

4.0 Factor that Blocked Progress

There were no factors that blocked the progress of testing. All the tests were performed based on the Test Plan specified.

5.0 Test Measure

Total Test Cases	40
Total Test Procedures	20
Total test Coverage Items	275
Total Functionality Defects Reported	20
Number of incident reports	29
Resources Consumed	N/A

6.0 New / Changed / Residual Risk

All incidents reported in the Test Log and Test Incident Report shows there are many major risks that should be fixed as soon as possible as they impose risks on the system.

7.0 Test Deliverables

The following are the list the entire test deliverables produced as a result of the test effort:

- i. Test Transmittal Report (EM_TRR_1.0.0)
- ii. Test Plan (EM_TP_1.0.0)
- iii. Test Design Specification (EM_TDS_1.0.0)
- iv. Test Case Specification (EM_TCS_1.0.0)
- v. Test Procedure Specification (EM_TPS_1.0.0)
- vi. Test Log (EM_TL_1.0.0)
- vii. Test Incident Report (EM_TIR_1.0.0)
- viii. Test Summary
- ix. Test Completion Report

8.0 Reusable Test Assets

All test cases in the Test Case Specification can be reused in the next iteration of testing once the defects found and reported in Test Log and Test Incident Report are fixed.

9.0 Lessons Learnt

All of the Expense Manager's Main flows features described within the Test Plan (EM_TP_1.0.0) are usable in production with all the main flows features passing all the tests design. Based on the Test Summary Report, we have completed over 40 Test Cases with over 275 Test Coverage Items. We learnt that 50% of the test cases are recorded with incident and over 29 of the new incidents were caused by logic error and is needed to be planned and corrected by the development team. Of the 29 incidents the majority of the incidents are fatal defects and is considered to be high risk and must be fixed as soon as possible, as such the current version of Expense Manager is not fit for release until all of the fatal defects are fixed, it is also mentioned within the static testing result that the system's security, reliability and maintainability is needed to be reviewed again by the developer to maintain high level of quality for the system. We have also learned that a more thorough logic check of the system before release may lessen the defects in the future.