Pocket Expenses Tracker

Course Code- 246

Section- 007

Date: 02/03/2018

Submitted by:

Group Members

- Humairah Mansuri
- Neenu Shaji
- Rajvir Kaur
- Manpreet Kaur

Submitted to:

Bindu Goel

Problem Statement

Problem

A new Pocket Expense Tracker Application need to be developed to help the user to manage their income and expenses more efficiently. It is very difficult to maintain the data manually and perform calculations so eventually this android application can solve all those existing problems.

Need

There is a need for Information system to collect and track the income and expenses of the user in an easier and less time-consuming way. The app views this collected data in the Graphical form for the easier understanding of the user. The user can maintain the track of all the Expenses on a daily or weekly basis and can check them anytime.

SYSTEM VISION DOCUMENT

Problem Description

Recording and maintaining daily expense is quite difficult and inefficient. It is very difficult to maintain the track of our daily expenses and income sources. Usually to maintain the record, people used to write that data manually. We are living in an era of technology, and it has revolutionized our lives. The Pocket Expense Tracker application will do all the work that a personal finance manager used to do. It is a fully-featured and efficient finance software that can be carried around in our pocket.

Subsystems

1. Authentication Subsystem
    a. Existing user login
    b. New user registration

2. Client settings Subsystem
    a. Password Updating is handled separately
3. Income Subsystem
    a. Data is maintained, and user can view it in the form of charts
4. Expense Subsystem
    a. Track is maintained for the expenses on daily, weekly and monthly basis.
5. Balance Sheet Subsystem
    a. User can also download the charts

6. Notification Subsystem
    a. App notifies the user about it , if the limit exceeds.

Business Benefits

The deployment of the Pocket Expense Manager is expected to provide the following benefits:

- Helps track income and expenses efficiently.
- Helps store wide-ranging and flexible information about income and expenses.
- Ease of use saves time and money.

System Capabilities

The new system will enable clients to:

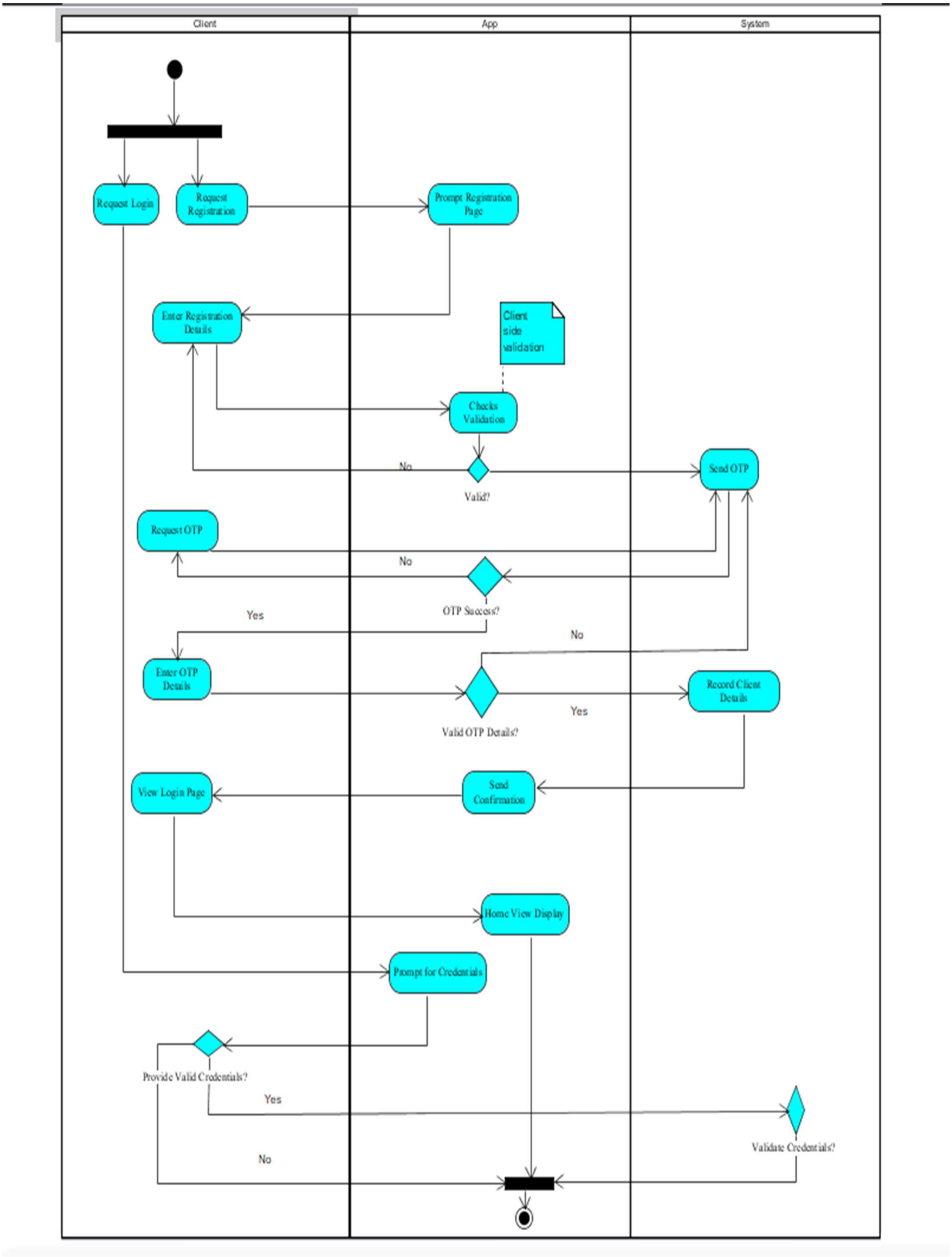
- Manage expenses and income sources efficiently.
- Can add new categories of expenses and income.
- Can choose default categories.
- Can add data through voice notes, camera, bar code scanner as well as through text fields.
- Can view income and expense statistics as charts on daily, weekly and monthly basis.
- Can set expense limit alarms which will warn the user about set expense amount.

WORKFLOWS

1.0 Authentication Subsystem

Allows the existing user to login and new user can register and create account.

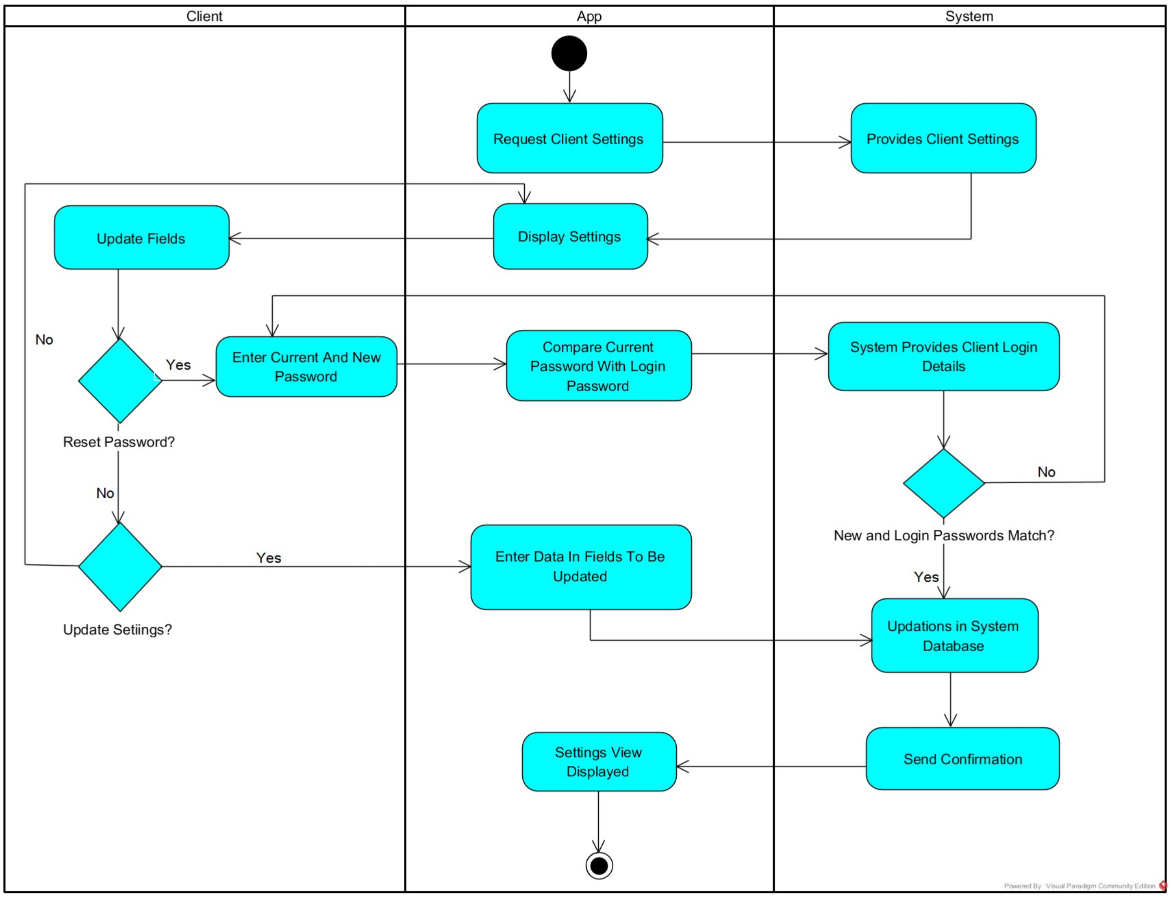
1.1 Client request to login , redirect to 1.13
1.2 Client request registration.
1.3 App prompts registration page.
1.4 Client enters registration details.
1.5 App validates details.
    1.5.1 If details are valid, App will redirect to 1.6.if details are invalid then app will redirect to 1.4.
1.6 App will send OTP to given number.
1.7 User can enter the OTP or can request for another one.
1.8 App validates the OTP.
    1.8.1 If it is valid then it will redirect to 1.9 or else it will redirect to 1.6.
1.9 System records user information.
1.10 Send confirmation for login.
1.11 App prompts user and generate welcome message to the Client .
1.12 Home Page Display
1.13 App prompts for Credentials
1.14 Client provides credentials
    1.14.1 If client provides credentials, proceed to 1.14; else proceed to 1.15
1.15 App passes credentials to be validated by the server
    1.15.1 If valid credentials, proceed to 1.15; else proceed to 1.13
1.16 App returns client to home view


Client
App
System
Request Login
Request Registration
Prompt Registration Page
Enter Registration Details
Client side validation
Checks Validation
No
Valid?
Send OTP
Request OTP
No
OTP Success?
Yes
No
Enter OTP Details
Valid OTP Details?
Yes
Record Client Details
View Login Page
Send Confirmation
Home View Display
Prompt for Credentials
Provide Valid Credentials?
Yes
Validate Credentials?
No

2.0 Client settings subsystem

Allows the client to update account settings through the mobile application.

2.1. App requests client settings from system
2.2. System provides client settings to the app
2.3. App provides view of client settings to the client
2.4. App prompts client to update client settings
    a. If client wants to reset password else go to 2.5
        i. Enter current and new passwords
        ii. Compare both passwords
        iii. System provides client login details
        iv. If new and login passwords match then go to 2.6, else go to 2.4.a.i
2.5. If client wants to edit settings go to 2.6 else go to 2.3
2.6. Enter data in fields to be updated
2.7. Update new information in the system
2.8. System updates modifications for the client’s account
2.9. System sends confirmation that settings have been updated
2.10. App brings client to home view



Client
App
System
Request Client Settings
Provides Client Settings
Display Settings
Update Fields
No
Yes
Enter Current And New Password
Compare Current Password With Login Password
System Provides Client Login Details
Reset Password?
No
No
New and Login Passwords Match?
Yes
Enter Data In Fields To Be Updated
Update Setiings?
Yes
Updations in System Database
Settings View Displayed
Send Confirmation
Powered By Visual Paradigm Community Edition

3.0 Income Subsystem

3.1. Client can add, modify, delete or view the Income.

   a. If client selects to add the Income, then app redirects to 3.2. If client selects to update the Income, then app redirects to 3.6. If client selects to delete the Income, then app redirects to 3.8. If client selects to view the Income, then app redirects to 3.7.

3.2. App gives client text fields to add the Income.

3.3. Client enters the record.

3.4. It goes to verification step.

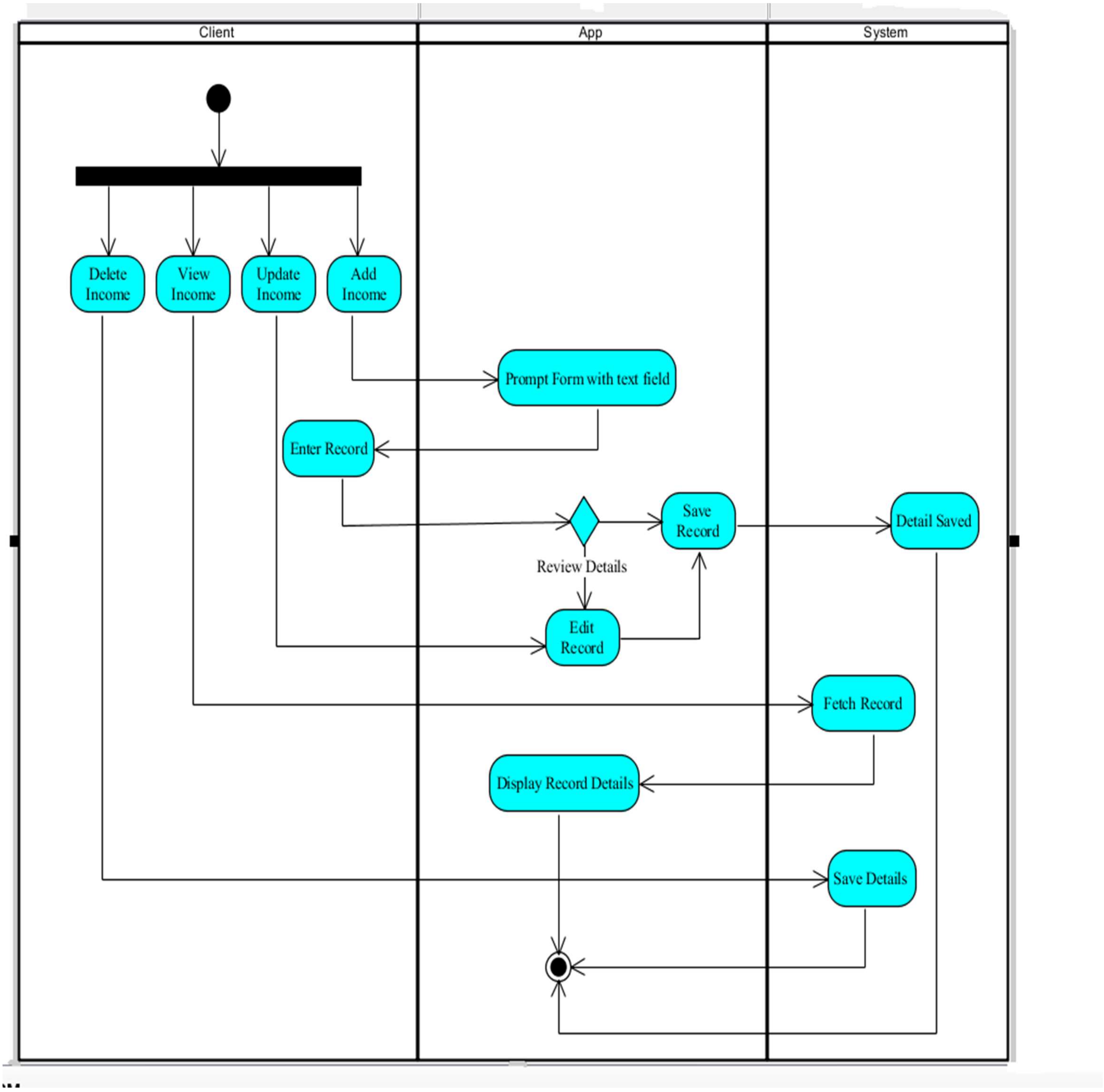
   a. If Client thinks that the detail is accurate then it saves the record and redirect to 3.5 or else, it goes for editing the record and again redirects to 3.4.

3.5. System saves the record and app displays the record.

3.6. App redirects to 3.4 if Client selects update Income.

3.7. App redirects to 3.5 if Clients selects to view Income.

3.8. If Client selects delete Income then the record is deleted and goes to the end node.


Client
App
System
Delete Income
View Income
Update Income
Add Income
Prompt Form with text field
Enter Record
Save Record
Review Details
Edit Record
Detail Saved
Fetch Record
Display Record Details
Save Details

4.0 Expense Subsystem

4.1. If client selects to add the expense, else go to 4.5

a. The four input methods are displayed, and the client chooses one.

1.1.a.1 If input method is text, display expense form and then go to 4.2.a else go to 4.1.a

1.1.a.2 If input method is voice, open microphone then go to 4.2.a, else go to 4.1.a

1.1.a.3 If input method is barcode, open camera then go to 4.2.a., else go to 4.1.a

1.1.a.4 If input method is picture, open camera to take picture of bill then go to 4.2.a, else go to 4.1.a

4.2. If client selects to update the expense, else go to 4.5

a. Edit the form

b. Save the form in app

c. Save the record in system, then go to 4.5

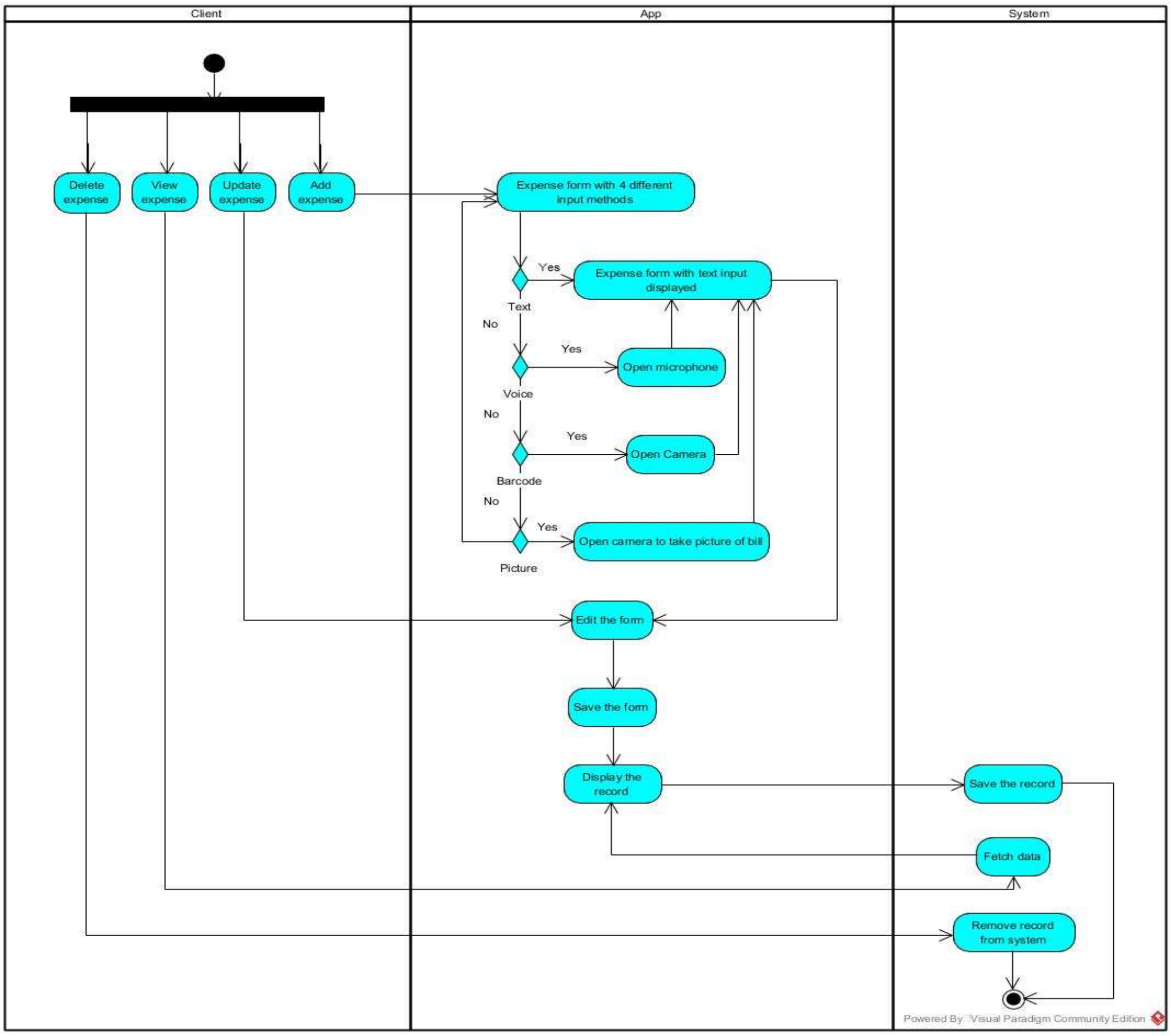
4.3. If the client selects to view the expense, else go to 4.5

a. Fetch data from the system

b. Display the data in app, then go to 4.5

4.4. If client selects to delete the expense, else go to 4.5

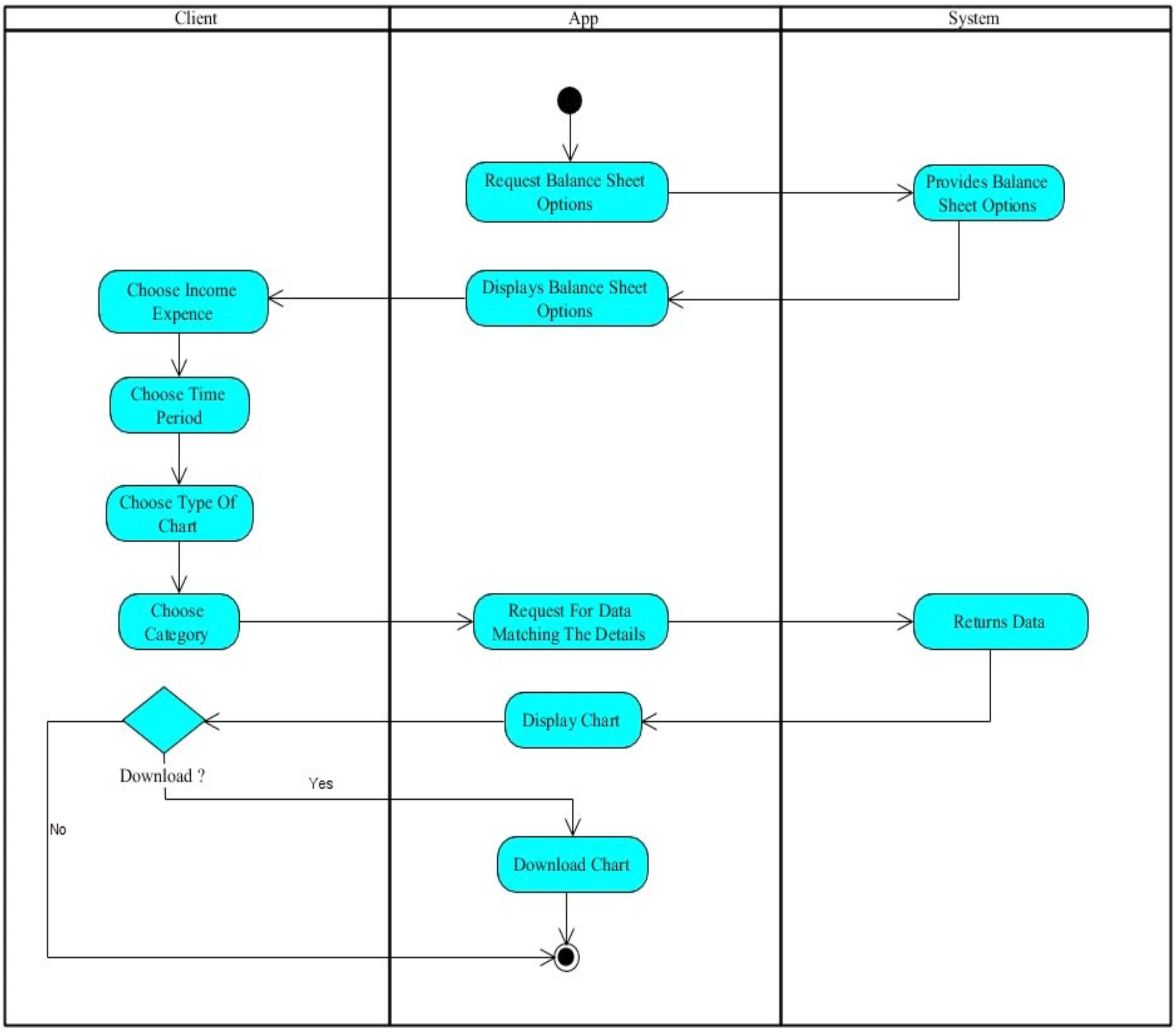
4.5. Returns to home screen in app



5.0 Balance Sheet Subsystem

Allows user to display income and expense statistics as different kinds of charts with specific parameters such as time-period and category.

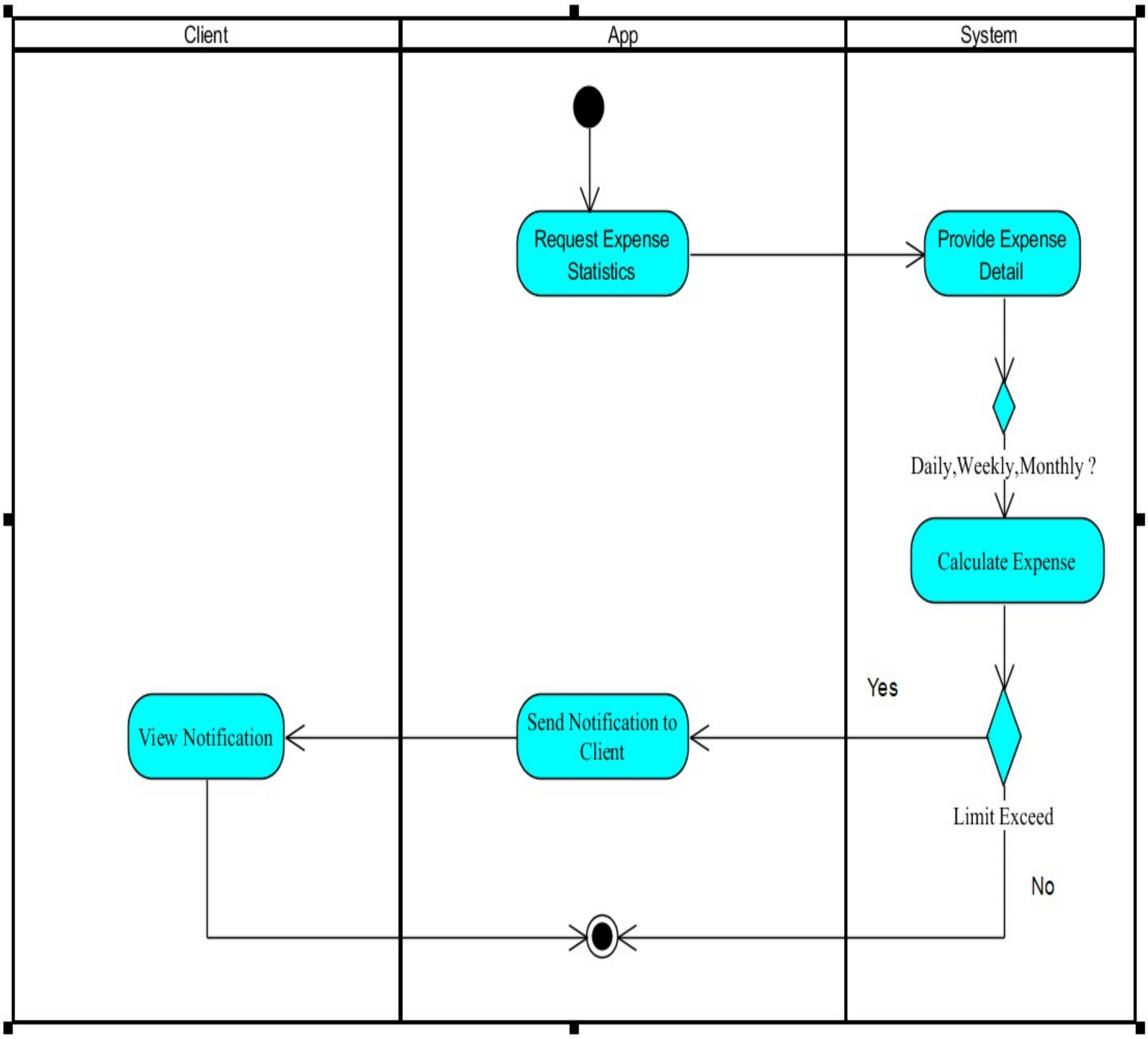
5.1 Client requests the Balance sheet options.
5.2 System provides Balance sheet options.
5.3 Application displays Balance sheet options.
5.4 Client chooses income or expense.
5.5 Client chooses specific time-period.
5.6 Client chooses the type of chart.
5.7 Client chooses the category of income or expense to be displayed.
5.8 App requests the system to return data with chosen parameters.
5.9 System returns data.
5.10 App displays the chart.
5.11 If Client chooses to download the chart, then go to 11.a.
   a. App downloads the chart.
5.12 End.



6.0 Notification Subsystem

This subsystem displays expense statistics to client when expense limit is exceeded on daily, weekly and monthly bases.

6.1 App requests Expense Statistics from System
6.2 System Provide Expense Details
6.3 System Calculate Expense on daily, weekly and monthly to App
6.4 App Checks expense limit
    6.4.1 If expense limit is exceeded, proceed 6.5 ;
    6.4.2 else end the process
6.5 Pushes notification to alert the client about expense statistics
6.6 Client Views his/her Notification



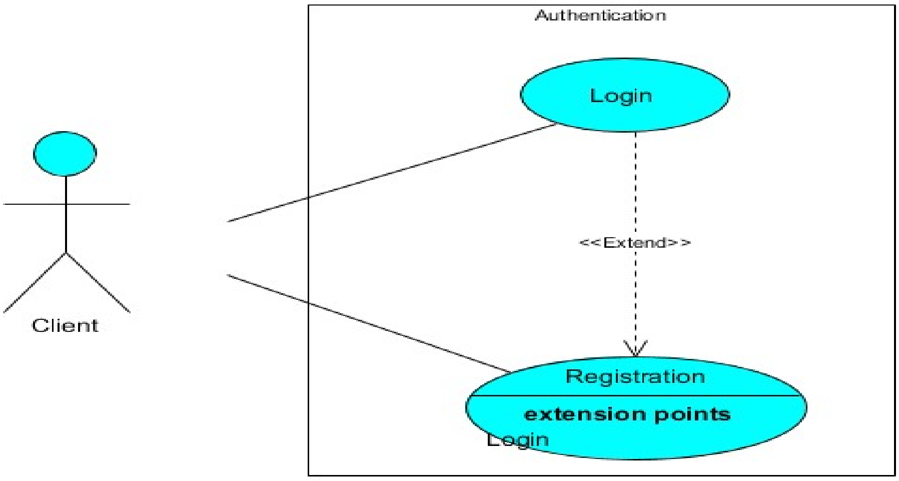
USE CASES

1.0 Authentication Subsystem

User	User’s goal and resulting use case
Client	Register with a new account to get access to application named as Pocket expense. Login to the existing system as the Client has created account before.

Use case	Brief use case description
Registration	Client Register for the first time to get access to pocket expense tracker
Login	Client Login into the application.

Authentication Subsystem	
Use cases	Users/Actors
Register	Client
Login	Client

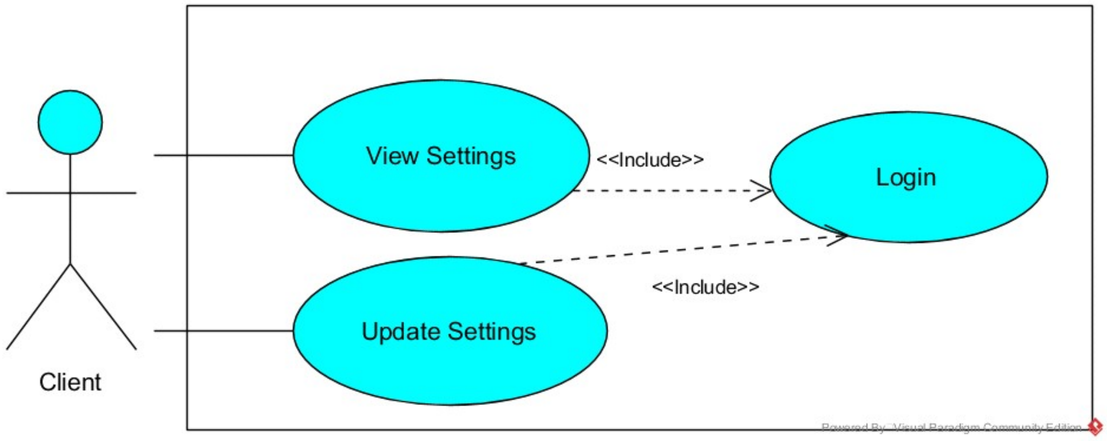


2.0 Client settings subsystem

User	User’s goals and resulting use cases
Client	View the Existing Account Details. Update the Acoount Information.

Use Case	Brief Use Case description
View Settings	Client decides to view his/her Acoount Settings.
Update Settings	Client decides to update his/her existing information and then modifies/updates the account information.

Client Settings Subsystem	
Use Case	Users/Actors
View Settings	Client
Update Settings	Client

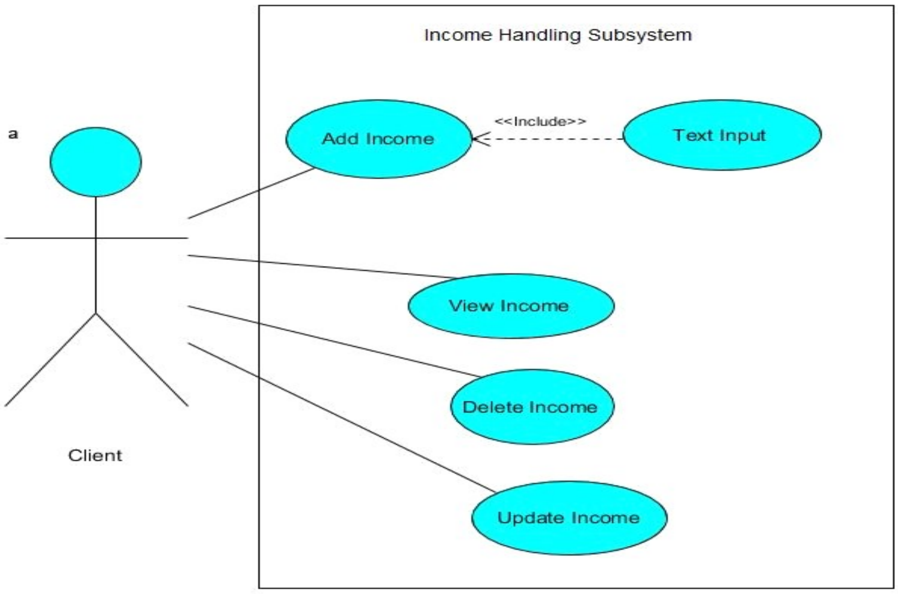


3.0 Income Subsystem

User	User’s goal and resulting use case
Client	Client adds income. Client can view the recorded data and can edit or delete the record accordingly.

Use case	Brief use case description
Add Income	Clients add the detail of the Income gained. Client can add Income category or payer by own or can use the default one.
Update Expense	Client can modify the information if he/she wants to do so.
Delete Expense	Client can delete the information if he/she wants to do so.
View Expense	Client can view the information if he/she wants to do so.

Income Subsystem	
Use cases	Users/Actors
Add Income	Client
Update Income	Client
Delete Income	Client
View Income	Client

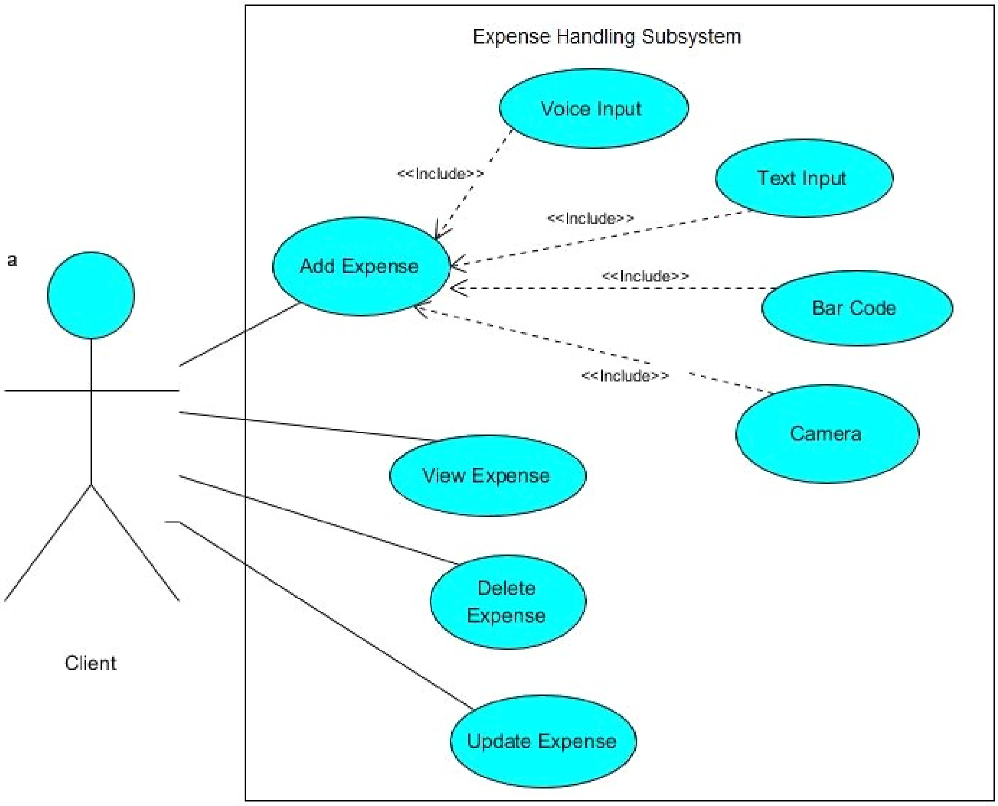


4.0 Expense Subsystem

User	User’s goal and resulting use case
Client	Client add expense with different inputs i.e. text, camera, QR Code, Voice notes. Client can view the recorded data and can edit or delete the record accordingly.

Use case	Brief use case description
Add Expense	Clients add the detail of the expense occurred. Client can add expense category or payee by own or can use the default one. This use case includes 4 other sub usecases. They are • Camera Entry • QR Code • Text Entry • Voice Entry
Update Expense	Client can modify the information if he/she wants to do so.
Delete Expense	Client can delete the information if he/she wants to do so.
View Expense	Client can view the information if he/she wants to do so.

Expense Subsystem	
Use cases	Users/Actors
Add Expense	Client
Update Expense	Client
Delete Expense	Client
View Expense	Client

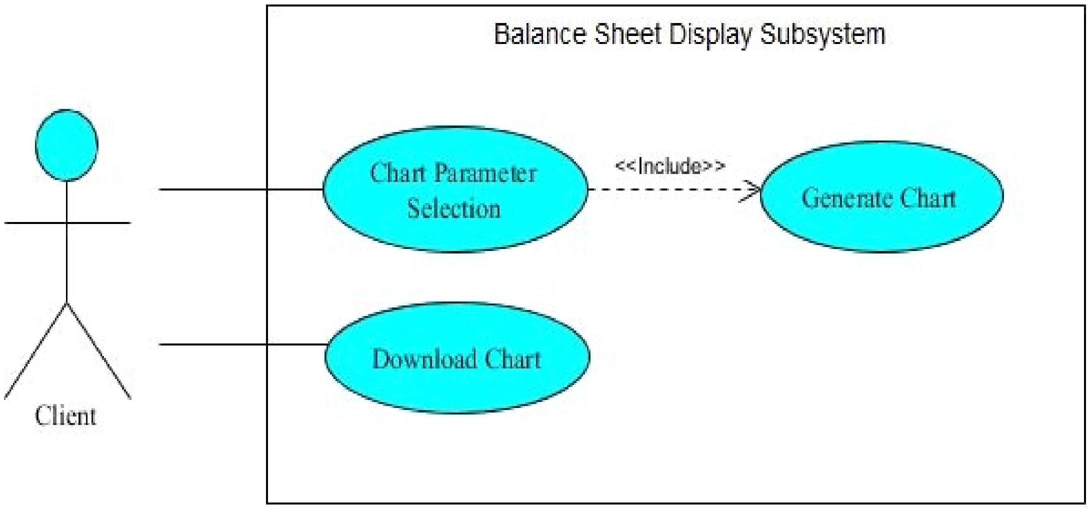

Expense Handling Subsystem
a
Client
Voice Input
Text Input
Add Expense
Bar Code
Camera
View Expense
Delete Expense
Update Expense
<<Include>>
<<Include>>
<<Include>>
<<Include>>

5.0 Balance Sheet Subsystem

User	User’s goal and resulting use case
Client	Selects the parameters for the balance sheet chart to be displayed. Downloads the displayed chart.
Database	Returns the data specified by the chosen parameters to generate the chart.

Use case	Brief use case description
Chart Parameter selection	Client/Actor selects the parameters for the chart to be displayed.
Generate chart	App generates the chart using the data returned by the database.
Download chart	App downloads the chart upon client request.

Balance Sheet Display Subsystem	
Use cases	Users/Actors
Chart Parameter selection	Client
Generate chart	Database
Download chart	Client

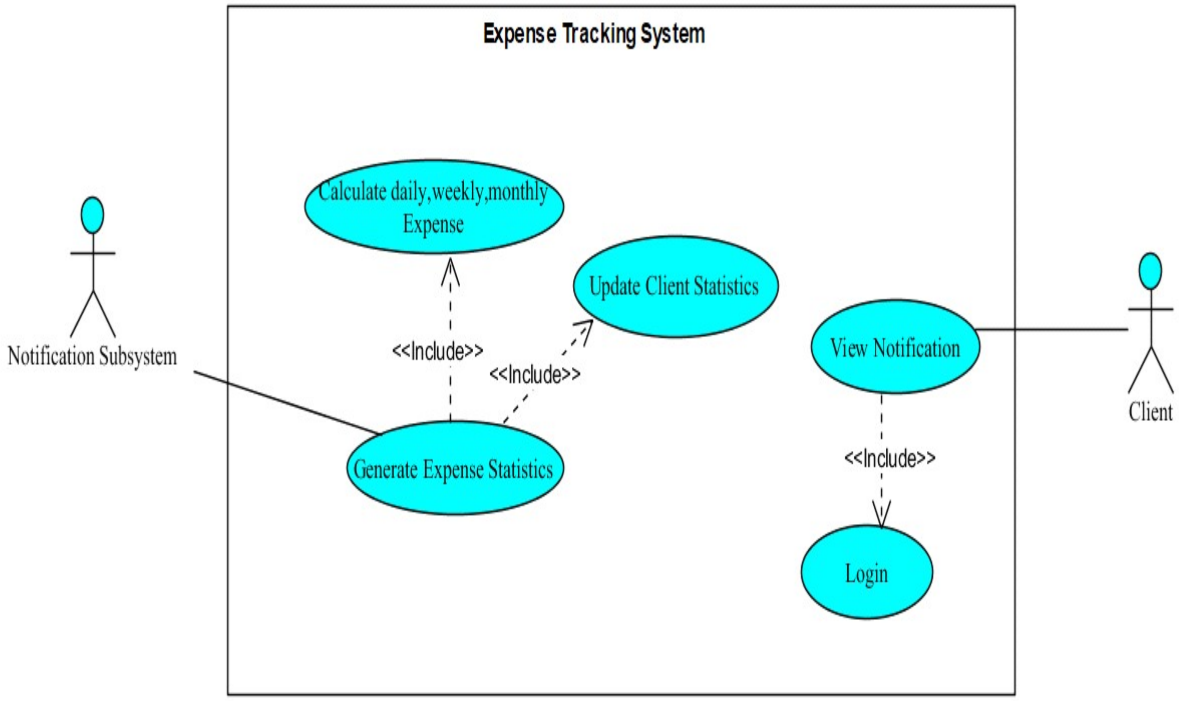


6.0 Notification Subsystem

User’s	User’s goals and resulting use cases
Notification Subsystem	Generate Expense details based on daily, weekly, monthly categories
Client	View Notification when login

Use Case	Brief Use Case description
Generate Expense Statistics	Generate Expense Statistics for client’s account
View Notification	Client view’s any alert distributed to his/her account

Notification Subsystem	
Use Case	Users/Actors
Generate Expense Statistics	Notification Subsystem
View Notification	Client



Fully Developed Use Case Descriptions and User Stories

Expense Handling Subsystem

The following analyzes the specific use cases that are part of the Expense Handling Subsystem.

Voice Input

The use case Voice Input which is a part of Add Expense has been broken down into specific details below in order to facilitate ease of implementation for developers.

User Story

As a client of the Pocket expense tracker application, I am able to add Expense with the help of voice note facility to make work easier by avoiding the traditional method of writing everything in the text fields.

Acceptance Criteria

1. For Voice note facility, Microphone is mandatory.
2. Once the data is stored with the help of microphone, user can view as text and can edit it if it is not appropriate.
3. System returns the recorded data and store in database for further use.

Fully developed use case description for Voice Input

Use case name:	*Voice Input.*
Scenario:	Add Expense with the use of microphone rather than typing.
Triggering event:	Client used to add records by typing all the details.
Brief description:	A Client adds all the details of the expense occurred by him/her. A Client adds all the detail with the use of microphone by just giving the voice note and system fetches the details and records the data.
Actors:	Client.
Related use cases:	Might be invoked by the 'Text inputs' use case.

Stakeholders:	Database.	
Preconditions:	- Client must be logged into his/her account. - Microphone must be available.	
Post conditions:	- Expense is recorded with the use of voice note. - Data is recorded in text. - Client can edit later by typing if client find some mistakes. - Data is saved in the database if client saves the expense.	
Flow of activities:	Actor	System
	1. Client indicates desire to add expense with the help of microphone.	1.1 System gives the privilege to use the microphone. 1.2 System prompts for client to record the data.
	2. Client record the expense by voice notes.	2.1 System fetches the recorded data. 2.2 System returns the recorded message in text fields.
	3. Client decides to save the information or edit late on if he/she wants some modification.	3.1 System prompts for action to take (save or edit). 3.2 System saves the recorded data in database. 3.3 System returns the recorded expense so that client can give one look.
Exception conditions:	2.1 Microphone doesn't work then voice note is not possible. 2.2 If client fails to record some data then text fields are null that is no data is recorded and then user has to type the texts and records the data.	

Barcode Scanner

The use case Barcode Scanner Input which is a part of Add Expense has been broken down into specific details below in order to facilitate ease of implementation for developers.

User Story

As a client of the Pocket expense tracker application, I am able to add Expense with the help of Barcode Scanner facility to make work easier by avoiding the traditional method of writing everything in the text fields.

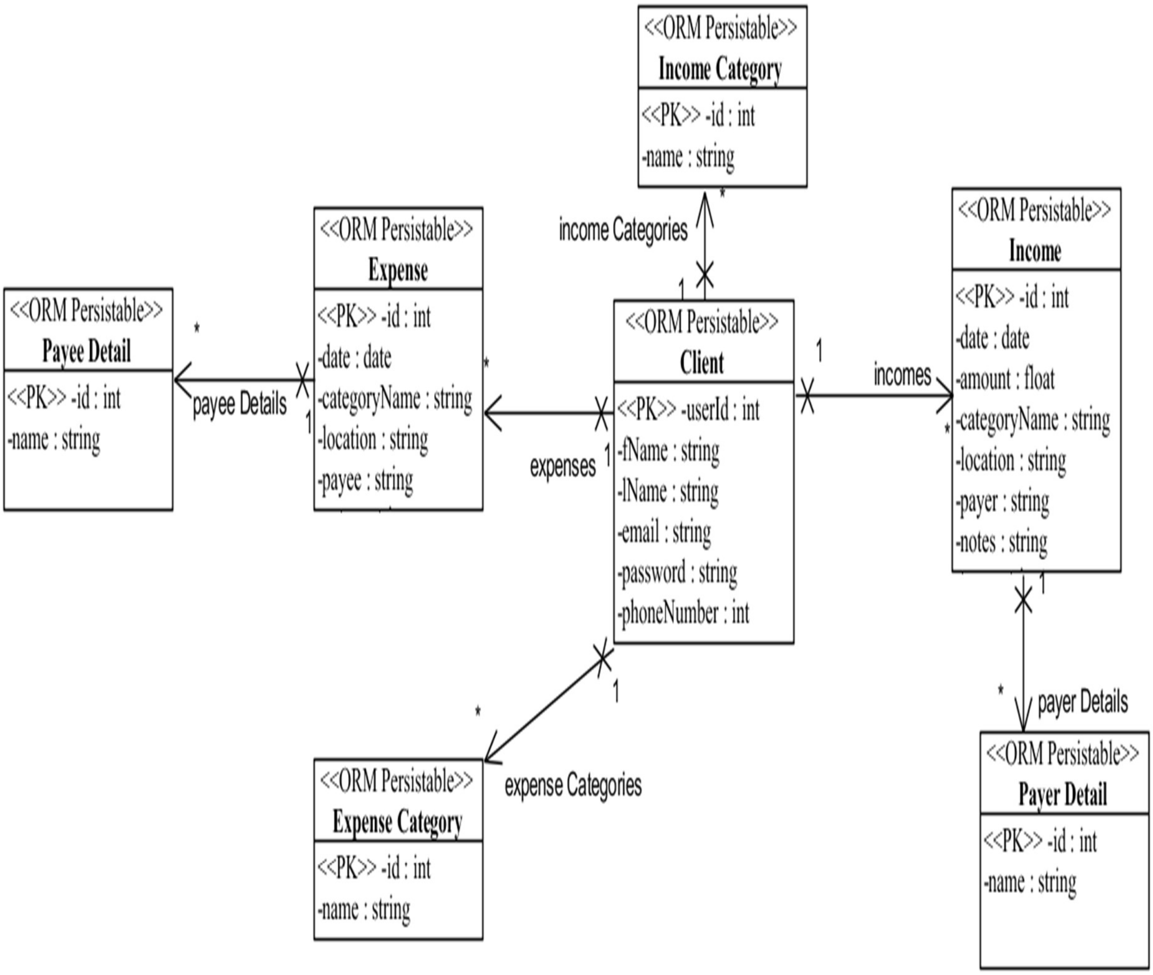
Acceptance Criteria

1. For Barcode Scanner facility, Camera is mandatory.
2. Once the data is stored with the help of Camera, user can view as text and can edit it if it is not appropriate.
3. System returns the recorded data and store in database for further use.

Fully developed use case description for Barcode Scanner

Use case name:	*Barcode Scanner.*
Scenario:	Add Expense with the use of Camera rather than typing.
Triggering event:	Client used to add records by typing all the details.
Brief description:	A Client adds all the details of the expense occurred by him/her. A Client adds all the detail with the use of Camera by just taking the picture of barcode and system fetches the details and records the data.
Actors:	Client.
Related use cases:	Might be invoked by the 'Text inputs' use case.
Stakeholders:	Database.
Preconditions:	• Client must be logged into his/her account. • Camera must be available.
Post conditions:	• Expense is recorded with the use of *Barcode Scanner*. • Data is recorded in text. • Client can edit later by typing if client find some mistakes. • Data is saved in the database if client saves the expense.

Domain Class Diagram



List of Technological Tools Used

Visual Paradigm community edition