

Software Requirements Specification

for
Spendwise
Version 1

Prepared by Group 14
Syracuse University CSE 687
April 8, 2024

Version	Date	A/D/C (Add, Delete, Change)	Author	Document Section #	Description
--	April 8th, 2024	--	Group 14	All	Initial Revision

Table of Contents

1 Introduction.....	7
1.1 Purpose.....	7
1.2 Scope.....	7
1.3 Definitions.....	7
1.4 References.....	7
1.5 References.....	7
2 Overall Description.....	7
2.1 Product Perspective.....	7
2.1.1 Spendwise Block Definition Diagram.....	8
2.1.2 Account.....	8
2.1.3 Add Expense.....	8
2.1.4 Add Income.....	8
2.1.5 Admin.....	8
2.1.6 Customer.....	9
2.1.7 Dashboard.....	9
2.1.8 Edit Expense.....	9
2.1.9 Edit Income.....	9
2.1.10 Expense.....	9
2.1.11 Login.....	9
2.1.12 Logout.....	9
2.1.13 Navigation.....	9
2.1.14 Profile.....	10
2.1.15 Reminder.....	10
2.1.16 Setting.....	10
2.1.17 Signup.....	10
2.1.18 Spendwise Management Block:	10
2.2 Product Functions.....	10
2.2.1 Spendwise Use Case Diagram.....	11
2.3 Use Case Description.....	11
2.3.1 Customer.....	11
2.3.1.1 Properties.....	11
2.3.1.2 Relationships.....	11
2.3.2 Super Admin.....	11
2.3.2.1 Properties.....	11
2.3.2.2 Relationships.....	12
2.3.3 System.....	12
2.3.3.1 Properties.....	12
2.3.3.2 Children Summary.....	12
2.3.4 Account Sync.....	12
2.3.4.1 Requirements.....	13
2.3.4.2 Scenarios.....	13
2.3.4.2.1 Scenario.....	13
2.3.5 Accounts.....	13
2.3.5.1 Requirements.....	13
2.3.5.2 Scenarios.....	13
2.3.5.2.1 Scenario.....	13
2.3.6 Add Expense.....	14
2.3.6.1 Scenarios.....	14
2.3.6.1.1 Scenario.....	14
2.3.7 Add Income.....	14

2.3.7.1 Requirements.....	14
2.3.7.2 Scenarios.....	14
2.3.7.2.1 Scenario.....	14
2.3.8 Dashboard.....	14
2.3.8.1 Requirements.....	15
2.3.8.2 Scenarios.....	15
2.3.8.2.1 Scenario.....	15
2.3.9 Edit Expense.....	15
2.3.9.1 Requirements.....	15
2.3.9.2 Scenarios.....	15
2.3.9.2.1 Scenario.....	15
2.3.10 Edit Income.....	16
2.3.10.1 Requirements.....	16
2.3.10.2 Scenarios.....	16
2.3.10.2.1 Scenario.....	16
2.3.11 Expense.....	16
2.3.11.1 Requirements.....	16
2.3.11.2 Scenarios.....	17
2.3.11.2.1 Scenario.....	17
2.3.12 Invalid Details.....	17
2.3.12.1 Requirements.....	17
2.3.12.2 Scenarios.....	17
2.3.12.2.1 Scenario.....	17
2.3.13 Login.....	17
2.3.13.1 Requirements.....	18
2.3.13.2 Scenarios.....	18
2.3.13.2.1 Scenario.....	18
2.3.14 Logout.....	18
2.3.14.1 Requirements.....	18
2.3.14.2 Scenarios.....	18
2.3.14.2.1 Scenario.....	18
2.3.15 Navigation.....	18
2.3.15.1 Requirements.....	19
2.3.15.2 Scenarios.....	19
2.3.15.2.1 Scenario.....	19
2.3.16 Profile.....	19
2.3.16.1 Requirements.....	19
2.3.16.2 Scenarios.....	19
2.3.16.2.1 Scenario.....	19
2.3.17 Reminder.....	20
2.3.17.1 Requirements.....	20
2.3.17.2 Scenarios.....	20
2.3.17.2.1 Scenario.....	20
2.3.18 Save Details.....	20
2.3.18.1 Requirements.....	20
2.3.18.2 Scenarios.....	21
2.3.18.2.1 Scenario.....	21
2.3.19 Settings.....	21
2.3.19.1 Requirements.....	21
2.3.19.2 Scenarios.....	21
2.3.19.2.1 Scenario.....	21

2.3.20 Signup.....	21
2.3.20.1 Requirements.....	22
2.3.20.2 Scenarios.....	22
2.3.20.2.1 Scenario.....	22
2.3.21 Track.....	22
2.3.21.1 Requirements.....	22
2.3.21.2 Scenarios.....	22
2.3.21.2.1 Scenario.....	22
2.3.22 Dashboard Requirement Diagram.....	23
2.3.23 Accessibility.....	23
2.3.24 Budget Management.....	23
2.3.25 Data Backup.....	23
2.3.26 Data Persistence.....	23
2.3.27 Expense Management.....	23
2.3.28 Expense Categorization.....	23
2.3.29 Expense Editing.....	23
2.3.30 Expense Reports and Analytics.....	24
2.3.31 Functional.....	24
2.3.32 Requirement.....	24
2.3.33 Requirement3.....	24
2.3.34 Scalability.....	24
2.3.35 Security and Access Control.....	24
2.3.36 User Authentication.....	24
2.3.37 Usability.....	24
3 Specific Requirements.....	24
3.1 Overview.....	24
3.1.1 Add Expense Activity Diagram.....	25
3.1.2 Add Expense.....	25
3.1.3 Create Expense.....	25
3.1.4 Login.....	25
3.1.5 Open Dashboard.....	25
3.1.6 Save Expense.....	25
3.1.7 Add Expense System Requirement Diagram.....	26
3.1.8 Dashboard Accessibility.....	26
3.1.9 Input Validation for Expense Creation.....	26
3.1.10 Create New Expense Entry.....	26
3.1.11 Expense Storage.....	26
3.1.12 User Authentication.....	27
3.1.13 Add Expense.....	27
3.1.14 Edit Expense Activity Diagram.....	27
3.1.15 Edit Expense.....	27
3.1.16 Dashboard.....	27
3.1.17 Edit Expense.....	28
3.1.18 Login.....	28
3.1.19 Save Changes.....	28
3.1.20 Validate Changes.....	28
3.1.21 Edit Expense System Requirement Diagram.....	28
3.1.22 Dashboard Access for Editing.....	28
3.1.23 Expense Editing Interface.....	28
3.1.24 Persisting Expense Edits.....	29
3.1.25 User Authentication for Editing.....	29

3.1.26 Validate Expense Edits.....	29
3.1.27 Edit Expense.....	29
3.1.28 Login Activity Diagram.....	30
3.1.29 Login.....	30
3.1.30 DecisionNode.....	30
3.1.31 Display Dashboard.....	30
3.1.32 Enter UserId and Password.....	30
3.1.33 Invalid Information.....	30
3.1.34 System checks if the password is correct.....	31
3.1.35 Login System Requirement Diagram.....	31
3.1.36 Account Lockout Mechanism.....	31
3.1.37 Authentication Process.....	31
3.1.38 Dashboard Access upon Successful Login.....	31
3.1.39 Incorrect Login Handling.....	32
3.1.40 User Identification Entry.....	32
3.1.41 Login.....	32
3.2 Performance Requirement.....	32
3.2.1 Spendwise Management Block:	32
3.2.2 Concurrent User Handling.....	32
3.2.3 Real-Time Data Update.....	32
3.2.4 System Load Time.....	32
3.2.5 Transaction Processing Time.....	32
3.3 Design Constraints.....	33
3.3.1 Spendwise Management Block:	33
3.3.2 Browser Compatibility.....	33
3.3.3 Compliance with Financial Data Handling Regulations.....	33
3.3.4 Cross-Platform Compatibility.....	33
3.3.5 Mobile Responsiveness.....	33

1. Introduction

1.1. Purpose

This Software Requirements Specification (SRS) is intended to delineate software requirements for the customer's financial system. This SRS is intended to provide guidance to the developers of the system to implement required functionality, as well as the test team to develop appropriate Verification and Validation (V&V) plans and procedures required to demonstrate to the customer that the system was built to this specification.

1.2. Scope

This document specifies the requirements for the following capabilities.

1. Spendwise, including user interface and web application support.
2. Spendwise has user account management which allow users to securely register, login, and access their financial data. It will use encryption and access control to protect sensitive user information.
3. It has expense tracking which will enable manual entry and categorization of various expenses and automatically sync and fetch expenses from linked accounts and cards.
4. It has budgeting tools which allow users to set up budgets across different categories and time periods. It will also send notifications when approaching or exceeding budgeted amounts.
5. It has goal setting which has features to define and monitor progress on financial goals like saving for a trip or paying off debt.

1.3. Definitions

Table 1 Acronyms and Definitions

BDD	Block Definition Diagram
SRS	Software Requirement Specification
UML	Unified Modeling Language
V&V	Verification and validation

1.4. References

1. IEEE Std 830-1998 - IEEE Recommended Practice for Software Requirements Specifications - Revision of IEEE Std 830-1993

1.5. References

This document follows the recommended format specified in IEEE Std 830-1998 IEEE Recommended Practice for Software Specifications. For Section 3, the specific template A.5 for organizing information by feature is followed.

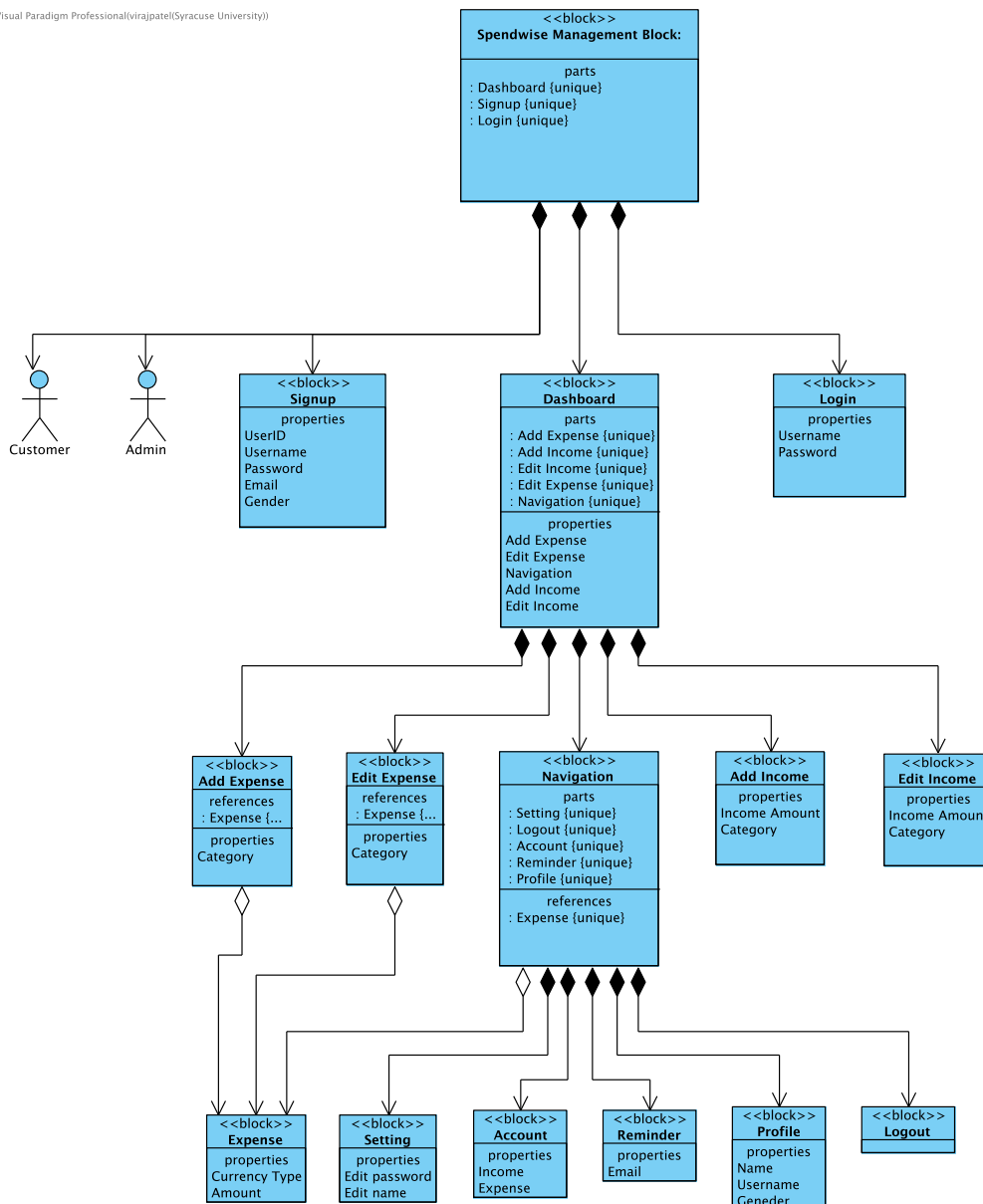
2. Overall Description

2.1. Product Perspective

Spendwise is a cutting-edge application that aims to transform the management of personal finances. With an emphasis on the needs of the user, this document specifies an extensive feature set and functionalities to enable users to manage their money wisely. The document describes capabilities that are hosted by a separately defined system. Figure 1 System Block Diagram shows the system overview, using a Unified Modeling Language (UML) Block Definition Diagram (BDD).

2.1.1. Spendwise Block Definition Diagram

Visual Paradigm Professional(virajpatel\Syracuse University)



2.1.2. Account

- Manages financial accounts with properties for Income and Expense.

2.1.3. Add Expense

References a unique Expense block and holds a property for the expense Category.

2.1.4. Add Income

Manages income details with properties like Income Amount and Category.

2.1.5. Admin

ID: AC04

This is likely a role within the system with unique properties and capabilities. The Admin typically has higher-level access to the system, which may include management of users, control over system settings, oversight of financial transactions, and access to comprehensive reports. The block definition diagram does not specify the exact properties or operations of the Admin, but given the context, one can assume the Admin has a broad range of controls within the application.

2.1.6. Customer

ID: AC01

This represents the end user of the system, and as a unique property, it suggests that individual customer data is distinctly managed within the system. The Customer role usually includes the ability to perform operations such as logging in, signing up, managing personal financial data, tracking expenses and income, and possibly setting reminders.

2.1.7. Dashboard

- Acts as the main interface, with parts for adding and editing expenses and income, navigating through the app, and includes unique operations like Profile(), Settings(), Accounts(), Reminder(), Expense(), and Logout().

2.1.8. Edit Expense

- Allows modification of expenses and references a unique Expense block, with a property for expense Category.

2.1.9. Edit Income

- Similar to Edit Expense, it enables editing income details with properties for Income Amount and Category.

2.1.10. Expense

- Manages expense details with properties like Currency Type and Amount.

2.1.11. Login

- Handles user sign-in with properties for Username and Password.

2.1.12. Logout

- This block likely handles the user sign-out process.

2.1.13. Navigation

Manages app navigation with operations to access Profile, Settings, Accounts, Reminder, Expense, and Logout. It includes parts and references related to these functions.

2.1.14. Profile

- Manages user profile information with properties like Name, Username, and Gender.

2.1.15. Reminder

- Handles reminders with an Email property.

2.1.16. Setting

- Allows users to edit their password and name.

2.1.17. Signup

- Manages user registration with properties like User ID, Username, Password, Email, and Gender.

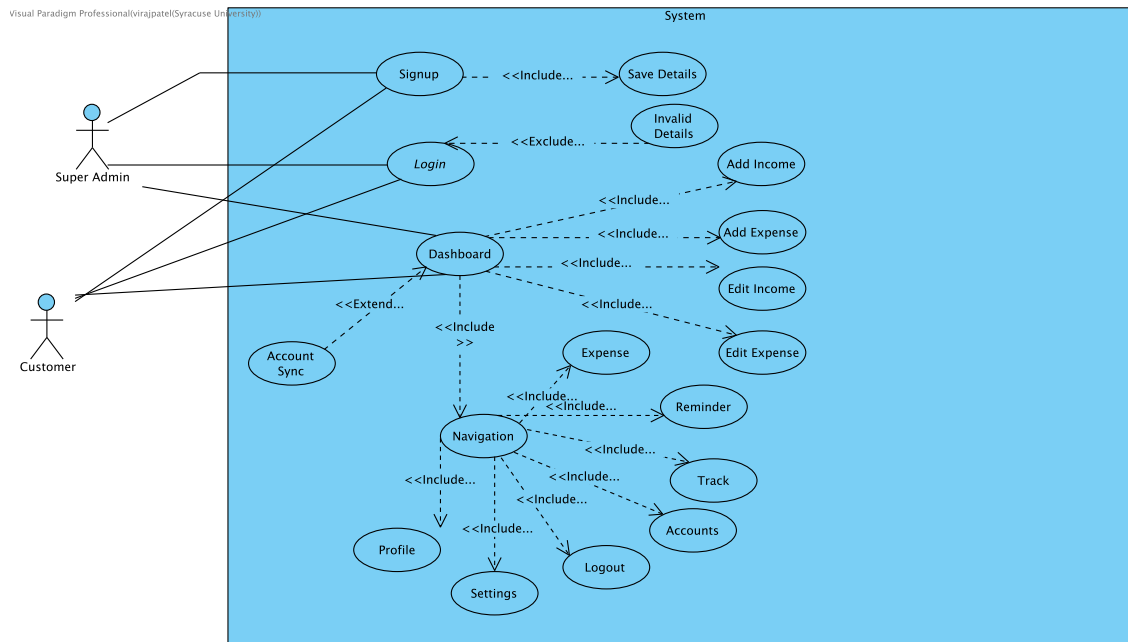
2.1.18. Spendwise Management Block:

- Central management block with operations like Login(), Signup(), and Dashboard(), indicating functions related to user access and main interface interaction.
- Contains parts that reference unique instances of Signup, Dashboard, Login, Admin, and Customer components.

2.2. Product Functions

The following use case diagram depicts the users of the system, and the intended way in which they will interact with the system.

2.2.1. Spendwise Use Case Diagram



2.3. Use Case Description

2.3.1. Customer

ID: AC01

This represents the end user of the system, and as a unique property, it suggests that individual customer data is distinctly managed within the system. The Customer role usually includes the ability to perform operations such as logging in, signing up, managing personal financial data, tracking expenses and income, and possibly setting reminders.

2.3.1.1. Properties

Abstract	false
Leaf	false
Root	false

2.3.1.2. Relationships

Relationship	From	To
unnamed	Customer	Signup
unnamed	Customer	Login
unnamed	Customer	Dashboard
unnamed	Spendwise Management Block:	Customer

2.3.2. Super Admin

ID: AC02







the person who has whole access to the app and who has created this app

2.3.2.1. Properties

Abstract	false
Leaf	false

Root	false
------	-------

2.3.2.2. Relationships



















Relationship	From	To
unnamed	 Super Admin	 Signup
unnamed	 Super Admin	 Login
unnamed	 Super Admin	 Dashboard

2.3.3. System

2.3.3.1. Properties

Abstract	false
Leaf	false
Root	false

2.3.3.2. Children Summary

Name	Description
 Account Sync	Keeps user data consistent across platforms.
 Accounts	Provides a view and management options for all of the user's financial accounts within the system.
 Add Expense	User can log expense details.
 Add Income	User can add income details.
 Dashboard	Central interface for managing finances after login.
 Edit Expense	Users can alter financial records.
 Edit Income	Users can alter financial records.
 Expense	Allows users to record and manage their expense transactions.
 Invalid Details	Handles scenarios where user input does not meet the system's validation criteria.
 Login	Authenticates registered users into the system.
 Logout	Terminates the user session and secures the account against unauthorized access.
 Navigation	Provides pathway to various sections.
 Profile	Allows users to update personal information.
 Reminder	Alerts users about upcoming financial events or tasks.
 Save Details	Captures and stores user details or transaction records in the system.
 Settings	Users can adjust their account settings.
 Signup	Allows new users to create an account by providing necessary details.
 Track	Enables users to monitor the activity and performance of their financial accounts.

2.3.4. Account Sync



ID: UC18

Keeps user data consistent across platforms.

Justification	Data synchronization across devices ensures that users have access to their up-to-date financial data from any device, improving the system's usability and the user's experience.
Level	N/A

Complexity	N/A
Use Case Status	N/A
Implementation Status	N/A
Preconditions	User has account data to sync.
Post-conditions	Sunny Day: The system successfully synchronizes the account with external data sources. Rainy Day: The synchronization fails due to network issues or incorrect configuration.
Author	N/A
Assumptions	N/A

2.3.4.1. Requirements

Name	ID	Kind
 Function for syncing data.	UC18.REQ001	
 Success and error handling for sync process.	UC18.REQ002	

2.3.4.2. Scenarios

2.3.4.2.1. Scenario

1. User initiates "Account Sync".
2. System confirms completion.



2.3.5. Accounts

ID: UC15

Provides a view and management options for all of the user's financial accounts within the system.

Justification	It is necessary to have a centralized location for users to manage all their accounts, facilitating better financial organization.
Level	N/A
Complexity	N/A
Use Case Status	N/A
Implementation Status	N/A
Preconditions	User is logged in and has navigated to the "Accounts" section.
Post-conditions	Sunny Day: The user successfully adds a new account or modifies an existing one. All financial details across accounts are accurately reflected in real-time on the dashboard or within the accounts section. Rainy Day: The user's attempt to add or modify an account fails because of issues such as reaching the maximum number of allowed accounts, providing incorrect account details, or encountering a service interruption.
Author	N/A
Assumptions	N/A

2.3.5.1. Requirements

Name	ID	Kind
 The system must aggregate all user financial accounts in one view.	UC15.REQ001	
 The system must provide options for adding, removing, or modifying accounts.	UC15.REQ002	

2.3.5.2. Scenarios

2.3.5.2.1. Scenario

1. User navigates to "Accounts".
2. System displays all associated accounts.

3. User manages accounts as needed.

2.3.6. Add Expense

ID: UC04

User can log expense details.

Justification	Logging expenses is crucial for users to monitor their spending patterns. It helps in budgeting and financial analysis, enabling users to make informed financial decisions.
Level	N/A
Complexity	N/A
Use Case Status	N/A
Implementation Status	N/A
Preconditions	User is logged in and on the "Dashboard".
Post-conditions	Sunny Day: The user adds income or expense details correctly, and the system updates the records accordingly. Rainy Day: The user enters invalid data that the system cannot process.
Author	N/A
Assumptions	N/A

2.3.6.1. Scenarios

2.3.6.1.1. Scenario

1. User selects "Add Expense" from the "Dashboard".
2. User enters expense details and submits.



2.3.7. Add Income

ID: UC03

User can add income details.

Justification	The ability to add income records is necessary for users to track their earnings. It's essential for financial management and planning, allowing users to understand their financial inflow.
Level	N/A
Complexity	N/A
Use Case Status	N/A
Implementation Status	N/A
Preconditions	User is logged in and on the "Dashboard".
Post-conditions	Sunny Day: The user adds income or expense details correctly, and the system updates the records accordingly. Rainy Day: The user enters invalid data that the system cannot process.
Author	N/A
Assumptions	N/A

2.3.7.1. Requirements

Name	ID	Kind
 Form for entering income details.	UC03.REQ001	
 Persistence of income data in the system.	UC03.REQ002	

2.3.7.2. Scenarios



2.3.7.2.1. Scenario

1. User selects "Add Income" from the "Dashboard".
2. User inputs details and submits.



2.3.8. Dashboard

ID: UC02

Central interface for managing finances after login.

Justification	A dashboard is the main interface that allows users to quickly access various financial management features and provides an overview of their financial status, improving user experience and efficiency.
Primary Actors	 Customer ,  Super Admin
Level	N/A
Complexity	N/A
Use Case Status	N/A
Implementation Status	N/A
Preconditions	User is logged in.
Post-conditions	Sunny Day: The user accesses the dashboard and views the necessary information without any issues. Rainy Day: There's an error loading the dashboard due to a system malfunction or connection issue.
Author	N/A
Assumptions	N/A

2.3.8.1. Requirements

Name	ID	Kind
 Must display summary of user's financial data.	UC02.REQ001	
 Navigation to all major functionalities.	UC02.REQ002	

2.3.8.2. Scenarios

2.3.8.2.1. Scenario

1. User logs in and sees "Dashboard".
2. User can select from options like "Add Income", "Add Expense", etc.
3. User can navigate to other functionalities.



2.3.9. Edit Expense

ID: UC06

Users can alter financial records.

Justification	The possibility to edit financial records gives users the flexibility to correct or update transactions, ensuring that the financial data remains accurate and current.
Level	N/A
Complexity	N/A
Use Case Status	N/A
Implementation Status	N/A
Preconditions	User has financial records to edit.
Post-conditions	Sunny Day: The user edits the income or expense records successfully. Rainy Day: The record to be edited does not exist or is locked due to concurrent editing by another user.
Author	N/A
Assumptions	N/A

2.3.9.1. Requirements

Name	ID	Kind
 Edit options for each financial entry.	UC06.REQ001	
 Updated data reflects immediately.	UC06.REQ002	

2.3.9.2. Scenarios

2.3.9.2.1. Scenario

1. User chooses to edit an income or expense item.
2. User makes changes and saves them.



2.3.10. Edit Income

ID: UC05

Users can alter financial records.

Justification	The possibility to edit financial records gives users the flexibility to correct or update transactions, ensuring that the financial data remains accurate and current.
Level	N/A
Complexity	N/A
Use Case Status	N/A
Implementation Status	N/A
Preconditions	User has financial records to edit.
Post-conditions	Sunny Day: The user edits the income or expense records successfully. Rainy Day: The record to be edited does not exist or is locked due to concurrent editing by another user.
Author	N/A
Assumptions	N/A

2.3.10.1. Requirements

Name	ID	Kind
 Edit options for each financial entry.	UC05.REQ001	
 Updated data reflects immediately.	UC05.REQ002	

2.3.10.2. Scenarios

2.3.10.2.1. Scenario

1. User chooses to edit an income or expense item.
2. User makes changes and saves them.


2.3.11. Expense


ID: UC11

Allows users to record and manage their expense transactions.

Justification	Managing expenses is vital for users to understand their spending patterns and maintain financial discipline.
Level	N/A
Complexity	N/A
Use Case Status	N/A
Implementation Status	N/A
Preconditions	User is logged in and has chosen to enter an expense.
Post-conditions	Sunny Day: The user enters a new expense by providing all required details (e.g., amount, category, date), and the system records it correctly and updates the balance. Rainy Day: The user inputs an expense with invalid details (e.g., negative amount, future date that's not allowed), and the system rejects the entry with a clear explanation of what needs to be corrected.
Author	N/A
Assumptions	N/A

2.3.11.1. Requirements

Name	ID	Kind
 The system must have a feature to enter and save expense details.	UC11.REQ001	

Name	ID	Kind
 The system must validate the entered information for accuracy.	UC11.REQ002	

2.3.11.2. Scenarios

2.3.11.2.1. Scenario

1. User chooses to record a new "Expense".
2. User inputs the details of the expense.
3. System saves the expense information.



2.3.12. Invalid Details

ID: UC09

Handles scenarios where user input does not meet the system's validation criteria.

Justification	Managing invalid inputs is necessary to ensure the integrity of the system's data and guide users towards correct data entry.
Level	N/A
Complexity	N/A
Use Case Status	N/A
Implementation Status	N/A
Preconditions	N/A
Post-conditions	N/A
Author	N/A
Assumptions	N/A

2.3.12.1. Requirements

Name	ID	Kind
 The system must provide clear error messages that help the user correct their	UC09.REQ002	
 The system must validate user inputs against predefined rules.	UC09.REQ001	

2.3.12.2. Scenarios



2.3.12.2.1. Scenario

1. User submits details.
2. System attempts validation and fails.
3. User is notified of the invalid details.
4. User is prompted to correct the details.

2.3.13. Login


ID: UC01

Authenticates registered users into the system.

Justification	A login mechanism is essential for authenticating users and ensuring that access to sensitive financial data is protected. It also allows the system to maintain the integrity of user data by preventing unauthorized access.
Primary Actors	 Customer ,  Super Admin
Abstract	true
Level	N/A
Complexity	N/A
Use Case Status	N/A
Implementation Status	N/A

Preconditions	User has an existing account.
Post-conditions	Sunny Day: The Super Admin or Customer enters the correct credentials and gains access to the system. Rainy Day: The Super Admin or Customer enters incorrect credentials and cannot log in.
Author	N/A
Assumptions	N/A

2.3.13.1. Requirements

Name	ID	Kind
 Secure authentication process. Error handling for failed logins.	UC01.REQ001	

2.3.13.2. Scenarios

2.3.13.2.1. Scenario

1. User enters login credentials.
2. System checks credentials against stored data.
3. On success, access to "Dashboard" is granted.
4. On failure, "Invalid Details" message is shown.



2.3.14. Logout

ID: UC14

Terminates the user session and secures the account against unauthorized access.

Justification	Logging out is essential for maintaining account security, especially on shared or public devices.
Level	N/A
Complexity	N/A
Use Case Status	N/A
Implementation Status	N/A
Preconditions	User is logged in.
Post-conditions	Sunny Day: The user logs out of the system successfully. Rainy Day: The logout process is interrupted, potentially leaving the session vulnerable.
Author	N/A
Assumptions	N/A

2.3.14.1. Requirements

Name	ID	Kind
 The system must ensure that the session is completely terminated upon logout.	UC14.REQ002	
 The system must provide a clear and easily accessible logout option.	UC14.REQ001	

2.3.14.2. Scenarios

2.3.14.2.1. Scenario

1. User selects "Logout".
2. System ends the session and logs the user out.
3. User is redirected to the login screen.



2.3.15. Navigation

ID: UC10

Provides pathway to various sections.

Justification	An intuitive navigation system simplifies how users interact with the application, reduces the learning curve, and enhances the overall user experience by providing easy access to all parts of the system.
Level	N/A
Complexity	N/A
Use Case Status	N/A
Implementation Status	N/A
Preconditions	User is within the system.
Post-conditions	Sunny Day: The user navigates through different sections of the system smoothly. Rainy Day: The user attempts to access an unauthorized section or encounters a broken link.
Author	N/A
Assumptions	N/A

2.3.15.1. Requirements

Name	ID	Kind
 Access from various parts of the system.	UC10.REQ002	
 Intuitive navigation system.	UC10.REQ001	

2.3.15.2. Scenarios

2.3.15.2.1. Scenario

1. User utilizes "Navigation" menu.
2. User selects desired section.



2.3.16. Profile

ID: UC17

Allows users to update personal information.

Justification	Allowing users to update their profile information enables them to keep their personal data current, which is important for personalization and any communication between the system and the user.
Level	N/A
Complexity	N/A
Use Case Status	N/A
Implementation Status	N/A
Preconditions	User is logged in.
Post-conditions	Sunny Day: The user views or updates their profile/settings without any problems. Rainy Day: The user inputs invalid information, or the system encounters an error while saving the changes.
Author	N/A
Assumptions	N/A

2.3.16.1. Requirements

Name	ID	Kind
 Immediate update and storage of new user data.	UC17.REQ002	
 Profile editing feature.	UC17.REQ001	

2.3.16.2. Scenarios

2.3.16.2.1. Scenario

1. User selects "Profile" from "Navigation".
2. User updates information and saves changes.



2.3.17. Reminder

ID: UC12

Alerts users about upcoming financial events or tasks.

Justification	Reminders help users manage their financial commitments effectively, ensuring that important tasks are not overlooked.
Level	N/A
Complexity	N/A
Use Case Status	N/A
Implementation Status	N/A
Preconditions	User is logged in and has set at least one reminder.
Post-conditions	Sunny Day: The system sets up a reminder for the user based on the defined criteria (like upcoming payments or budget limits), and the user receives the reminder notification at the appropriate time. Rainy Day: The system fails to set or trigger a reminder due to reasons such as incorrect date/time settings by the user, lack of permissions for sending notifications, or a bug in the scheduling functionality.
Author	N/A
Assumptions	N/A

2.3.17.1. Requirements

Name	ID	Kind
 The system must allow users to set, edit, and delete reminders.	UC12.REQ001	
 The system must notify users reliably at the specified times.	UC12.REQ002	

2.3.17.2. Scenarios

2.3.17.2.1. Scenario

1. User sets up or receives a "Reminder".
2. System notifies the user at the specified time or event.
3. User takes action based on the reminder.



2.3.18. Save Details

ID: UC08

Captures and stores user details or transaction records in the system.

Justification	Persisting information in the system is crucial to maintain continuity of service and user experience.
Level	N/A
Complexity	N/A
Use Case Status	N/A
Implementation Status	N/A
Preconditions	User has entered details that need to be saved.
Post-conditions	User's details or transactions are saved and can be retrieved or accessed later.
Author	N/A
Assumptions	N/A

2.3.18.1. Requirements

Name	ID	Kind
 The system must ensure data integrity during save operations.	UC08.REQ001	
 The system must provide feedback to the user after saving.	UC08.REQ002	

2.3.18.2. Scenarios

2.3.18.2.1. Scenario

1. User enters details that require saving.
2. System validates and saves the details.
3. User receives confirmation of saved details.


2.3.19. Settings

ID: UC16

Users can adjust their account settings.

Justification	Providing a settings feature allows users to configure the system according to their preferences, which can include security settings, notification preferences, and other personalizations enhancing user satisfaction and security.
Level	N/A
Complexity	N/A
Use Case Status	N/A
Implementation Status	N/A
Preconditions	N/A
Post-conditions	Sunny Day: The user navigates to the settings menu, adjusts preferences such as notification settings, privacy options, or account details, and the changes are saved successfully. Rainy Day: The user attempts to save the changes in settings, but due to an error (like a network issue or data validation error), the changes are not saved, and the user receives an error message.
Author	N/A
Assumptions	N/A

2.3.19.1. Requirements

Name	ID	Kind
 Settings feature for account personalization.	UC16.REQ001	

2.3.19.2. Scenarios



2.3.19.2.1. Scenario

1. User selects "Settings" from "Navigation".
2. User changes settings as desired and saves.

2.3.20. Signup


ID: UC07

Allows new users to create an account by providing necessary details.

Justification	Signing up is the first interaction a user has with the system. It is necessary to capture user details for personalized service and secure access. By having a signup process, the system can establish a unique identity for each user, which is critical for individual financial management and security.
Primary Actors	 Customer ,  Super Admin
Level	N/A
Complexity	N/A
Use Case Status	N/A

Implementation Status	N/A
Preconditions	User is not already registered.
Post-conditions	Sunny Day: The Super Admin provides all required details correctly and successfully creates an account. Rainy Day: The Super Admin provides invalid details, and the system rejects the signup attempt.
Author	N/A
Assumptions	N/A

2.3.20.1. Requirements

Name	ID	Kind
 The system must provide a registration form. User inputs must be validated for format and uniqueness.	UC07.REQ001	

2.3.20.2. Scenarios

2.3.20.2.1. Scenario

1. User selects the "Signup" option.
2. User inputs required details.
3. System validates the information.
4. If validation passes, the system proceeds to "Save Details".
5. If validation fails, the system indicates "Invalid Details".



2.3.21. Track

ID: UC13

Enables users to monitor the activity and performance of their financial accounts.

Justification	Tracking is fundamental for users to stay informed about their financial health and to detect any irregularities promptly.
Level	N/A
Complexity	N/A
Use Case Status	N/A
Implementation Status	N/A
Preconditions	User is logged in and has financial data to track.
Post-conditions	Sunny Day: The user uses the tracking feature to review their income and expenses over a selected period, gaining insights into their financial habits with accurate and up-to-date information. Rainy Day: The user tries to track finances, but the system fails to retrieve or display the data due to a system error or misconfiguration, leading to incomplete or missing information.
Author	N/A
Assumptions	N/A

2.3.21.1. Requirements

Name	ID	Kind
 The system must provide financial tracking features, including graphs, statistics, or summaries.	UC13.REQ001	
 The system must update tracking data in real-time or near-real-time.	UC13.REQ002	

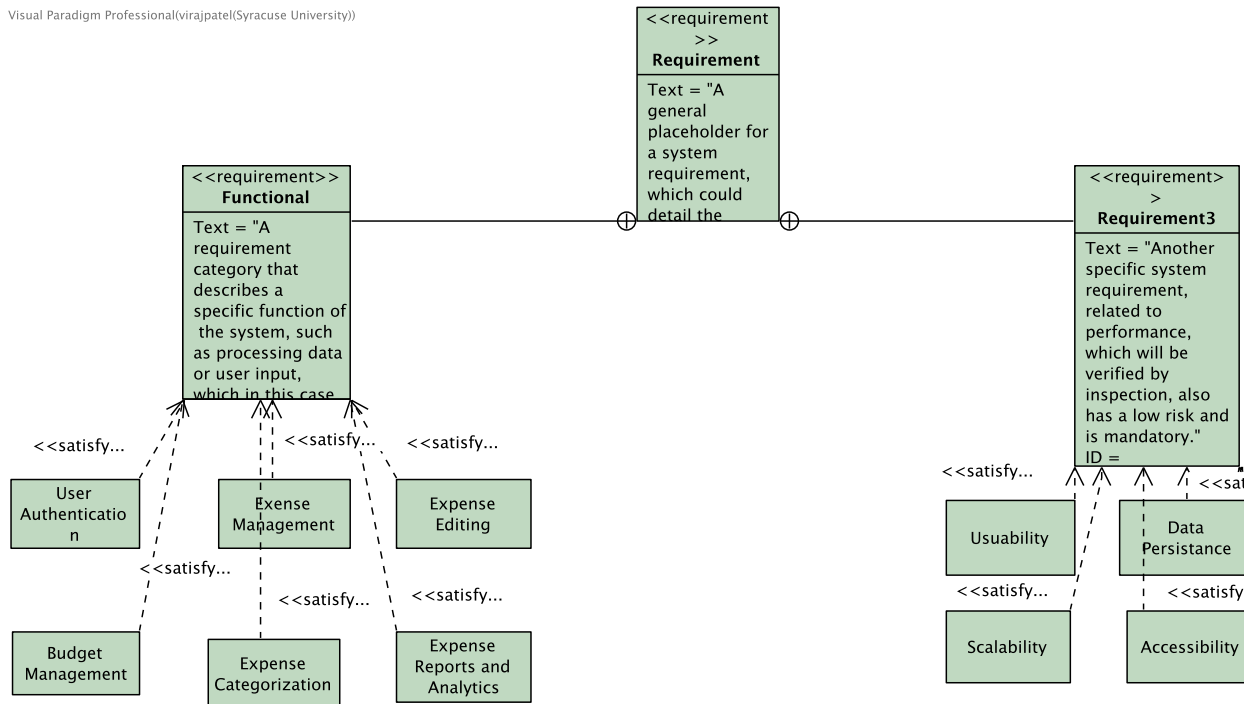
2.3.21.2. Scenarios

2.3.21.2.1. Scenario

1. User selects the "Track" option.
2. System presents a dashboard of financial activities and trends.
3. User reviews the information.

2.3.22. Dashboard Requirement Diagram

Visual Paradigm Professional(virajpatel(Syracuse University))



2.3.23. Accessibility

A non-functional requirement that the system should be usable by people with a wide range of abilities and disabilities.

2.3.24. Budget Management

A system feature that helps in planning, tracking, and controlling the budget within the organization.

2.3.25. Data Backup

A requirement ensuring that the system has a method for saving data in a way that it can be restored in case of loss or corruption.

2.3.26. Data Persistence

A system characteristic that ensures data remains stored and is not lost over time.

2.3.27. Expense Management

A module or set of functionalities that allows users to manage expenses within the system.

2.3.28. Expense Categorization

A feature for organizing expenses into different categories for better tracking and reporting.

2.3.29. Expense Editing

A feature allowing users to make changes or corrections to expense records.

2.3.30. Expense Reports and Analytics

A functionality that generates reports and insights based on expense data.

2.3.31. Functional

ID: REQ012

A requirement category that describes a specific function of the system, such as processing data or user input, which in this case must be verified through analysis, carries a low risk, and is mandatory.

2.3.32. Requirement

ID: REQ001

A general placeholder for a system requirement, which could detail the expected behavior or attributes of the system.

2.3.33. Requirement3

ID: REQ001.REQ002

Another specific system requirement, related to performance, which will be verified by inspection, also has a low risk and is mandatory.

2.3.34. Scalability

A non-functional requirement that specifies the system's ability to handle growing amounts of work or its potential to be enlarged to accommodate that growth.

2.3.35. Security and Access Control

A critical non-functional requirement focusing on protecting the system from unauthorized access and managing user permissions.

2.3.36. User Authentication

A system function that verifies the identity of a user, typically necessary for system security.

2.3.37. Usability

A non-functional requirement referring to how user-friendly the system is.

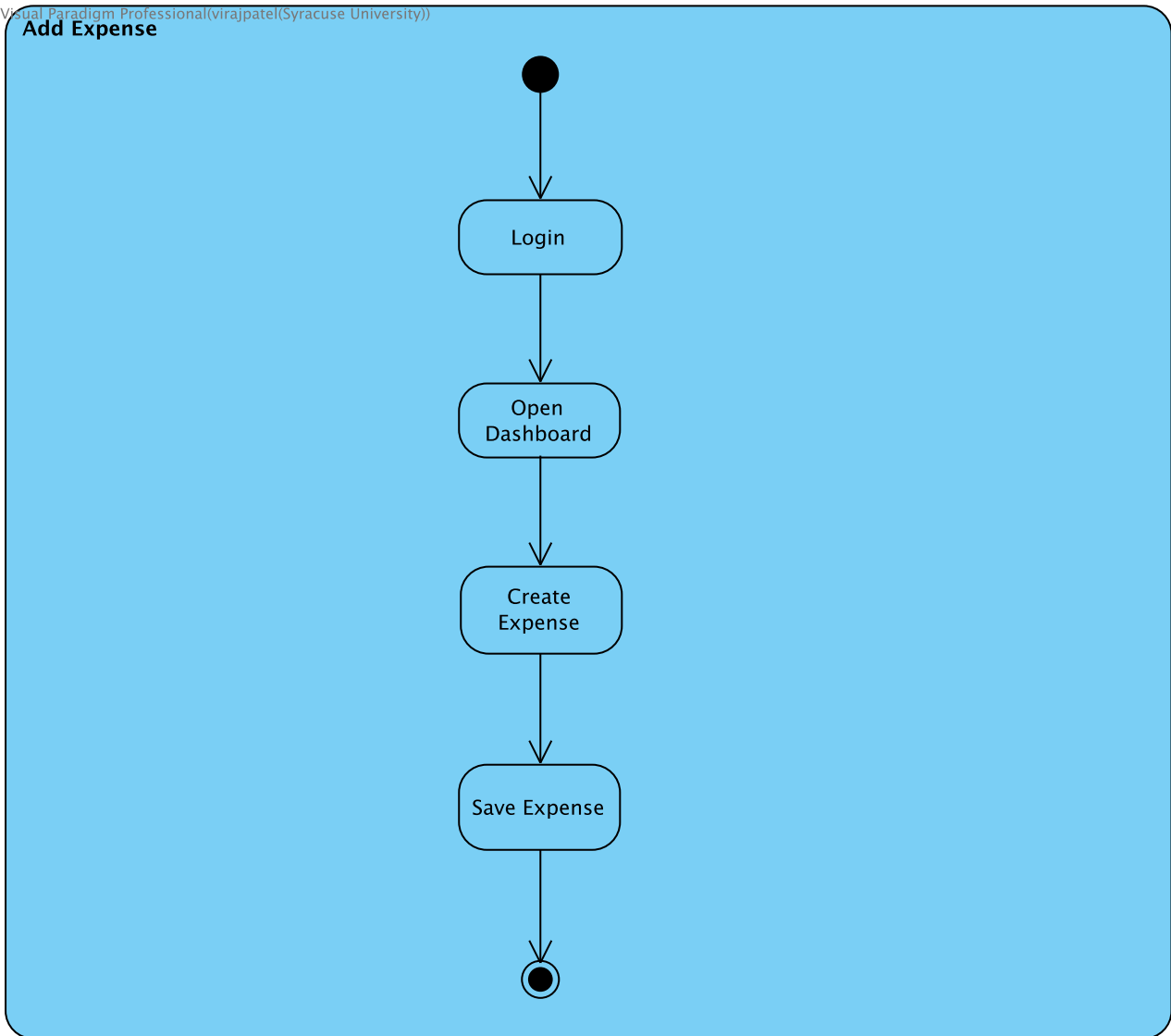
3. Specific Requirements

3.1. Overview

Spendwise is a comprehensive financial management tool designed to help users track expenses, manage income, and analyze spending habits. It offers dynamic data visualization through JavaScript libraries, ensuring insightful representations of financial data. Security is a top priority, with robust authentication mechanisms and encryption protocols safeguarding user information. The user experience is seamless, featuring an intuitive interface, real-time updates, and customization options. Spendwise aims to empower users in achieving their financial goals with clarity and confidence, emphasizing usability, security, and optimization. Interface Requirements

3.1.1. Add Expense Activity Diagram

Visual Paradigm Professional(virajpatel(Syracuse University))
Add Expense



3.1.2. Add Expense

3.1.3. Create Expense

This step involves the creation of a new expense entry, which is another action indicated by a rectangle.

3.1.4. Login

This step involves user authentication. It is denoted by a rounded rectangle, which often represents a process or an action that needs to be performed.

3.1.5. Open Dashboard

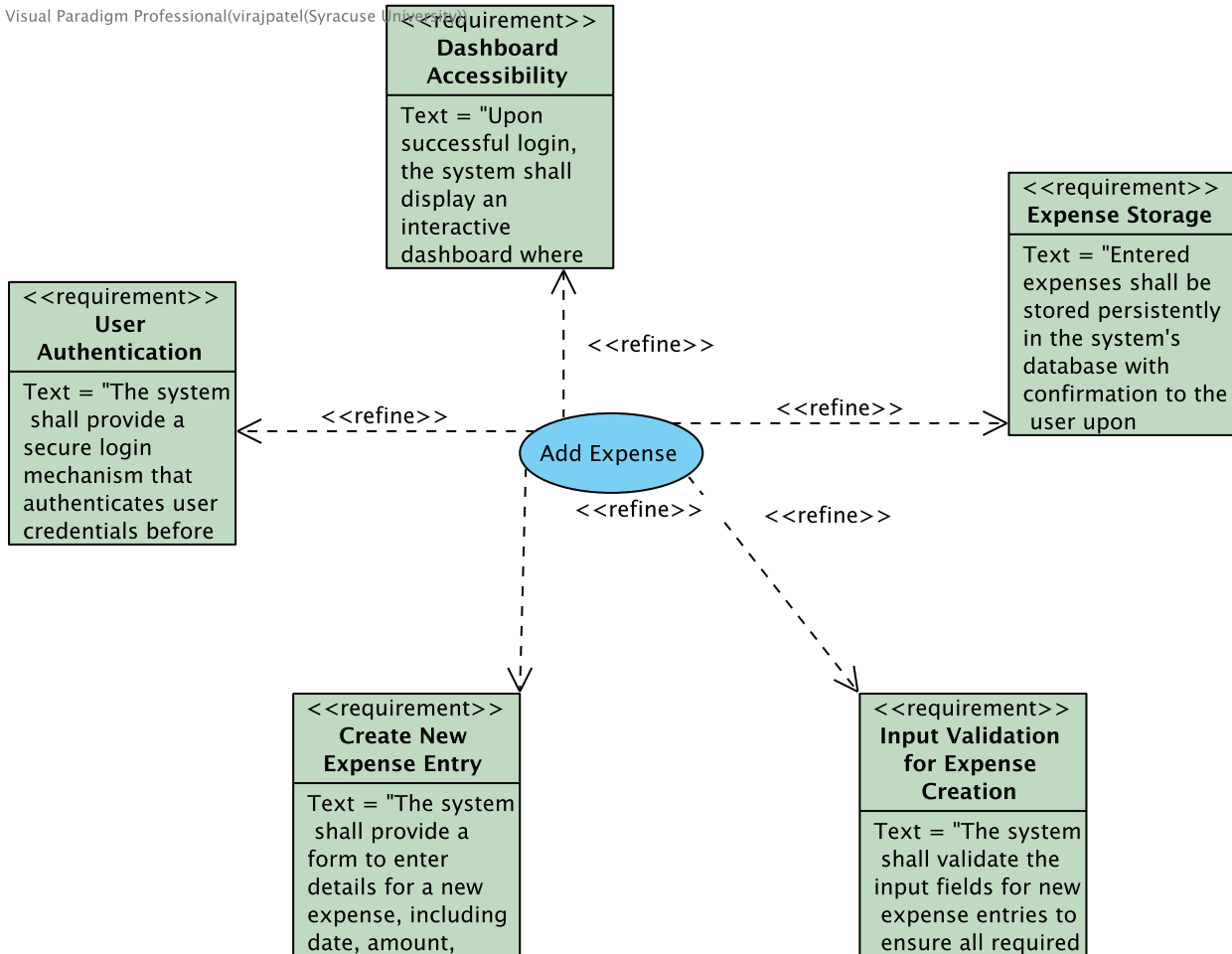
Once logged in, the next step is to open the dashboard. A standard rectangle is used to show individual steps or actions in a process.

3.1.6. Save Expense

After creating the expense, the action to save the expense is depicted. This is the final action step before the process ends.

3.1.7. Add Expense System Requirement Diagram

Visual Paradigm Professional(virajpatel(Syracuse University))



3.1.8. Dashboard Accessibility

ID: REQ014

Upon successful login, the system shall display an interactive dashboard where the user can navigate to various functionalities including adding new expenses.

3.1.9. Input Validation for Expense Creation

ID: REQ016

The system shall validate the input fields for new expense entries to ensure all required fields are filled with appropriate data formats.

3.1.10. Create New Expense Entry

ID: REQ015

The system shall provide a form to enter details for a new expense, including date, amount, category, and an optional description.

3.1.11. Expense Storage

ID: REQ017

Entered expenses shall be stored persistently in the system's database with confirmation to the user upon successful saving.

3.1.12. User Authentication

ID: REQ013

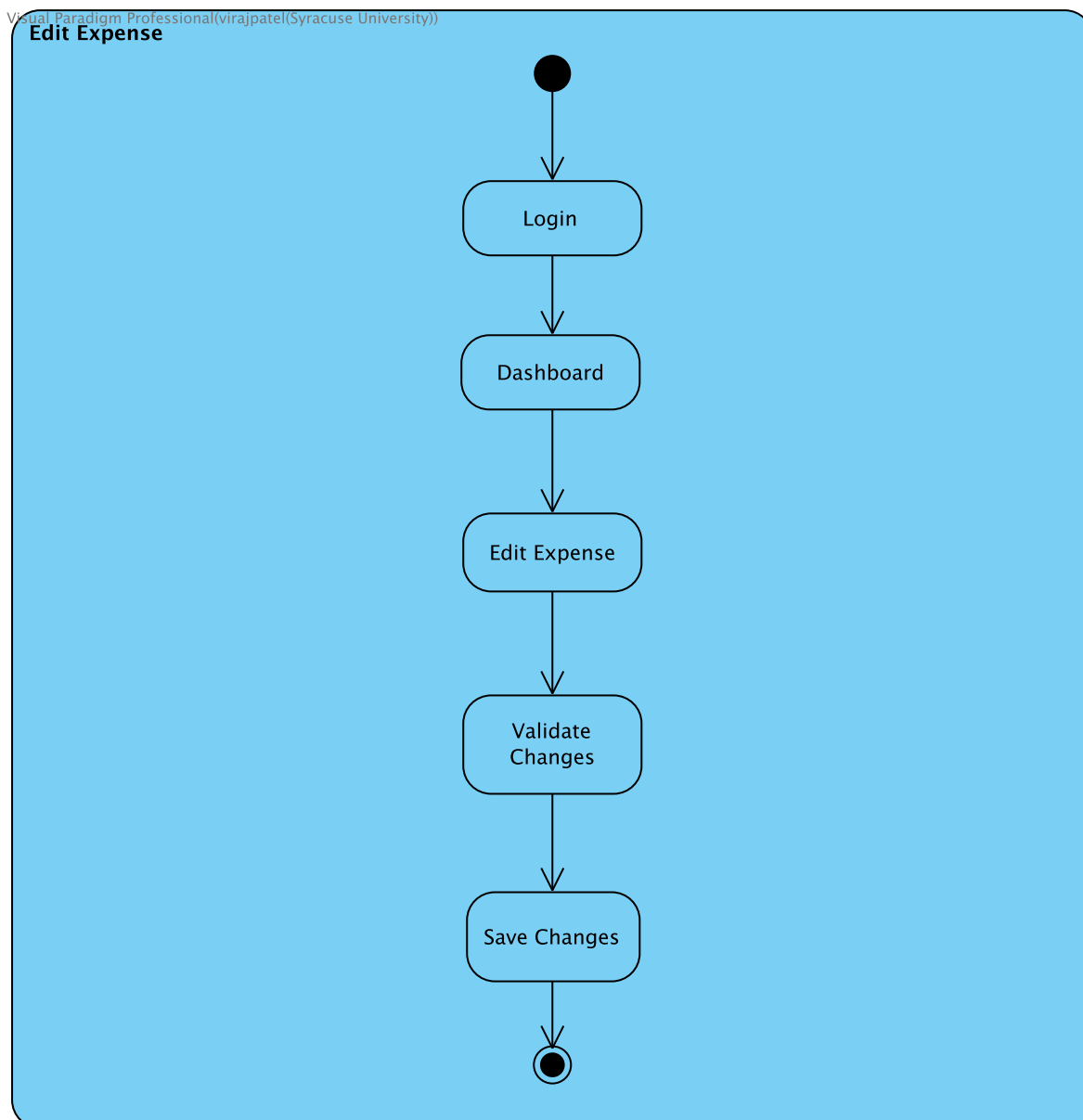
The system shall provide a secure login mechanism that authenticates user credentials before allowing access to the expense dashboard.

3.1.13. Add Expense

ID: UC04

User can log expense details.

3.1.14. Edit Expense Activity Diagram



3.1.15. Edit Expense

3.1.16. Dashboard

After logging in, the user arrives at the dashboard, which is likely where they can navigate to different functionalities. The rectangle here represents another action step.

3.1.17. Edit Expense

This action step allows the user to make changes to an existing expense. This step is crucial for updating any expense information.

3.1.18. Login

The first action step where a user would log into the system.

3.1.19. Save Changes

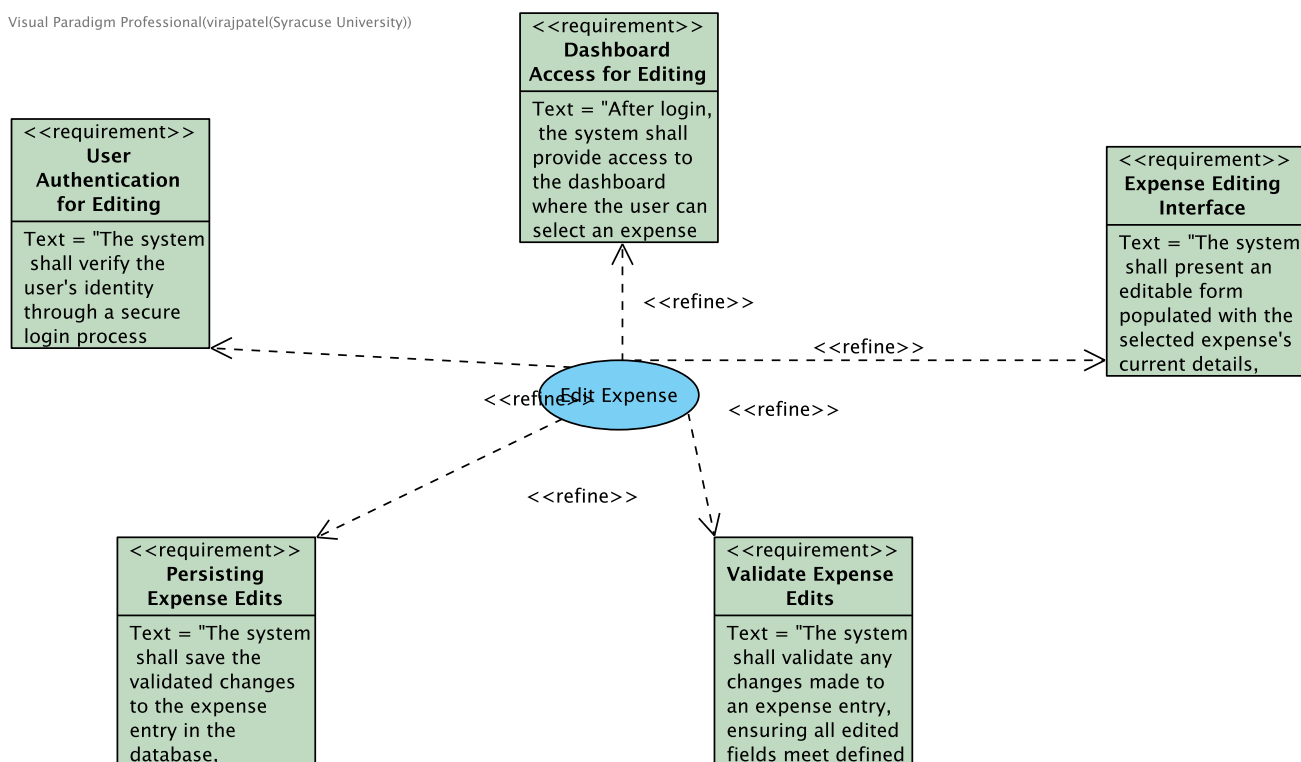
he final action step in this flowchart where the validated changes are saved to the system.

3.1.20. Validate Changes

Following the edits, this process involves checking or confirming that the changes are correct and can be committed. This could involve error checking or data validation steps.

3.1.21. Edit Expense System Requirement Diagram

Visual Paradigm Professional(virajpatel(Syracuse University))



3.1.22. Dashboard Access for Editing

ID: REQ020

After login, the system shall provide access to the dashboard where the user can select an expense entry to edit.

3.1.23. Expense Editing Interface

ID: REQ021

The system shall present an editable form populated with the selected expense's current details, allowing for modifications.

3.1.24. Persisting Expense Edits

ID: REQ019

The system shall save the validated changes to the expense entry in the database, overwriting the previous entry.

3.1.25. User Authentication for Editing

ID: REQ018

The system shall verify the user's identity through a secure login process before granting access to edit expense entries.

3.1.26. Validate Expense Edits

ID: REQ022

The system shall validate any changes made to an expense entry, ensuring all edited fields meet defined criteria for format and completeness.

3.1.27. Edit Expense

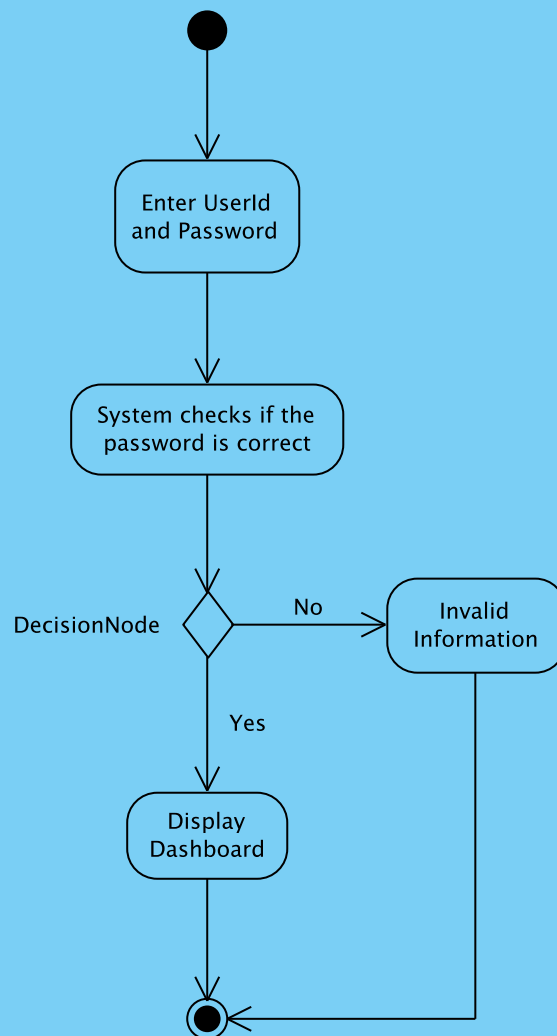
ID: UC06

Users can alter financial records.

3.1.28. Login Activity Diagram

Visual Paradigm Professional(virajpatel(Syracuse University))

Login



3.1.29. Login

3.1.30. DecisionNode

This diamond shape represents a decision point in the process. The system takes the input from the previous step and decides whether it is valid or not.

3.1.31. Display Dashboard

If the decision is 'Yes' (the credentials are correct), the system proceeds to display the dashboard to the user.

3.1.32. Enter UserId and Password

This step is where a user would input their credentials into the system.

3.1.33. Invalid Information

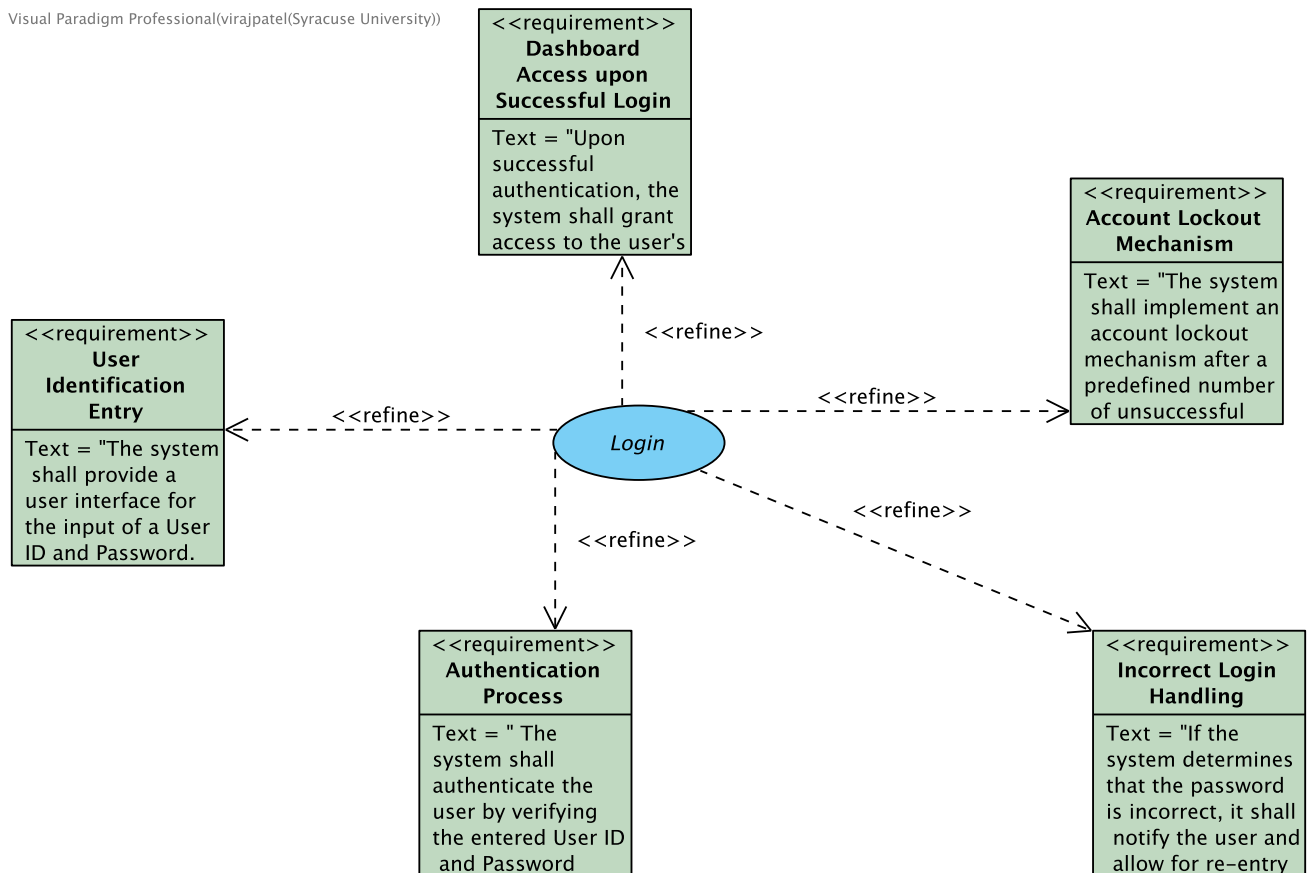
If the decision is 'No' (the credentials are not correct), the user is informed that the information is invalid.

3.1.34. System checks if the password is correct

This action represents the system processing the credentials to verify if the password matches the user ID.

3.1.35. Login System Requirement Diagram

Visual Paradigm Professional(virajpatel(Syracuse University))



3.1.36. Account Lockout Mechanism

ID: REQ026

The system shall implement an account lockout mechanism after a predefined number of unsuccessful login attempts to prevent unauthorized access.

3.1.37. Authentication Process

ID: REQ025

The system shall authenticate the user by verifying the entered User ID and Password against the stored credentials.

3.1.38. Dashboard Access upon Successful Login

ID: REQ024

Upon successful authentication, the system shall grant access to the user's dashboard.

3.1.39. Incorrect Login Handling

ID: REQ027

If the system determines that the password is incorrect, it shall notify the user and allow for re-entry of credentials.

3.1.40. User Identification Entry

ID: REQ023

The system shall provide a user interface for the input of a User ID and Password. This interface must ensure the confidentiality of the password input.

3.1.41. Login

ID: UC01

Authenticates registered users into the system.

3.2. Performance Requirement

3.2.1. Spendwise Management Block:

- Central management block with operations like Login(), Signup(), and Dashboard(), indicating functions related to user access and main interface interaction.
- Contains parts that reference unique instances of Signup, Dashboard, Login, Admin, and Customer components.

3.2.2. Concurrent User Handling

ID: REQ030

The system shall support up to 1000 concurrent users without significant degradation of performance (not exceeding a 10% increase in response time).

3.2.3. Real-Time Data Update

ID: REQ031

The system must update and display the dashboard data in real-time, with a latency no greater than 3 seconds after any transaction is made.

3.2.4. System Load Time

ID: REQ028

The system should load the Dashboard, Signup, and Login pages within 2 seconds over a standard broadband connection.

3.2.5. Transaction Processing Time

ID: REQ029

Financial transactions such as adding expenses or income should be processed and reflected in the user's account within 1 second.

3.3. Design Constraints

3.3.1. Spendwise Management Block:

- Central management block with operations like Login(), Signup(), and Dashboard(), indicating functions related to user access and main interface interaction.
- Contains parts that reference unique instances of Signup, Dashboard, Login, Admin, and Customer components.

3.3.2. Browser Compatibility

ID: REQ033

The web components of the system must be compatible with the latest versions of Chrome, Firefox, Safari, and Edge browsers.

3.3.3. Compliance with Financial Data Handling Regulations

ID: REQ035

The system design must comply with relevant financial data protection regulations, such as GDPR for European users and PCI DSS for payment processing.

3.3.4. Cross-Platform Compatibility

ID: REQ032

The system must be designed to operate seamlessly across different platforms, including Windows, macOS, iOS, and Android.

3.3.5. Mobile Responsiveness

ID: REQ034

The system's user interface must be responsive and adapt to different screen sizes, particularly for mobile devices.

