



Cairo University
Faculty of Computers and Artificial Intelligence

CS251

Intro. to Software Engineering

Project Name

Software Requirements Specifications

Version 1.0

Email	ID	Name
Kerolus.Akram@gmail.com	20231126	كيرلس اكرم ماضي عبد المسيح
marco782005@gmail.com	20231129	ماركو رأفت زكريا سوارس
mina005smart@gmail.com	20230438	مينا ماهر توفيق صالح جرجس

Month & YearContents

Team.....	3
Document Purpose and Audience.....	3
Introduction.....	3
Software Purpose.....	3
Software Scope.....	3
Definitions, acronyms, and abbreviations.....	4
Requirements.....	4
Functional Requirements.....	4
Non Functional Requirements.....	4
System Models.....	5
Use Case Model.....	5
Enriched User Stories.....	5
System Navigation Map.....	7
Tools.....	7
Ownership Report.....	7
Policy Regarding Plagiarism: [To be removed] اقرأ هذا الجزء ثم احذفه.....	8

Instructions [To be removed] اقرأ هذا الجزء ثم احذفه

Team

ID	Name	Email	Mobile
20230438	مينا ماهر توفيق صالح جرجس	mina005smart@gmail.com	01206201907
20231129	ماركو رافت زكريا سوارس	marco782005@gmail.com	01274813802
20231126	كيرلس اكرم ماضي عبد المسيح	Kerolus.Akram@gmail.com	01210990045

Document Purpose and Audience:

- This document is explaining the our vision in building a budgeting app software and how we planned to build it
- IT is about a building and designing a budgeting app software that helps in organizing and monitoring the users pays and his taxes or any financial obligations

Target audience:

1. Software Development Team
2. Stakeholders

Introduction:

Software Purpose:

The purpose of this Personal Budgeting Software is to help users effectively manage their income, expenses, and savings through AI-driven insights, automated budgeting, and real-time financial tracking. Designed specifically for the Egyptian region, the software addresses key market gaps such as poor reporting, lack of AI integration, and limited multi-currency support, supporting offline features. By offering features like predictive expense analysis, goal setting, and secure financial data management, the software empowers individuals, freelancers, and families to make informed financial decisions, improve their savings habits, and achieve their long-term financial goals effortlessly.

Software Scope:

1. Automated Budgeting:

- > Automatically categorize transactions using AI.
- > Set spending limits and track progress in real-time.

2. Income and Expense Tracking:

- > Sync with local banks and e-wallets for real-time transaction tracking.
- > Support multi-currency for users with international income or expenses.

3. Financial Goal Setting:

- > Set and track short-term and long-term goals (e.g., saving for a car, Umrah, or retirement).
- > Provide AI-driven recommendations to achieve goals faster.

4. Advanced Reporting and Insights:

- > Generate visual reports (e.g., charts, graphs) to show spending trends.
- > Offer predictive insights (e.g., forecast future spending based on past behavior).

5. 24/7 Customer Support:

- > Offer live chat, email support, and AI chatbots for instant help.
- > Ensure Arabic-language support for all communication channels.

6. Offline Functionality:

- > Allow users to track expenses and budgets offline.
- > Sync data automatically when the app reconnects to the internet.

Requirements:

phase	definition
budgeting:	The process of creating a plan to manage income, expenses, and savings over a specific period.
Fixed Expenses:	Recurring expenses that remain relatively constant, such as rent, loan payments, and subscriptions.
Variable Expenses:	Costs that fluctuate from month to month, such as dining, shopping, and utility bills.
Emergency Fund:	A reserve of savings used for unexpected expenses like medical emergencies or job loss.
E-wallet:	Electronic Wallet (EX: Vodafone Cash, Fawry, Instapay)
KYC:	Know Your Customer (Identity verification process for financial applications)
MFA:	Multi-Factor Authentication

Functional Requirements:

Requirement ID	REQ Statement
FR01	The application shall have a login/sign-up screen as soon as the user opens the application in first time
FR02	The application shall display a box to input username and password if the user chooses to log in
FR03	Users can add many wallets manually.
FR04	Optional secure fields for bank/e-wallet integration to enable transaction synchronization
FR05	<ul style="list-style-type: none">● Preconfigured expense categories (Food, Transportation, Utilities, etc.)● Add Custom Category button for user-defined spending types

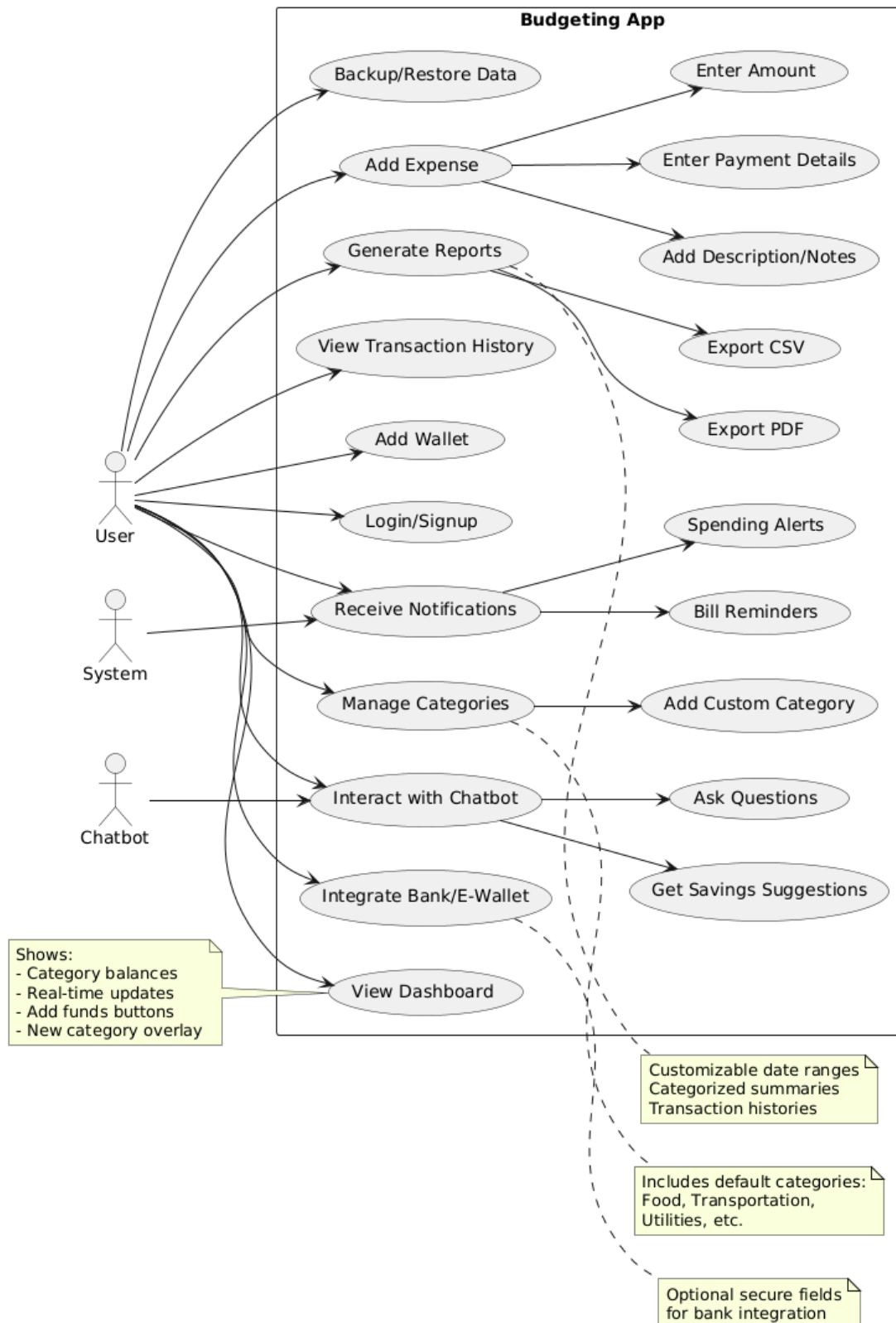
FR06	<p>Payment details:</p> <ul style="list-style-type: none"> ● Wallet Selection: Choose source wallet/account from saved options ● Category Assignment: Select from default or custom categories ● Details Entry: <ul style="list-style-type: none"> ○ Amount (required) ○ Description field (e.g., "Weekly groceries") ○ Notes/remarks (optional free-form text)
FR07	<p>The main interface displays</p> <ul style="list-style-type: none"> ● all configured spending categories with their allocated balances ● featuring per-category action buttons to securely add funds from cash reserves linked payment cards ● providing real-time balance updates and an option to create new categories through an integrated overlay menu.
FR08	<p>Add a button that opens a chat bot that is used to :</p> <ul style="list-style-type: none"> ● Suggests ways to save money and avoid unnecessary expenses. ● 24/7 AI chatbot for common questions.
FR09	Implement a proactive notification center that delivers bill payment reminders and real-time alerts for unusual spending patterns, with secure monitoring of financial activity.
FR010	Implement a report export feature with customizable date range filtering, generating categorized summaries of expenditures and detailed transaction histories in multiple formats (PDF, CSV) to analyze spending habits, track financial trends, and maintain compliant records.
FR11	Selecting any category or wallet to display all associated expenditures.
FR12	Implement local backup functionality with export/import options, enabling users to save financial data archives (.zip/.csv) to their device storage and restore them for secure data portability.

Non Functional Requirements:

Name	Details
Performance	<ul style="list-style-type: none">● Response time under 2 seconds for core actions (e.g., adding transactions, viewing balances).● Report generation within 5 seconds for standard datasets.
Security	<ul style="list-style-type: none">● End-to-end encryption for data● Multi-factor authentication (MFA) and biometric login support.● Compliance with GDPR, CCPA, and financial regulations (e.g., PCI-DSS).● Regular penetration testing and security audits.
Usability & Accessibility	<ul style="list-style-type: none">● Intuitive UI/UX with guided onboarding/tutorials.● Multi-language support (e.g., English, Arabic).● time taken to make an account not more than 5 minutes● time taken for user to choose his categories and add his balance will not exceed 10 minutes● time taken to export a report will be at most 2 minutes for worst case.
Reliability & Availability	<ul style="list-style-type: none">● Automated daily backups and disaster recovery.● the system should be available for 95% of the time considering the time for maintenance and updates.
Compatibility	<ul style="list-style-type: none">● Cross-platform support (iOS , Android ,Desktop App).● Responsive design for all screen sizes (mobile, tablet, desktop).● Integration with third-party services (banks, accounting software) via high secure APIs.
Maintainability	<ul style="list-style-type: none">● The system should have a 70% chance of maintainability for 48 hours. Which means that if a component in the system had a critical issue, there is a 65% chance that this component will be fixed in 2 days.
Scalability	<ul style="list-style-type: none">● Cloud-native architecture (AWS Cloud) with auto-scaling to seamlessly adapt to high user demand..● Support concurrent access for up to 10,000 users with horizontal scalability.
Portability	<ul style="list-style-type: none">● Data export in CSV/Excel formats.● Cross-device synchronization (real-time updates).● Manually export/import user data.
Offline Functionality	<ul style="list-style-type: none">● Core features available offline with sync on reconnect.
Notifications	<ul style="list-style-type: none">● Low-latency alerts for budget/bill reminders .

System Models:

Use Case Model:



Enriched User Stories & Normal and Exceptional Scenario:

- **User Story #1**

User Story ID	US #1
User Story Name	User login and sign up
Actors	New user , old user
Description	The user want to sign up or to log in to access his dashboard after opening the app for the first time
Per condition	app is opened
Post condition	user is registered and directed to the dashboard.
Acceptance Criteria	1. sign up: The new user when he enters a valid username, email, and password, the system creates his account and logs him in to his dashboard 2. login: the returning user, when he enters his correct user name, password , the system authenticates him and displays his dashboard.

- **Scenarios :**

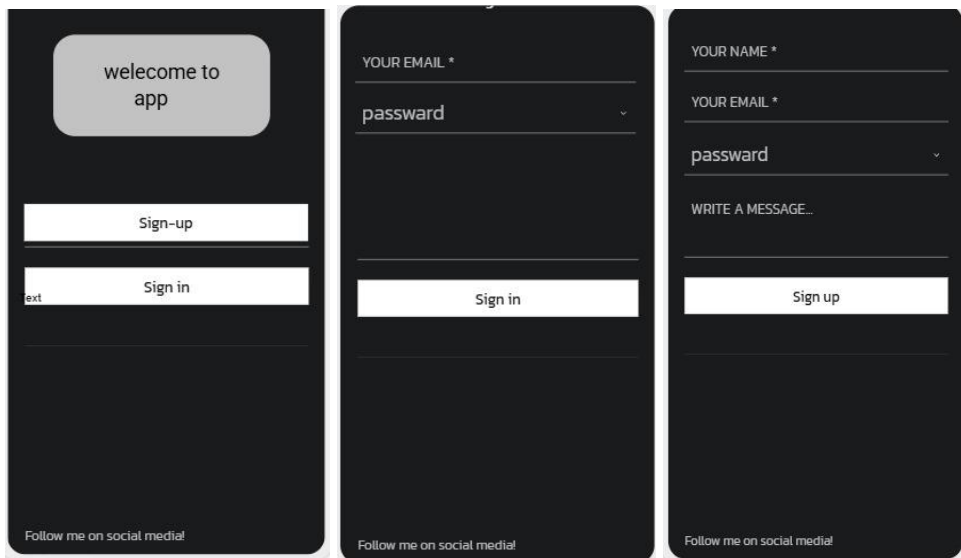
Normal Scenario :

Actor Action	System Response
1- User enters username/password. 2- User clicks "Log In."	
	2. System validates credentials. 4. System redirects to dashboard.

Exceptional Scenario :

Actor Action	System Response
1. User enters invalid password. 3. User clicks "Log In."	
	2. System detects mismatch. 4. System displays "Invalid credentials. Please try again."

- **Screen Design :**



- **Data Dictionary:**

Element Label	Type/Length	Data Validation / Business Rule
Email	Email	Valid format, no duplicates
Password	Password (8+ chars)	1 uppercase, 1 number, 1 special character
Username	Text (3–20 chars)	Unique, alphanumeric

- **User Story #2**

User Story ID	US #2
User Story Name	Wallet Management (Manual + Integration)
Actors	User
Description	the user wants to add or manages multiple wallets manually or links bank/e-wallet accounts to sync transactions.
Per condition	User is logged in
Post condition	Wallets are added and transactions are saved.
Acceptance Criteria	<p>1. insert wallet manually:</p> <p>the user on the "Wallets" screen, when he adds a wallet name and balance, it appears in his wallet list.</p> <p>2. Bank linking:</p> <p>the user links an external account, when he completes secure authentication, transactions sync automatically.</p>

- **Scenarios :**

Normal Scenario:

Actor Action	System Response
1. User clicks "Add Wallet."	
	2. System opens wallet form.
3. User enters the wallet name and balance.	
	4. System saves wallet to list.

Exceptional Scenario :

Actor Action	System Response
1. User leaves "Balance" field empty.	
	2. System detects missing field.
3. User clicks "Save."	
	4. System displays "Balance is required."

- **Screen Design:**

- **Data Dictionary:**

Element Label	Type/Length	Data Validation / Business Rule
Wallet Name	Text (1–30 chars)	Unique per user
Balance	Decimal (≥ 0)	Non-negative

- **User Story #3**

User Story ID	US #3
User Story Name	Expense Category Configuration
Actors	User
Description	I want to categorize expenses using default categories (EX: Food) or create custom categories for personalized tracking.
Per condition	User is adding/editing a transaction
Post condition	Categories are available in dropdowns.
Acceptance Criteria	1.Default Categories: I categorize a transaction, preconfigured options like Transportation are visible. 2.Custom Categories: when I click Add Custom Category then when I name and save it, the category appears in the dropdown.

- **Scenarios :**

Normal Scenario :

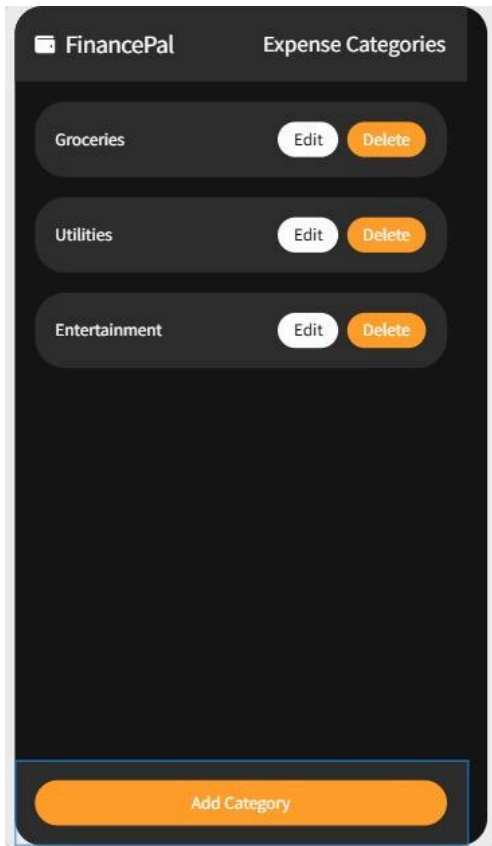
Actor Action	System Response
1. User clicks "Add Custom Category."	
	2. System opens category form.
3. User enters "Gym Membership."	
	4. System adds category to dropdown.

Exceptional Scenario :

Actor Action	System Response
1. User enters a duplicate category name.	
2. User clicks "Save."	
	3. System detects duplication.

4. System displays "Category already exists."

- **Screen Design:**



- **Data Dictionary:**

Element Label	Type/Length	Data Validation / Business Rule
Category Name	Text (1–25 chars)	Unique per user
Default Categories	Predefined list	Non-editable (e.g., Food, Transportation)

- **User Story #4**

User Story ID	US #4
User Story Name	Transaction Entry & Details
Actors	User
Description	As a user, I want to log transactions with details like wallet source, category, amount, and notes to track spending accurately.
Per condition	User clicks "Add Transaction."
Post condition	Transaction is saved and reflected in reports.

Acceptance Criteria

Given I'm adding a transaction, when I select a wallet, category, enter an amount (e.g., \$30), and add a description (e.g., "Lunch"), the transaction is saved and visible in the dashboard.

- Scenarios :

Normal Scenario :

Actor Action	System Response
1. User selects "Food" category and enters \$50. 3. User clicks "Save Transaction."	
	2. System validates inputs. 4. System updates wallet balance.

Exceptional Scenario:

Actor Action	System Response
1. User leaves "Amount" field empty. 3. User clicks "Save Transaction."	
	2. System detects missing fields. 4. System displays "Amount is required."

- Screen Design :

The screenshot shows the FinancePal app interface for adding a transaction. At the top, there is a back arrow and the app name "FinancePal". Below this, there are four input fields: "Select Wallet" with "Personal Wallet" selected, "Category" with "Food & Drinks" selected, "Amount" with "138" entered, and "Notes" with "Additional notes" as a placeholder. At the bottom of these fields is an orange "Submit Transaction" button. The bottom of the screen features a navigation bar with icons for Home, Transactions, Dashboard, Notifications, and Chat.

- Data Dictionary:

Element Label	Type/Length	Data Validation / Business Rule
Wallet	Dropdown	Must select from saved wallets

Element Label	Type/Length	Data Validation / Business Rule
Category	Dropdown	Select from default/custom categories
Amount	Decimal (>0)	Required, non-zero
Notes	Text (1–100 chars)	Optional

- **User Story #5**

User Story ID	US #5
User Story Name	AI Chatbot for Financial Support
Actors	User
Description	As a user, I want to access a 24/7 chatbot for savings tips, expense avoidance strategies, and answers to common questions.
Per condition	User clicks the chatbot icon.
Post condition	Chatbot interface opens.
Acceptance Criteria	Given I ask, “How to reduce utility bills?”, the chatbot provides actionable suggestions (e.g., “Use energy-efficient appliances”).

- **Scenarios :**

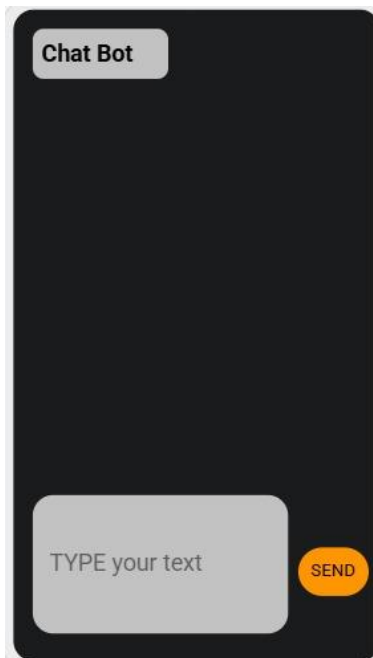
Normal Scenario:

Actor Action	System Response
1. User types "How to save on groceries?" 3. User clicks "Send."	
	2. System processes query. 4. Chatbot displays tips (e.g., "Buy in bulk").

Exceptional Scenario :

Actor Action	System Response
1. User types "How to save on groceries?" 3. User clicks "Send."	
	2. System processes query. 4. Chatbot server is down.
5. Users refresh the page.	

- **Screen Design :**



- **Data Dictionary:**

Element Label	Type/Length	Data Validation / Business Rule
User Query	Text (1–200 chars)	non empty

- **User Story #6**

User Story ID	US #6
User Story Name	Proactive Alerts & Notifications
Actors	User
Description	As a user, I want real-time alerts for bill reminders and unusual spending patterns to avoid financial risks.
Per condition	User has bills or spending thresholds configured.
Post condition	Notifications are delivered.
Acceptance Criteria	Given my credit card spending exceeds \$500 in a day, the system sends an alert: “Unusual spending detected.”

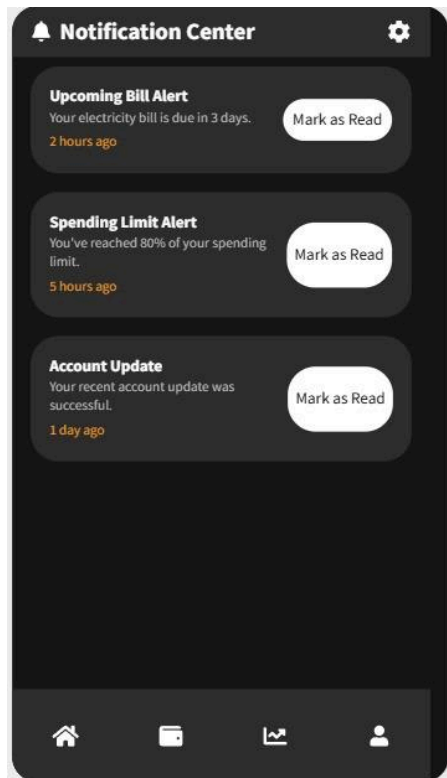
- **Scenarios :**

Normal Scenario

Actor Action	System Response
1. User’s bill due date approaches.	
	2. System detects pending payment.

3. System sends push notification: "Electricity bill due in 2 days."

- **Screen Design :**



- **Data Dictionary:**

Element Label	Type/Length	Data Validation / Business Rule
Alert Message	Text (1–150 chars)	Non-empty
Due Date	Date	Future date required for bill reminders

- **User Story #7**

User Story ID	US #1
User Story Name	Financial report Generation
Actors	the user
Description	A user wants to export a .PDF report with all his payments to print it
Per condition	user goes to report and click on export button
Post condition	the file will be downloaded
Acceptance Criteria	suppose I clicked on export button the file should be downloaded successfully on my phone and could open it

- **Scenarios :**

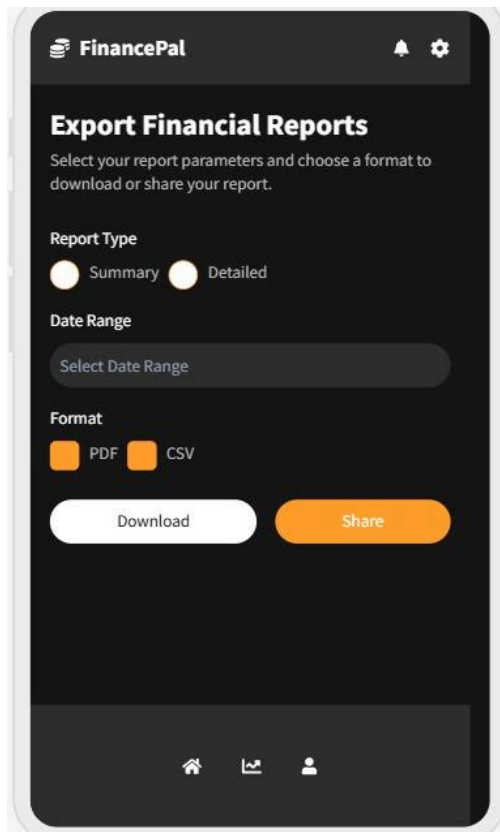
Normal Scenario

Actor Action	System Response
1. User clicks "Export summary report as PDF."	
	2. System generates report. 3. System downloads PDF to device.

Exceptional Scenario

Actor Action	System Response
1. User clicks "Export summary report as PDF."	
	2. System generates report. 3. System downloads report to device with the wrong format.

- **Screen Design :**



- **Data Dictionary:**

Element Label	Type/Length	Data Validation / Business Rule
Date Range	Start/End Date	End date \geq start date

Element Label	Type/Length	Data Validation / Business Rule
Export Format	Radion buttons (PDF/CSV)	–

- **User Story #8**

User Story ID	US #8
User Story Name	expenditure filtering by category/Wallet
Actors	A user
Description	As a user i want to filter transactions by category or wallet
Per condition	user will select category or wallet
Post condition	a filtered list with information will be displayed
Acceptance Criteria	After selecting the category or the wallet, all the related transaction should appear to me as a user

- **Scenarios :**

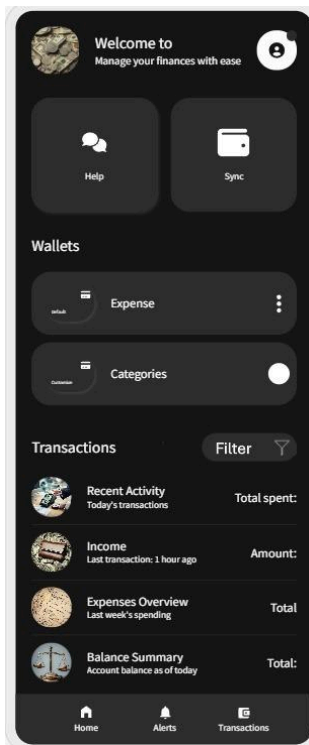
Normal Scenario

Actor Action	System Response
1. User clicks filter int transactions tab.	
	2. System fetches transactions. 3. System displays all "Utilities" expenses.

Exceptional Scenario

Actor Action	System Response
1. User selects a category with no transactions.	
	2. System detects empty list. 3. System displays: "No transactions found for this category."

- **Screen Design :**



- **Data Dictionary:**

Element Label	Type/Length	Data Validation / Business Rule
Filter	Dropdown	Show only existing categories and wallet

- **User Story #9**

User Story ID	US #9
User Story Name	Data Backup & Restoration
Actors	User
Description	As a user, I want to export financial data as a backup file (.zip/.csv) and restore it later for data portability.
Per condition	User navigates to "Backup & Restore."
Post condition	Data is exported/imported.
Acceptance Criteria	<ol style="list-style-type: none"> Export: <ol style="list-style-type: none"> When I click "Export Backup," a .zip file is saved to my device. Restore: <ol style="list-style-type: none"> When I select a backup file, all data is restored to the app.

- **Scenarios:**

Normal Scenario

Actor Action	System Response
1. User clicks "Export Backup."	
	2. System compiles data. 3. System saves .zip file to device storage.

Exceptional Scenario

Actor Action	System Response
1. User uploads a corrupted backup file.	
	2. System detects invalid format. 3. System displays: "Invalid backup file. Choose a valid .zip/.csv."

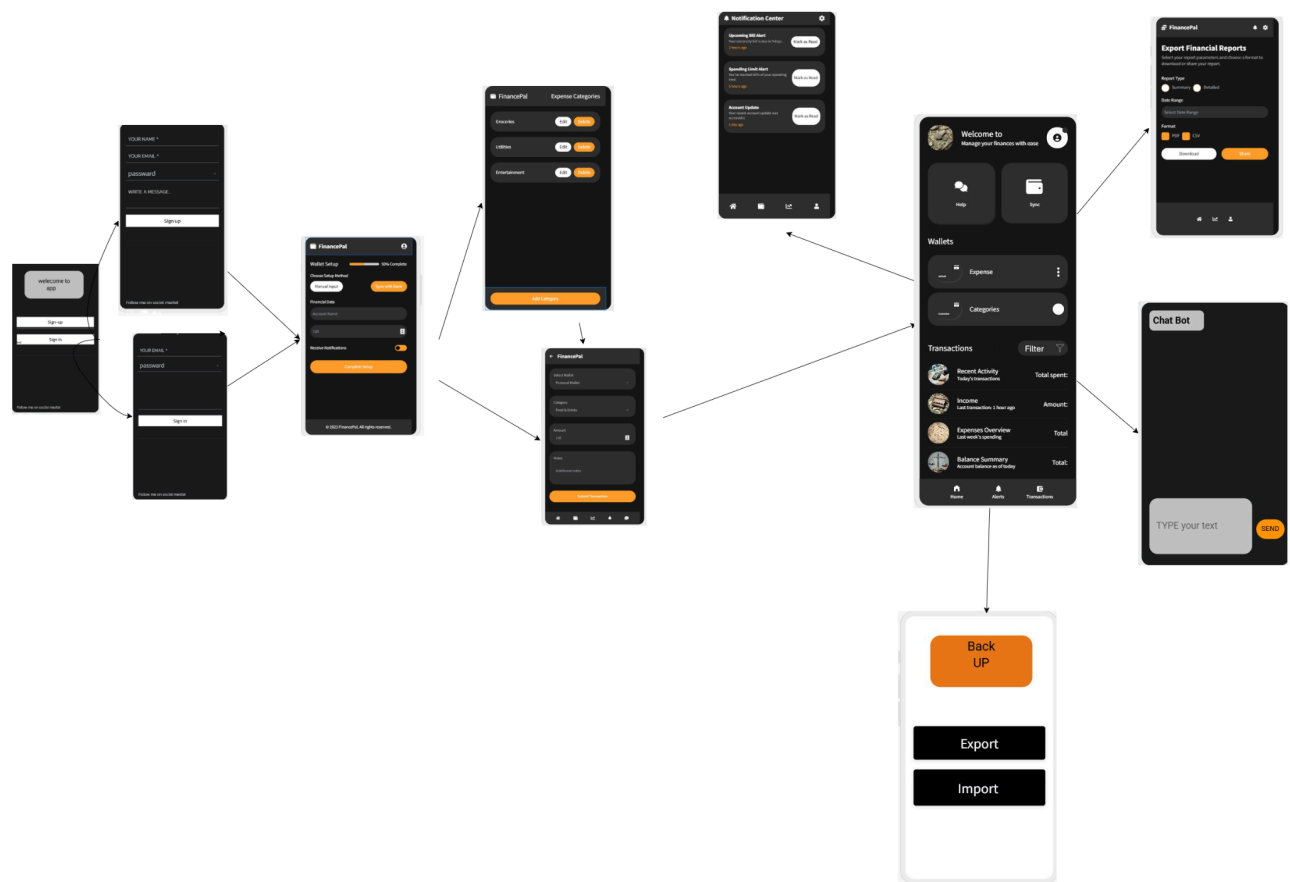
- **Screen Design:**



• **Data Dictionary:**

Element Label	Type/Length	Data Validation / Business Rule
Export	Button	
Import	button	

System Navigation Map :



Tools:

- PlantUML
- Google Docs
- DeepSeek
- Uizard

Ownership Report:

Student	Items he created
Mina Maher	Introduction User case model Requirements User stories scenarios 7,8,9.
Marco Rafaat	Introduction Requirements user case model User stories scenarios 1,2,3.
Kerolus Akram	Introduction user case model Requirements User stories scenarios 4,5,6.