Software Requirements Spec. (SRS): <Expense Tracker Software>

# Document Information

This document is based on IEEE Standard 830.2

## List of Authors

|  |
| --- |
| Names |
| Charles Eboson |

## Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Notes |
| 0.1 | 26.10.2024 | Charles Eboson | First draft |
| 0.2 | 29.10.2024 | Charles Eboson | Added a login feature |
| 0.3 | 28.11.2024 | Charles Eboson | Modified use cases  **Noted Removal**: Removed the ability to view recent transactions |

Content

0 Document Information 1

0.1 List of Authors 1

0.2 Revision History 1

1 Introduction 2

1.1 Purpose 2

1.2 Scope 2

1.3 Glossary 2

2 General description 2

2.1 Product perspective 2

2.2 Product functions 3

2.3 User characteristics 3

2.4 Constraints 3

2.5 Assumptions and dependencies 3

3 External interface requirements 3

4 Functional requirements 4

5 Non-functional requirements 4

6 Use Case Modelling 4

6.1 Use Cases 4

6.2 Traceability Matrix 4

6.3 Detailed Use Case Descriptions 4

7 References 5

# Introduction

## Purpose

The Expense Tracker Software is designed to track, organise and analyse user’s personal expenses over various time periods, providing essential tools for managing personal finances effectively.

## Scope

The Expense Tracker Software will help users track their daily, monthly, or yearly expenses by recording transactions, categorizing expenses, viewing summaries, and generating reports. Many users face challenges in recalling cash flow and accounting for past transactions, and this software aims to tackle those challenges by offering an organized, accessible solution that promotes effective budget management and financial awareness.

## Glossary

The following abbreviations and domain-specific terms are used throughout the document.

|  |  |  |  |
| --- | --- | --- | --- |
| Abbreviation | Term | Description | Reference |
|  |  |  |  |

# General description

## Product perspective

The Expense Tracker Software is a new, self-contained application created to address common challenges in personal financial management. Developed based on extensive research into personal finance tracking needs and user behaviours, this software is not part of a larger product family nor a replacement for an existing system; it is designed from the ground up to function independently.

The Expense Tracker is intended as a standalone tool, requiring no dependencies or external system integrations. Users interact directly with the application, which provides all necessary features within a single interface. The software captures and categorizes expenses, generates summary reports, and allows users to view spending patterns over different timeframes.

## Product functions

* Add expenses
* Add income
* View transaction history
* Categorize expenses
* Calculate total balance
* Generate summary report
* Edit or delete transactions
* Save data to file
* Load data from file

## User characteristics

**General User**

* User can login and logout
* User can add expenses
* User can add income
* User can view transaction history
* User can categorize expenses
* User can calculate total balance
* User can generate summary report
* User can edit or delete transactions
* User can save data to file
* User can load data from file

## Constraints

None

## Assumptions and dependencies

The software is written in English, and it is assumed that everyone should understand English to use the software

# External interface requirements

The software will run on Windows and MacBook

# Functional requirements

|  |  |  |
| --- | --- | --- |
| Identifier | Priority (MoSCoW) | Requirement |
| REQ-1 | MUST | The system shall allow users add expenses |
| REQ-2 | MUST | The system shall allow users add income |
| REQ-3 | MUST | The system shall allow users view transaction history |
| REQ-4 | MUST | The system shall allow users edit / delete transactions |
| REQ-5 | SHOULD | The system shall allow users categorize expenses |
| REQ-6 | SHOULD | The system shall allow users view report summary |
| REQ-7 | MUST | The system shall allow users view total balance |
| REQ-8 | MUST | The system shall allow users save / load data from file |
| REQ-9 | MUST | The system shall allow users login and logout |

# Non-functional requirements

# Use Case Modelling

## Use Cases

UC diagram:

A diagram of a flowchart

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Use Case Name | Initiating actor | Actor’s goal |
| UC1 | **Add Transaction** | **User** | **To add either income or expenses** |
| UC2 | **View Transaction History** | **User** | **To view previous transactions** |
| UC3 | **Categorize Expenses** | **User** | **To group expenses into various categories** |
| UC4 | **Edit / Delete Transaction** | **User** | **To modify or delete a transaction** |
| UC5 | **View Report Summary** | **User** | **To view a summary of expenses in comparison to the income** |
| UC6 | **View Total Balance** | **User** | **To view Income minus expenses** |
| UC7 | **Save / Load data from file** | **User** | **To save and load a data from a file** |
| UC8 | **User Authentication** | **User** | **To login and logout of the system** |

## Traceability Matrix

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Requirements Identifier / Use Case ID** | **REQ-1** | **REQ-2** | **REQ-3** | **REQ-4** | **REQ-5** | **REQ-6** | **REQ-7** | **REQ-8** | **REQ-9** |
| **UC1** | X | X |  |  |  |  |  |  |  |
| **UC2** |  |  | X |  |  |  |  |  |  |
| **UC3** |  |  |  |  | X |  |  |  |  |
| **UC4** |  |  |  | X |  |  |  |  |  |
| **UC5** |  |  |  |  |  | X |  |  |  |
| **UC6** |  |  |  |  |  |  | X |  |  |
| **UC7** |  |  |  |  |  |  |  | X |  |
| **UC8** |  |  |  |  |  |  |  |  | X |

## Detailed Use Case Descriptions

1. Use Case ID / Name: **UC1: Add Transaction**

Related requirements: **REQ-1 & REQ-2**

Initiating actor: User

Actor’s goal: To add income or transactions

Participating actors: N/A

Preconditions:

* User wants to add an income or expenses or both

Postconditions:

* User have successfully added an income or expenses or both

Flow of events for main success scenario:

* User has an income or expenses to be added
* If the user wants to add expenses, user chooses the category of the expenses
* User executes the transaction, and transaction is saved into the system

1. Use Case ID / Name: **UC2: View Transaction History**

Related requirements: **REQ-3**

Initiating actor: User

Actor’s goal: To view all previous transactions

Participating actors: N/A

Preconditions:

* User wants to know all the transactions made in the past

Postconditions:

* User have successfully viewed all the transactions that was executed in the past
* User doesn’t see any transaction if none has been made in the past

Flow of events for main success scenario:

* User requests to the view transaction history
* The system searches the database for all existing transactions
* User views all previous transactions or is informed if no transactions exist

1. Use Case ID / Name: **UC3: Categorize Expenses**

Related requirements: **REQ-6**

Initiating actor: User

Actor’s goal: To group expenses into various categories

Participating actors: N/A

Preconditions:

* User wants to have an idea of the categories of items or services being spent on

Postconditions:

* User views their expenses in a categorized manner

Flow of events for main success scenario:

* User goes into the expenses section
* User has the option to group all expenses into different categories
* User views all the expenses grouped into their various categories

1. Use Case ID / Name: **UC4: Edit / Delete Transaction**

Related requirements: **REQ-5**

Initiating actor: User

Actor’s goal: To modify or delete a transaction

Participating actors: N/A

Preconditions:

* User wants to modify or delete a transaction made in the past

Postconditions:

* User successfully modifies or deletes the transaction

Flow of events for main success scenario:

* User goes into the transaction’s history section
* User has the option to filter transactions by (name, date, categories and amount range)
* User modifies or deletes the transaction of their choice

1. Use Case ID / Name: **UC5: View Report Summary**

Related requirements: **REQ-7**

Initiating actor: User

Actor’s goal: To view a summary of expenses in comparison to the income

Participating actors: N/A

Preconditions:

* User wants to view a summary of expenses in comparison to income

Postconditions:

* User views the total balance left and a summary of expense in comparison with income and can therefore be able to make analysis

Flow of events for main success scenario:

* User goes into the Report Summary section
* User has the option to view a brief or an extensive summary of expenses in comparison to income
* User views a brief or extensive summary

1. Use Case ID / Name: **UC6: View Total Balance**

Related requirements: **REQ-8**

Initiating actor: User

Actor’s goal: To view the total balance left which is Income minus expenses

Participating actors: N/A

Preconditions:

* Th user wants to have a quick view of their total balance

Postconditions:

* User views their total balance

Flow of events for main success scenario:

* User launches the Expense Tracker software
* User sees a quick view of their total balance

1. Use Case ID / Name: **UC7: Save / Load data from file**

Related requirements: **REQ-9**

Initiating actor: User

Actor’s goal: To save current transactions and settings to a file and load them later for continuity

Participating actors: N/A

Preconditions:

* The user has some transactions or data in the system they want to save for future use
* The user has the necessary permissions to access file storage

Postconditions:

* If saving, the user's data is successfully written to a specified file
* If loading, the system retrieves data from the specified file and updates the current state with the loaded data
* In case of file errors (e.g., file not found, corruption), the user is notified with appropriate error messages

Flow of events for main success scenario:

Save Data:

* User selects the **Save Data** option
* System prompts user to choose file location and name
* System saves current data to the specified file and confirms success

Load Data:

* User selects the **Load Data** option
* System prompts user to select a file
* System loads data from the selected file and confirms success.

1. Use Case ID / Name: **UC8: User Authentication**

Related requirements: **REQ-10**

Initiating actor: User

Actor’s goal: To login and logout of the system securely

Participating actors: N/A

Preconditions:

* The user must have an existing account or create one

Postconditions:

* On successful login, the user gains access to the software
* On logout, the user has securely existed from the session, protecting sensitive data

Flow of events for main success scenario:

Login

* User launches the Expense Tracker Software
* User is prompted to enter login credentials (username and password)
* User enters their credentials
* If credentials are correct, user gains access to their account and is directed to the main dashboard
* If credentials are not correct, system displays error prompting the user to retry
* User selects the Forgot Password if they have forgotten their credentials or due to multiple failed login attempts. user then follows systems prompts to reset the password

Logout

* User selects the logout option
* System logs the user out and is redirected to the login screen

# References

[1] Prof. Dr. Kolja Eger, Software Engineering Lecture Slides, HAW Hamburg, Hamburg, 2024.