Personal Finance Tracker
Specification

South East Technological University Carlow
Software Development

Name: Matthew Ufumeli
Student no: C00273575
Supervisor: Chris Staff
Date: 06/12/2024

Table of Contents

Personal Finance Tracker

1. Introduction

The **Personal Finance Tracking App** enables users to manage their income, expenses, and budgets by manually uploading CSV files from their bank accounts or by connecting directly. The app processes the CSV files and bank transactions to provide insightful visual reports on the user's financial health. This functional specification outline objectives, requirements and functionalities of this useful web app

---

2. Project Overview

The **Personal Finance Tracking App** is a web-based tool designed to help users track their financial health by uploading CSV files from their bank or credit card accounts. These files are processed and categorized automatically, allowing users to visualize their spending habits, income trends, and budget performance through charts and graphs such as bar charts, pie charts, and line graphs.

The app is strictly a financial tracking tool, focused entirely on budgeting, expense tracking, and financial analysis. It does not handle or manage actual funds—there is no capability for storing, transferring, or withdrawing money. Users simply upload CSV files or manually input their income and expenses for analysis.

2.1 Objectives

- **Simplify Personal Finance Tracking:** Provide a user-friendly interface for managing personal finances through manual input or CSV file uploads from bank or debit card accounts.
- **Automate Categorization:** Automatically categorize income and expenses from CSV files into predefined or customizable categories.
- **Offer Financial Insights:** Generate visual reports, including bar charts, pie charts, and line graphs, to show spending patterns, income trends, and budget performance.
- Support Budgeting: Allow users to set and track budgets over time and analyze their progress.
- **Ensure Data Privacy:** Ensure that the app does not store or handle sensitive information like bank credentials. All financial data is strictly for analysis purposes.

2.2 Target Audience

The Personal Finance Tracking App is designed for anyone with a personal bank account who wants to gain better control over their finances. The specific target audience includes:

- **Individuals Managing Personal Finances:** People who want a simple tool to manage and visualize their income and expenses without complex accounting software.
- **Budget-conscious Users:** Users who need to keep a close watch on their spending habits and improve their financial discipline.
- **Non-technical Users:** Individuals looking for a straightforward solution that doesn't require technical expertise to use.
- **Young Professionals and Students:** People starting their financial journeys who want a better understanding of their spending behavior and saving goals.

---

3. Core Functionality & Non-Core Functional Requirements

3.1 Core Functions

CSV File uploading: Allows users to upload CSV files from their bank accounts.

Manual Transaction Entry: Users can manually input income or expense data, if they don't want to upload CSV files.

Budgeting Tools: Users set up budgets for specific categories (e.g., food, utilities, entertainment) over a specified period (weekly, monthly, yearly). Tracks the user's progress against the set budget, with notifications or alerts when close to exceeding it.

Visual Reports: The app generates various visual reports, including charts and graphs to display financial data.

Categorization Management: The system automatically categorizes transactions or lets users create, edit, or delete categories to better match their personal finance needs. Allow users to assign or reassign transactions to different categories as needed.

Income vs. Expense Analysis: Summarize and compare total income versus total expenses for selected timeframes.

3.2 Non-Core Functions

User Account Management: Offer an option to users to create an account, save data and preferences.
Multi-Currency Support Offer multi-currency options for users with accounts in different countries or currencies, converting and displaying financial data in their preferred currency.

Data Export: Allow users to export their processed data, including visual reports, into formats like CSV or PDF for offline analysis or record-keeping.

Notifcations: Tracks the user's progress against the set budget, with notifications or alerts when close to exceeding it.

AI Implementation: The app could incorporate AI for predictive financial insights or expense categorization.

3.3 Open Banking

To integrate Open Banking in Ireland, the project would need to adhere to specific regulatory frameworks. The main regulation governing this is the PSD2 (Payment Services Directive 2), an EU directive that allows third-party providers (like fintechs) to access customer financial data from banks through APIs. This regulation ensures that users have control over their financial information while maintaining security and privacy standards.[1]

A practical approach might involve creating a dummy application simulating API interactions with bank accounts, mirroring the functionality of open banking. This would allow testing features like account aggregation and report generation without needing direct connections to actual banking APIs. It can simulate interactions using mock data and API calls, aligning with how real fintech solutions are tested during development stages.[2]

By integrating Open Banking functionality, it will allow:

- **Random Purchases:** Simulated transactions for testing purposes.
- **Budget Balance Query:** Let users track remaining balances.
- **Notifications for Thresholds:** Notify users when spending nears or exceeds set limits.

Some providers often include APIs and sandbox environments. Some examples:

- **Plaid:** Popular in North America and Europe, offering bank data access. https://plaid.com/en-eu/
- **TrueLayer:** UK and Europe, with robust Open Banking APIs. https://truelayer.com/
- **Tink:** European focus with advanced analytics. https://tink.com/
- **Dummy Services:** These use simulated environments like OpenBankProject's sandbox.https://www.openbankproject.com/

Since live Open Banking APIs may require regulatory compliance, dummy services will be used for development. Once the dummy features are functional, live Open Banking APIs will then be integrated focused on:

- **Authentication**: Implement OAuth2.0 for user consent to link their bank accounts.
- **Real Transactions**: Replace dummy purchases with real-time bank transaction data via APIs.
- **Compliance**: Ensure GDPR and PSD2 compliance for handling sensitive user data.

4. Context Diagram

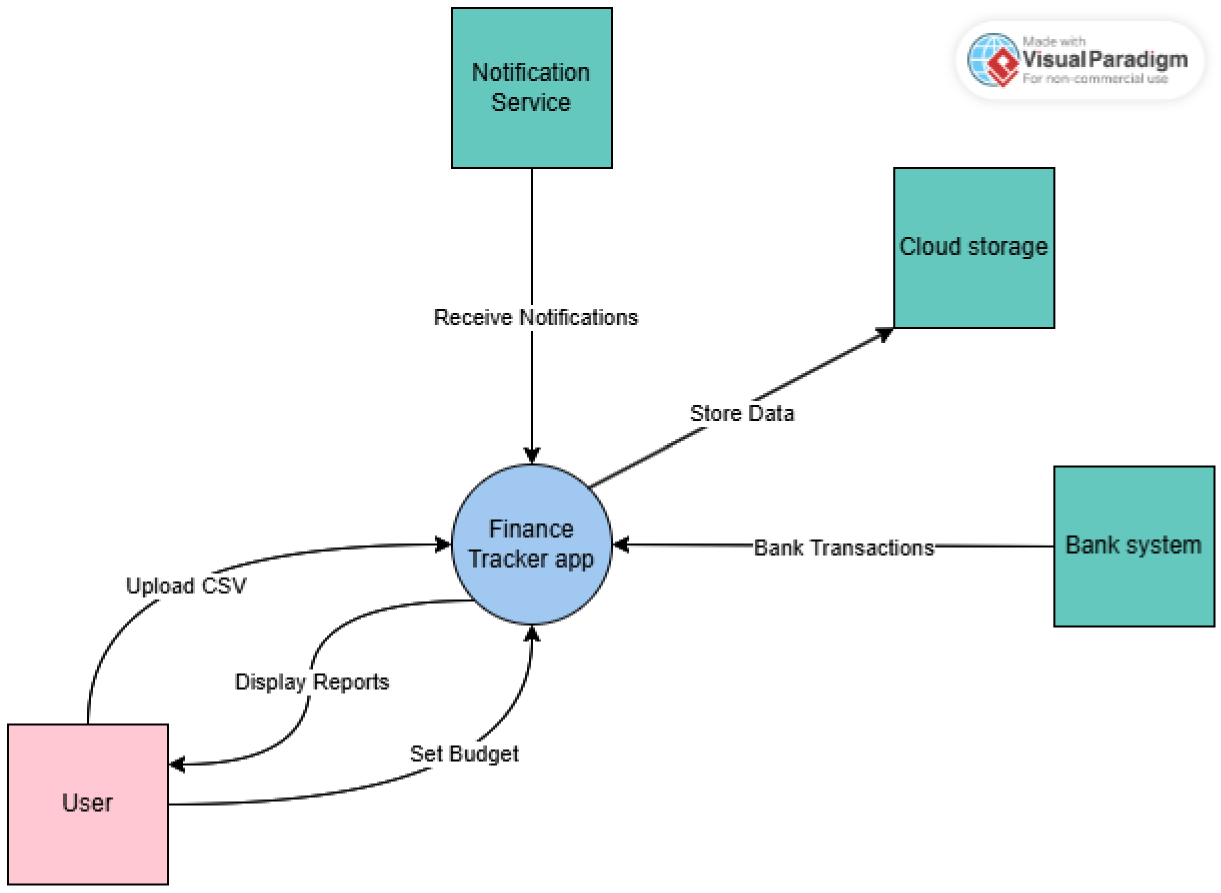

Fig 1: System Context Diagram

5. Use Cases

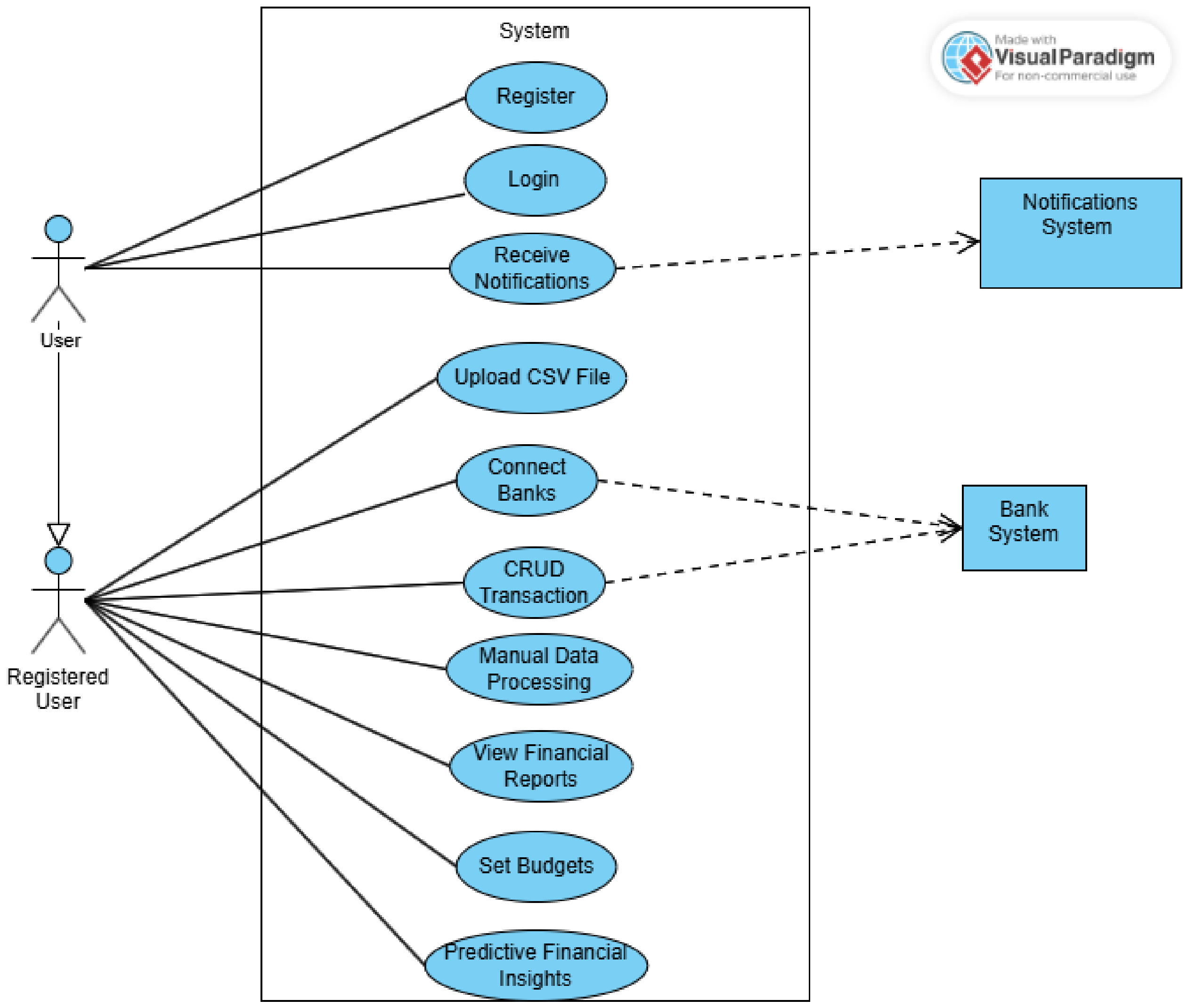

Fig 2: Use Case Diagram

5.1 Brief Use cases

Name	**Upload CSV File**
Actors	Registered User
Description	This use case begins when a user wishes to upload a CSV file to the system. The user first logs into their bank account and downloads the relevant CSV file containing their financial transactions. Afterward, the user navigates to the Finance Tracker, selects the option to upload a file, and imports the downloaded CSV for further processing. The system processes the file, categorizes the transactions, and updates the user's financial records for analysis.

Name	**View Financial Reports**
Actor(s)	Registered User
Description	This use case begins when a user wishes to view their financial reports. The user starts by uploading a CSV file containing their bank transaction data. The system processes the file, categorizes the expenses and income, and generates visual reports, such as charts and graphs. The user can then view detailed financial summaries and insights based on the uploaded data.

Name	**Receive Notifications**
Actor(s)	Registered User
Description	This use case begins when a user wishes to receive notifications. The user navigates to the settings menu, selects "Notifications," and toggles the option to turn on notifications. Once activated, the user becomes eligible to receive notifications both within the app and via email for relevant updates, reminders, or alerts about their financial activity.

Name	**CRUD Transaction**
Actor(s)	Registered User
Description	This use case begins when a user wishes to manage their expenses and incomes by manually entering or modifying transaction records. Users can create, read, update, and delete (CRUD) their financial transactions this way.

Name	**Manual Data Processing**
Actor(s)	Registered User
Description	This use case begins when a user wishes to manually input and categorize financial data without uploading a CSV file. It provides flexibility for users to update or adjust their records directly within the app.

Name	**Predictive Financial Insights**
Actor(s)	Registered User
Description	This use case begins when a user wishes to receive predictions about their future financial performance based on historical data. The system processes the user's uploaded transactions, applies data analysis techniques, and forecasts trends such as future spending, income growth, and potential budget overruns. Users can view these insights through charts and recommendations to better plan their financial decisions.

5.2 Detailed Use case

Name	**Predictive Financial Insights**
Actor(s)	Registered User
Description	This use case begins when a user wishes to receive predictions about their future financial performance based on historical data. The system processes the user's uploaded transactions, applies data analysis techniques, and forecasts trends such as future spending, income growth, and potential budget overruns. Users can view these insights through charts and recommendations to better plan their financial decisions.
Main Success Scenario:	1. The user uploads a CSV file with financial transactions or manually enters their data. 2. The system processes the data using financial algorithms and AI models. 3. The user selects the "Predictive Insights" feature. 4. The system analyzes trends and generates predictions on future spending, income, and potential savings. 5. The user views detailed charts and recommendations for financial planning. 6. Predictions are stored and updated as new data is uploaded.
Alternatives	3.a **The system detects incomplete or corrupted data in the CSV file.** ● It notifies the user to correct or re-upload the data. 5.a **The user finds that predictions are not aligned with their financial habits.** ● The user can adjust the algorithm's settings or manually fine-tune the insights. 6.a **The system fails to connect to the cloud for prediction storage.**

Name	Connecting to Bank
Actor(s)	Registered User, Bank API
Description	This use case begins when a user wishes to connect their bank account to the Personal Finance Tracker app to enable automatic transaction syncing. The system interacts with an Open Banking API, allowing the user to authenticate and authorize access to their bank data. Upon successful connection, the system retrieves and processes the user's transactions, updating the dashboard and enabling seamless financial tracking.
Main Success Scenario:	1. The user logs into the application. 2. The user selects "Connect Bank Account." 3. The user selects their bank. 4. The user logs into their bank and authorizes access. 5. The system retrieves transaction data. 6. The system updates the user dashboard. 7. The system confirms the connection.
Alternatives	**3.a The user's selected bank is not supported.** ● The system displays a message: "This bank is not supported. Please try another bank or use manual CSV upload." ● The user can return to the bank selection screen or exit the process. **4.a The user denies authorization at the bank's authentication portal.** ● The system receives a denial response from the Open Banking API. ● The system notifies the user: "Bank connection canceled. You can retry the process anytime." **5.a The Open Banking API fails to fetch transaction data.** ● The system displays a message: "Unable to retrieve transactions at this time. Please try again later." ● The user can proceed without automatic syncing or retry the operation. **6.a The system detects an expired or invalid authentication token.** ● The system prompts the user to reauthorize their bank account connection. ● The user is redirected to the bank's authentication portal to

	generate a new token.

6. Metrics

To determine whether the Personal Finance Tracking App is successful, the following metrics to track are:

6.1 Functionality Completeness:

- **CSV Upload and Processing:** The app being able to correctly handle and process CSV files, extracting relevant financial data such as income, expenses, and transaction details.
- **Categorization Accuracy**: Categorizing transactions well (e.g., groceries, entertainment) based on the CSV data or manual input.
- **Budgeting:** Users being able to set and track budgets for different categories and that the web-app provides and gives accurate feedback.
- **Report Generation:** Generating clear and insightful visual reports (bar charts, pie charts, line graphs) based on the user's financial data.
- **Manual Data Entry:** The app allows for easy manual input of transactions.
- **Simulated API Interactions:** Simulated transactions using dummy APIs must reflect realistic data, enabling robust testing of features like account aggregation, budget balance queries, and threshold notifications.
- **Randomized Test Scenarios:** Simulated transactions (e.g., random purchases) test the system's ability to categorize, summarize, and include these transactions in reports.

6.2 Usability:

- **User Interface (UI) Quality**: Ease of navigation and simplicity in performing core tasks (uploading CSV files, setting budgets, etc.).
- **Error Handling:** How the app handles errors, such as incorrectly formatted CSV files, invalid manual inputs, or missing data.
- **API Mock Testing Feedback:** The app should display meaningful feedback when testing simulated Open Banking interactions, ensuring the user understands simulated vs. real data inputs.
- **Real-Time Sync:** Mock API interactions should simulate real-time syncing to create a seamless experience for testing scenarios.

7. Data Privacy and Security

7.1 Privacy Considerations:

Minimal Data Collection:

- Only collect the data that is necessary for the app to function. E.g personal details like names, phone numbers, or addresses, are not asked for.
- After processing the CSV file and generating reports, users are offered the option to delete or export their data. Ensuring that users have complete control over their data. This includes the ability to delete their data at any time (right to erasure), and export their data in a standard format (e.g., CSV, JSON) for their personal use.

Clear Privacy Policy:

It'll be made clear about how user data will be used, stored, and protected. This can be presented as a simple privacy statement in the app, detailing:

1. What data is collected.
2. How the data is used.
3. How long the data is retained.

8. References

[1]. 2019, Jonathan Keane, https://fora.ie/open-banking-ireland-4919941-Dec2019/ [Accessed Oct 25, 2024]
[2]. Postman learning center, https://learning.postman.com/docs/designing-and-developing-your-api/mocking-data/mocking-with-examples/, [Accessed Oct 25, 2024]