Document Analysis for The Financial Domain

CashFlow Coders - Group 6

Name	Index No
Siriwardhana N.D	225543D
Fernando T.S.F	225515U
Thilakarathna T.W.M.K.C	225545K
Didulantha P.K.I	225510B

CM2900 Industry Based AI Software Project

Department of Computational Mathematics

Faculty of Information Technology University of Moratuwa

July 2025

Document Analysis for the Financial Domain

Siriwardhana N.D	225543D
Fernando T.S.F	225515U
Thilakarathna T.W.M.K.C	225545K
Didulantha P.K.I	225510B

This report was submitted to the Department of Computational Mathematics, Faculty of Information Technology, University of Moratuwa, Sri Lanka, for the partial fulfillment of the requirements of the Degree of BSc Hons (AI) degree program

July 2025

Declaration

We hereby declare that the work presented in this report is our own and has not been submitted, either in whole or in part, for any other degree, diploma, or qualification at any other university or institution of higher education. All sources of information used in this report, whether published or unpublished, have been duly acknowledged and properly cited in the text. A complete list of references is provided at the end of the report.

Name of the student	Signature
Siriwardhana N.D	
Fernando T.S.F	
Thilakarathna T.W.M.K.C	
Didulantha P.K.I	

Date: 15/7/2025

Supervised by:

Name of the Supervisor	Signature	Date
Prof.AS Karunananda		16.07.25
Ms.Uvini MataraArachchi		18/7/2025

Dedication

This project is dedicated to our parents, whose unwavering support, love, and encouragement have been a constant source of strength and inspiration. Their belief in our potential has empowered us to overcome challenges and pursue excellence with determination.

We also express our heartfelt gratitude to our beloved lecturers, whose invaluable guidance, patience, and wisdom have shaped our academic journey and sparked our passion for learning.

Lastly, we extend our sincere appreciation to all those who have supported us—your belief in our vision and your encouragement have been instrumental in bringing this work to life. This project stands as a testament to your faith in us.

Acknowledgments

We would like to express our sincere and heartfelt gratitude to our supervisor, Prof. A.S. Karunananda, for his invaluable guidance, insightful feedback, and continuous support throughout the course of this project. His expertise, encouragement, and dedication have played a vital role in shaping the direction and success of our work.

We are also deeply thankful to Ms.Uvini MataraArachchi, our mentors from Zone 24X7, for her generous mentorship and practical insights. The willingness to share her professional knowledge and real-world experience has significantly enriched our understanding and contributed immensely to the development of the project "Document Analysis for the Financial Domain".

To everyone who supported us throughout this journey, thank you for your time, patience, and belief in our efforts. This project has been a remarkable learning experience made possible by your contributions.

Abstract

This report details the development of a user friendly, AI-powered personal banking assistant mobile application. The application is designed to empower bank account holders through insightful predictions, a powerful chatbot, and an informative dashboard for analyzing transaction histories, among other features. A significant challenge faced by many bank account holders is the unexpected insufficiency of account balances for recurring monthly transactions such as subscriptions (e.g., YouTube Premium, streaming services, utility bills, and loan repayments). Traditional banking systems often fail to provide proactive insights into future financial obligations, leaving users unaware of potential shortfalls until payments fail, leading to service disruptions, penalties, or overdraft fees. Manual transaction handling is inefficient and error-prone, as humans struggle to analyze complex financial patterns and predict future cash flow accurately. Furthermore, financial advisory services, while beneficial, remain inaccessible to many due to cost, limited availability, and a lack of personalization. The proposed AI-driven personal banking assistant addresses these challenges by continuously analyzing a bank account holder`s financial transactions, identifying spending patterns, forecasting upcoming expenses, and notifying users about potential savings deficits before critical payments are due. Its primary goal is to replicate the experience of consulting with an experienced human banking assistant, providing personalized and proactive financial insights.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Background and Motivation

Managing bank accounts is a fundamental part of adult life, yet it remains one of the most neglected and anxiety-inducing responsibilities for many individuals. As digital banking becomes more widespread, users are left with large volumes of raw transaction data, but no effective mechanism to interpret it meaningfully. Although banks provide statements and basic summaries, they typically lack the intelligence to guide users toward better financial decisions or help them prepare for future obligations.

A common issue faced by users is the inability to foresee upcoming financial constraints particularly for recurring payments like subscriptions, utility bills, rent, and loan installments. In many cases, users only realize the insufficiency of their account balance after a payment attempt fails. This leads not only to penalties and service disruptions but also to a loss of financial confidence. Traditional manual tracking methods such as spreadsheets are cumbersome, error-prone, and unsuitable for real-time financial management. Furthermore, financial advisory services remain expensive, often generalized, and inaccessible to the average person.

Our motivation to develop this project was rooted in the realization that there is a critical need for an intelligent, automated, and personalized financial banking assistant one that can proactively alert users to potential upcoming financial issues, offer suggestions to optimize cash flow, and assist in financial planning. While several financial management apps exist, they fall short in terms of predictive analytics, explainability of insights, interactive user experience, and holistic task management.

Our team envisioned a solution that mimics the support of a human financial advisor delivered in a mobile-friendly format using AI. By combining time-series forecasting, transaction categorization, and natural language interaction through an AI chatbot, we aim to reduce the mental load of financial handling for users. The system would not just react to transactions after they occur, but predict and suggest preventions for financial pitfalls before they happen.

Additionally, we recognized that effective financial planning requires more than just predictions it requires actionable tasks. Many users forget upcoming financial commitments unless manually tracked. Therefore, incorporating a to-do list tailored for financial tasks, with auto reminders and chatbot integration, further enhances the practical value of the system.

Multiple banks in one application is also a gap in Sri Lankan digital banking industry has. Our approach addresses this by enabling users to manage multiple bank accounts within a single application.

This project is not just about creating another banking app; it's about transforming the way people interact with their financial lives from reactive monitoring to proactive planning. Our solution empowers users with timely insights, clear recommendations, and intelligent tools to make better financial choices without requiring technical expertise or manual effort.

1.2 Aim and Objectives

1.2.1 Aim

Developing an intelligent, mobile - banking application that empowers bank account holders to monitor, predict, and manage their financial activities. The system aims to deliver personalized insights, timely alerts, and task management support through predictive analytics, transaction categorization, and AI-driven interaction, enabling proactive financial planning and reducing the risk of payment failures and financial stress.

1.2.2 Objectives

- Integrating a user-friendly mobile application with advanced features of a banking application.
- Automated analysis of transaction data to identify recurring income and expense patterns using advanced time series forecasting and predicting future transactions.
- Generation of personalized alerts for potential insufficient balances before critical recurring transactions are due.

- Categorization of past transactions using rule-based and machine learning techniques for a clear financial overview.
- Providing users with the ability to manually manage transaction categories and organize uncategorized entries.
- Summarized financial dashboards displaying income, expense, and balance trends over user-selected time periods.
- A task-based to-do list system that allows users to add upcoming payments and track their financial commitments.
- The agentic chatbot interface allows users to get answers about personal finance summaries, predictions and to-do items. Real-time notifications for due tasks and low savings.
- Secure user authentication and data handling with privacy-focused design.
- Allowing management of multiple bank accounts from different banks.
- Allow users to handle multiple banks in one application.

1.3 Brief Introduction to the Solution

The proposed solution is a mobile-based personal banking assistant designed to help individual users manage their bank accounts more intelligently and proactively. The system addresses a common pain point among account holders, unexpected balance shortages during recurring payments such as subscriptions, utility bills, and loan repayments by providing real-time insights, forecasts, and timely notifications.

Powered by advanced AI techniques, the assistant continuously analyzes transaction histories to detect spending patterns, predict upcoming income and expenses, and alert users in advance if their balance may become insufficient. This enables users to avoid service disruptions, overdraft fees, and financial surprises.

Beyond forecasting, the assistant integrates several features to support day-to-day banking awareness. It includes a natural-language chatbot that lets users interact with their financial data conversationally retrieving transaction summaries, asking about upcoming payments, or even adding financial tasks to a built-in to-do list. The app also provides categorized views of spending, visual dashboards, account summaries, and cross-account analysis for better visibility.

To ensure privacy and security, sensitive data is encrypted and LLM interactions use anonymized or dummy inputs. The system is built on a scalable microservice architecture using FastAPI for the backend, MongoDB for storage, and React Native (via Expo) for a smooth cross-platform mobile experience.

In essence, this assistant replicates the role of a smart, always-available banking consultant empowering users with proactive financial awareness, intuitive interaction, and personalized insights right from their smartphones.

Users:

- Individual bank account holders
- Credit card users

Inputs:

- Bank transaction data
- To-do list tasks
- Chatbot commands
- Bank account integration and user credentials

Processes:

- Transaction categorization using rule-based and NLP techniques
- Forecasting of future income, expenses, and balances
- Chatbot-based financial assistance
- Risk detection (e.g., low savings, upcoming tasks)
- Notification generation and delivery
- Dashboard rendering and interactive data filtering

Outputs:

- Real-time dashboard with financial summaries and predictions
- Categorized transaction views and past transaction views
- Interactive to-do list with reminders
- Chatbot responses with personalized insights
- Notifications for financial anomalies and upcoming tasks
- User-friendly mobile interface for financial planning and control

1.4 Report structure

This report is organized into clearly defined chapters, each addressing a specific aspect of the project to ensure clarity and coherence throughout the documentation.

- **Chapter 2: Literature Review**
  This chapter presents a critical review of existing systems and solutions similar to our own. It highlights the gaps and limitations in current approaches and establishes motivation for our system.

- **Chapter 3: Technologies Used**
  Details the key technologies, frameworks, and tools adopted in the development of our system, along with the rationale behind their selection.

- **Chapter 4: Approach**
  Describes the overall strategy and methodology we employed in designing and developing the system, including system architecture, data flow, and processing mechanisms.

- **Chapter 5: Analysis and Design**
  Provides an in-depth look at the analysis of requirements and the design phase, including use case diagrams, process models, and architectural designs.

- **Chapter 6: Implementation**

  Discusses the implementation details of each module in the system, outlining how the proposed solution was realized in practice.

- **Chapter 7: Evaluation**

  Explains the methods used to evaluate the effectiveness, accuracy, and usability of the system, supported by test cases and results.

- **Chapter 8: Conclusion and further work**

  Summarizes the key outcomes of the project, reflects on the challenges faced, and highlights future improvement areas.

- **References**

  Lists all the academic, technical, and online sources referenced throughout the report, ensuring transparency and academic integrity.

Chapter 2: Literature Review

2.1 Introduction

Personal banking assistant apps have advanced rapidly, integrating AI, automation, and predictive analytics to help users monitor and manage their finances. Unlike traditional apps that only show balances and transactions, modern tools offer spending insights, risk alerts, and transaction forecasts. Popular apps like Mint, My Finances, and Cleo feature budgeting tools, real-time notifications, and AI suggestions. However, many focus mainly on budgeting and lack robust forecasting or automation. This section reviews existing solutions, highlighting their strengths and gaps compared to the proposed AI assistant.

2.2 Related Systems

2.2.1 Mint

Mint is a widely used personal finance tool that helps users track their financial activity by connecting bank accounts and automatically categorizing transactions. It provides basic insights into spending habits, low balance alerts, and reminders for upcoming bills. Additionally, Mint offers credit score monitoring and visual dashboards to improve financial visibility.

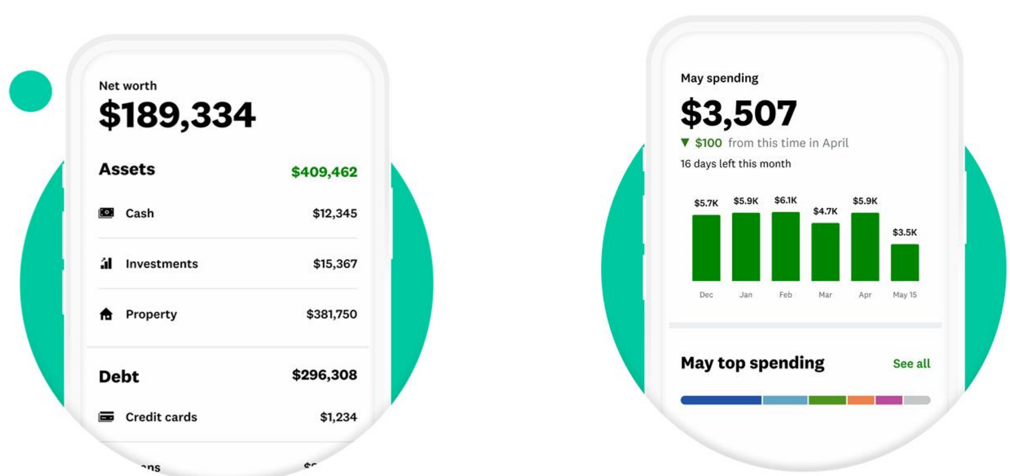


Figure 2.2.1.1 Mint App

However, Mint's core capabilities are centered around historical tracking rather than proactive banking assistance. It lacks advanced predictive features such as forecasting upcoming recurring payments or flagging future balance shortages. Moreover, Mint does not support conversational interactions or provide personalized transaction suggestions, limiting its usefulness as a true banking assistant. It also includes ads and third-party offers, which may affect the user experience.

Feature	**Our Solution**	**Mint**
Forecast upcoming transactions and balances	Yes – Uses time series forecasting	No – Focuses on historical transactions only
Provide explanations for predictions	Yes – Detailed insight into forecasting logic	No – No prediction functionality
AI chatbot for transaction queries and to-do interaction	Yes – LangChain-powered chatbot with natural language support	No – No chatbot or conversational interface
Financial task management (To-do list)	Yes – Custom financial task tracking and reminders	No – No built-in task manager
Notifications for low savings and upcoming commitments	Yes – Real-time proactive alerts	Yes – Only basic bill and balance alerts
Manual entry and tracking of upcoming payments	Yes – Add tasks manually or via chatbot	No – Not supported
Categorization of transactions (auto + manual)	Yes – Rule-based + ML-enhanced + user override	Yes – Auto categorization only
Multi-bank account support	Yes – View and filter by multiple linked accounts	Yes – Supported

Table 2.2.1.1 Our Solution Vs Mint App

2.2.2 My Finances

The My Finances app is an easy-to-use tool for managing personal finances. It allows users to link their accounts, track expenses, set budgets, and monitor income. These features help users monitor their financial activities effectively.

Users can view interactive graphs and charts that show their spending habits, income distribution, and savings progress. The app also provides a transaction history for detailed financial analysis. It includes bill reminders and alerts for due dates to help users avoid missed payments. Additionally, users can set savings goals and get insights for better debt management. Users can also generate detailed financial reports and create custom categories for a personalized experience.

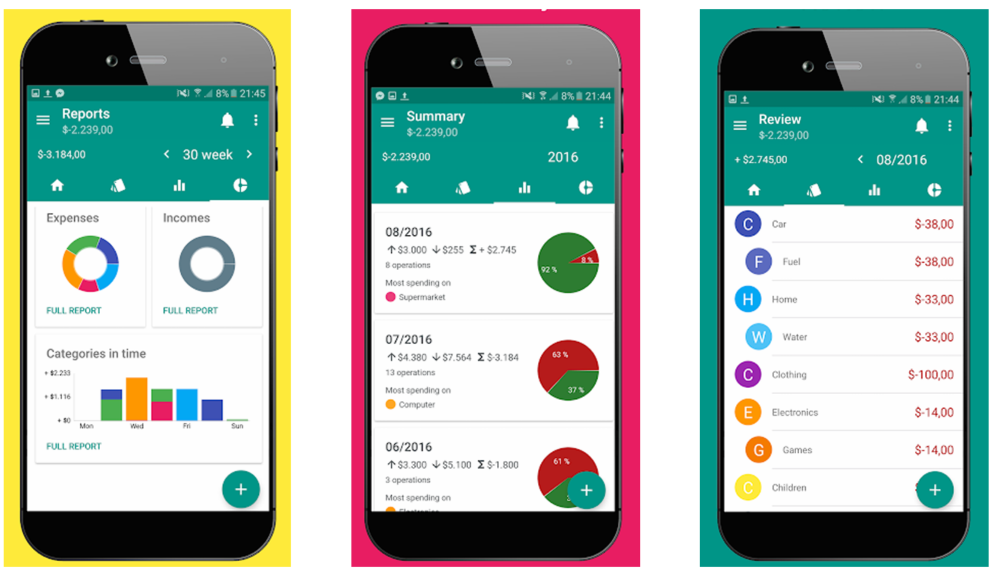


Figure 1.2.2.1 My Finances App

Even though My Finances app provides above mentioned features, it does not provide AI-driven predictions for future expenses and income. It does not have an AI-powered chatbot for financial queries and transaction insights. Also, there’s no to-do list for financial tasks to add, delete and edit the upcoming transactions. My Finances relies on only traditional charts and graphs, without providing AI-powered personalized recommendations or suggestions.

Overall, this is also a powerful tool which can be improved in many other ways.

Feature	Our Solution	My Finances
Predict future income, expenses, and balances	Yes – Advanced forecasting models	No – No predictive analytics
Chatbot for financial interaction	Yes – Conversational and context-aware	No – Lacks chatbot support
Task-based to-do list with reminders	Yes – Fully integrated with notifications	No – Not available
Manually manage upcoming transactions	Yes – Supported	No – Not supported
Personalized insights and suggestions	Yes – Based on prediction and pattern analysis	No – Only provides static graphs and reports
Transaction categorization (auto + manual)	Yes – With user control and uncategorized grouping	Yes – Basic categorization only
Dashboards with prediction overlay	Yes – Visual dashboards with historical and forecast data	Yes – Basic visual analytics
Multiple bank accounts management	Yes – Can add multiple bank accounts	Yes - Can add multiple bank accounts

Table 1.2.2.1 Our solution Vs My Finance App

2.2.3 Cleo

The Cleo app is a virtual finance assistant that helps you manage your money. It uses conversational AI to provide smart budgeting tips and personalized spending insights. Cleo categorizes your transactions and sends you alerts about your financial activity. It helps you to save your money and keeps track of your credit score, so you stay informed about your financial health.

You can also get cash advances and link accounts from different banks, giving you a clear view of your finances. The app has an easy-to-use chat interface where you can track your spending, set budgets, and get financial advice in a friendly way. By combining helpful tips with a simple chat experience, Cleo makes it easier for you to handle your money and make smart financial choices.

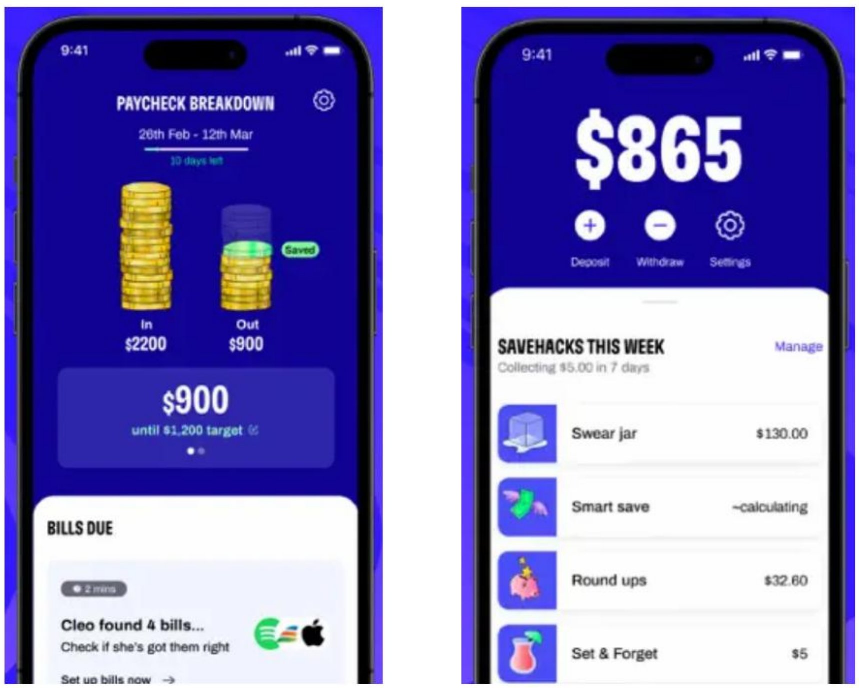


Figure 2.2.2.1 Cleo

The Cleo app does not offer features for predictive analysis or financial suggestions that could assist users in managing and planning their future finances effectively. Furthermore, it does not provide the option for users to manually input upcoming transactions, which may enhance the ease of day-to-day financial management.

Feature	Our Solution	Cleo
Conversational AI	Yes – LLM-based chatbot with full data access	Yes – Chatbot
Prediction of upcoming transactions	Yes – Uses historical patterns + ML models	No – Not available
Task management for upcoming payments	Yes – Integrated to-do list with alerts	No – Cannot add future tasks
Manual future transaction entry	Yes – Supports adding via form or chatbot	No – Not supported
Forecast explanation and transparency	Yes – Each prediction can be explained	No – No prediction engine
Financial alerts and risk detection	Yes – Real-time alerts for risks and due items	Limited – Alerts only on basic overspending
Multiple bank accounts management	Yes – Detailed filtering tools for custom views	Yes

Table 2.2.2.1 Our System Vs Cleo

2.2.4 My finance AI – Research Paper: AI-Driven Personal Finance Management: Revolutionizing Budgeting and Financial Planning

MyFinanceAI is an AI-driven personal finance system designed to address modern financial challenges. It offers features like transaction categorization, predictive budgeting, income and expense forecasting, fraud detection, bill tracking, and investment insights. The platform includes an AI chatbot for answering questions, summarizing transactions, and providing personalized financial advice.

However, the system has notable limitations: it lacks manual control over upcoming transactions, the ability to view past and predicted transactions, time-based transaction summaries, notifications, and an interactive chatbot for system-related queries.

Feature	Our Solution	MyFinanceAI (Research)
AI-powered prediction of income and expenses	Yes – Real-time, personalized forecasting	Yes – Based on predictive budgeting
Forecast explanation and transparency	Yes – Users can understand why a prediction is made	No – Not detailed in research model
Chatbot integration	Yes	Yes – With budgeting queries
Task management for recurring and ad-hoc payments	Yes – To-do list and reminders	No – Lacks manual transaction management
Notifications for low balance and upcoming tasks	Yes – Immediate alerts	No – Not specified
Multi-bank support and user filters	Yes – Built-in	Unspecified
Secure LLM interaction with dummy values	Yes – User privacy prioritized	Not covered

Table 2.2.3.1 Our System Vs My Finance AI

2.3 Research papers referred to get existing ideas and to build our solution

1. AI-Driven Personal Finance Management: Revolutionizing Budgeting and Financial Planning [1]

2. Predicting Future Incoming Bank Transactions By Detecting Recurring Transaction Sequences [2]

3. WONGA: The Future of Personal Finance Management – A Machine Learning-Driven Approach for Predictive Analysis and Efficient Expense Tracking [3]

4. Deep learning enhancing banking services: a hybrid transaction classification and cash flow prediction approach [4]

5. Customer Transaction Prediction System. Devendra Prakash Jaiswala, Srishti Kumara, Partha Mukherjeea [5]

6. Deep Learning Techniques for Bank Transaction Categorization [6]

7. Recurrent Neural Networks for Time Series Forecasting [7]

8. Predicting Future Incoming Bank Transactions By Detecting Recurring Transaction Sequences [2]

2.5 Summary

Based on the analysis of existing systems, while several applications offer partial functionality such as transaction tracking, budget visualization, or basic chatbot interfaces, none provide a fully integrated, intelligent banking assistant experience. Most tools lack predictive capabilities, do not support proactive financial task management, and fail to offer explainable insights or real-time alerts for potential financial risks.

Furthermore, current solutions rarely allow users to manually plan and manage upcoming financial obligations within the same platform, nor do they combine conversational AI, forecasting, task tracking, and secure multi-account management into one cohesive system. These limitations highlight the uniqueness of our proposed solution, which brings together AI-powered forecasting, chatbot-driven interaction, and actionable financial planning to empower users with proactive and personalized banking support.

Chapter 3: Technologies Adapted

3.1 Introduction

The development of an intelligent personal banking assistant depends on the deliberate selection of suitable technologies. This chapter explains the frameworks used to build a user-friendly, AI-driven mobile application aimed at improving financial literacy and management. The chosen technologies were selected to resolve key challenges such as recurring transaction shortfalls, inefficient manual tracking, and the limited accessibility of traditional financial advice.

The implemented technologies allow the system to analyze transactions, detect spending habits, predict future expenses, and alert users of possible balance issues. The architecture includes React Native with Expo for the mobile interface, FastAPI for backend operations, and MongoDB as a scalable database. AI models enhance the assistant's intelligence through predictive analytics, automated transaction grouping, and conversational support, creating an experience similar to a human financial advisor.

To categorize transactions, the system generates embeddings using transaction descriptions via the GTR-T5-large model. These embeddings are then clustered by HDBSCAN, grouping similar transactions (e.g., tuition or ATM withdrawals). This categorization precedes financial forecasting, which is performed using models like TFT and N-BEAST. For conversational support, LangChain-based LLM agents are integrated with Gemini API and a local Llama 3.2 instance. The next sections evaluate the rationale for each technology in depth.

3.2 Front-End Technologies

3.2.1 Introduction

The development of an effective personal banking assistant requires careful selection of front-end technologies that can support banking history analysis and transaction prediction capabilities. This evaluation focuses on identifying the optimal mobile application development framework that balances cross-platform compatibility, development efficiency, and user experience quality while ensuring seamless integration with machine learning backends for real-time financial analysis.

3.2.2 Technology Evaluation Framework

The evaluation process assessed three primary technologies against key criteria essential for financial applications: cross-platform compatibility for broad accessibility, development efficiency for rapid deployment, superior user experience for intuitive data visualization, robust integration capabilities with backends, and performance optimization for real-time transaction processing.

3.2.3 React Native with Expo

React Native with Expo emerged as the leading candidate due to its mature cross-platform architecture utilizing JavaScript and React to create native-like applications from a single codebase. The framework's component-based structure facilitates reusable UI components, crucial for consistent financial data presentation across multiple screens. Expo's integration provides comprehensive development tools, including hot reloading, over-the-air updates, and simplified native API access, features particularly valuable for financial applications requiring frequent updates and seamless user experiences.

3.2.4 Flutter Framework

Flutter was initially considered an alternative cross-platform solution for mobile application development. However, preliminary assessment revealed concerns regarding platform compatibility that influenced the evaluation process.

3.2.5 Kotlin for Mobile Development

Kotlin for Mobile Development was evaluated as a native Android option, offering strong type safety and Java interoperability. However, its platform-specific nature required additional iOS development effort, making it inefficient compared to React Native's unified cross-platform approach.

3.2.6 Technology Selection and Justification

React Native with Expo was chosen for its efficient cross-platform development, enabling a single codebase for iOS and Android. This reduces development time and maintenance while expanding market reach. Expo's features, like hot reloading and over-the-air updates, support rapid iterations essential for adapting to evolving financial needs and user feedback.

The stack supports responsive, user-friendly interfaces that present complex financial data clearly. This is vital for delivering understandable insights to users. React Native's rich library ecosystem and strong API support also make it easy to integrate backend machine learning models, ensuring smooth, real-time financial predictions and transaction analyses for enhanced personal banking experiences.

3.2.7 Technology Alignment with Project Objectives

The selected React Native with Expo technology stack directly supports the personal banking assistant's core objectives through responsive data visualization for complex financial patterns, real-time processing capabilities for immediate prediction feedback, cross-platform accessibility ensuring broad user adoption, scalable architecture supporting future enhancements, and efficient maintenance through unified codebase management. This technology selection ensures the personal banking assistant can effectively deliver predictive financial insights while maintaining optimal user experience across diverse mobile platforms.

3.3 Back-End Technologies

3.3.1 Introduction

The backend infrastructure of the personal banking assistant serves as the computational core responsible for data storage, transaction analysis, AI model integration, and API provision. Given the system's requirement to process banking history and generate predictive insights, the backend technology selection is critical for achieving optimal performance and seamless AI model integration.

3.3.2 FastAPI Framework

FastAPI is a modern, high-performance Python framework ideal for building scalable APIs. Built on Starlette and Pydantic, it supports asynchronous programming, allowing efficient handling of thousands of requests per second—crucial for real-time financial systems. It also offers automatic documentation, strong data validation, and smooth integration with frontend and machine learning services.

3.3.3 Technology Selection and Justification

FastAPI was selected as the backend framework due to its outstanding compatibility with AI integration needs. It excels in performance, supports seamless interaction with libraries like scikit-learn, TensorFlow and PyTorch, and allows native Python-based model deployment. This simplifies building features like transaction categorization, spending analysis, and predictive modeling without extra APIs or data transformations.

The framework's asynchronous architecture ensures fast, efficient handling of concurrent requests, crucial for real-time mobile applications requiring heavy AI computations. FastAPI supports high-load processing and scalable infrastructure, enabling the banking assistant to maintain speed and reliability. It also simplifies ETL operations, helping analyze banking history and extract features for accurate financial predictions.

By offering strong support for data validation, API documentation, and integration with modern ML tools, FastAPI ensures robust, maintainable, and scalable backend development. These qualities make it highly suitable for financial systems that demand both computational efficiency and secure, real-time processing of sensitive banking transactions.

3.3.4 Alternative Technology Consideration

Node.js was initially considered as an alternative backend technology, offering a JavaScript runtime built on Chrome's V8 JavaScript engine with an event-driven, non-blocking I/O model suitable for real-time applications. However, Node.js was rejected due to AI model integration complexity, requiring additional layers to interface with Python-based machine learning frameworks, which introduces potential performance bottlenecks and maintenance overhead. The predominance of machine learning and data science libraries in Python creates a significant advantage for FastAPI in AI-centric applications, providing direct access to libraries such as pandas, NumPy, and scikit-learn.

3.3.5 Conclusion

FastAPI's selection directly supports the personal banking assistant's AI-powered capabilities through high-performance processing of concurrent AI model inference requests while maintaining system responsiveness for real-time financial insights. The API-first design with automatic documentation and standardized interfaces enables seamless integration with frontend applications and potential third-party financial services, making it the optimal choice for the system's backend infrastructure.

3.4 Database Technologies

3.4.1 Introduction

The personal banking assistant requires robust database technology to handle diverse financial records, transaction histories, and predictive analytics data. The database selection is crucial for ensuring optimal performance, scalability, and data integrity for AI-powered transaction prediction capabilities.

3.4.2 Technology Evaluation

MongoDB was evaluated for its flexible, document-based data model, ideal for handling varied financial transactions without strict schemas. It supports horizontal scaling via sharding, advanced queries through indexing, and high availability with replica sets. PostgreSQL and MySQL offered structured alternatives with ACID compliance and mature SQL capabilities.

3.4.3 Selected Technology and Justification

MongoDB was selected as the optimal database technology based on its document-oriented flexibility for storing hierarchical financial data without rigid schema constraints, high performance for concurrent transaction processing with fast read/write operations, horizontal scalability to accommodate future growth in transaction data, and seamless integration with machine learning workflows for storing training data and prediction results.

3.4.4 Rejection of Relational Database Systems

Relational databases were rejected due to schema rigidity limitations that cannot accommodate diverse and evolving financial data structures, vertical scaling constraints requiring expensive infrastructure upgrades, and the hierarchical nature of financial data being more naturally represented in MongoDB's document model than normalized table structures.

3.5 Technologies for Embedding Generation

3.5.1 Introduction

The personal banking assistant incorporates advanced AI technologies for intelligent financial analysis through automated transaction categorization. A critical component involves creating numerical representations (embeddings) from textual transaction descriptions to enable semantic understanding and context preservation for accurate categorization and predictive analysis.

3.5.2 Evaluation Framework

The evaluation utilized Silhouette Scores with Cosine Distance as the primary metric for assessing embedding quality, measuring how similar an object is to its own cluster compared to other clusters. Multiple datasets (set1, set2, set3, set4) were employed to ensure robust evaluation across diverse transaction types and descriptions.

3.5.3 Technology Evaluation

GTR-T5-Large Model represents a state-of-the-art transformer-based architecture specifically designed for generating high-quality embeddings from textual data. Performance results demonstrated exceptional consistency: Set1 achieved 0.890726, Set2 achieved 0.896519, Set3 achieved 0.784076, and Set4 achieved 0.837463 Silhouette Scores.

All-MPNet-Base-V2 Model utilizes a masked and permuted pre-training approach for bidirectional context capture. Performance results showed variability: Set1 achieved

0.899160, Set2 achieved 0.797428, Set3 achieved 0.838361, and Set4 achieved 0.731000 Silhouette Scores.

Paraphrase-MPNet-Base-V2 Model was designed for paraphrase-aware embeddings, relevant for financial transactions expressed using different terminology. Performance results indicated inconsistency: Set1 achieved 0.900565, Set2 achieved 0.809865, Set3 achieved 0.853892, and Set4 achieved 0.762257 Silhouette Scores.

3.5.4 Selected Technology and Justification

GTR-T5-Large was selected as the optimal embedding generation technology based on its consistent high performance across all datasets, with Silhouette Scores ranging from 0.784 to 0.897. This consistency is crucial for financial systems requiring high accuracy across diverse transaction types. The model's high Silhouette Scores indicate excellent clustering quality, meaning transaction descriptions are accurately represented in vector space with clear semantic separation between categories. The architecture enables capture of nuanced semantic relationships in financial transaction descriptions, including variations in merchant names, transaction types, and contextual information critical for accurate categorization, while maintaining computational efficiency for real-time processing.

3.5.5 Rejection of Alternative Models

All-MPNet-Base-V2 demonstrated competitive performance in certain datasets but showed inconsistent results across different test sets, particularly significant performance drops in set2 (0.797428) and set4 (0.731000), indicating potential reliability issues for production deployment. Paraphrase-MPNet-Base-V2 showed strong performance in set1 (0.900565) but variable performance across other datasets suggested potential overfitting or limited generalization capabilities for diverse financial transaction descriptions.

3.5.6 Integration with Personal banking assistant

The selected GTR-T5-Large model supports automated transaction categorization through semantic embedding generation that preserves meaning and contextual relationships, efficient similarity calculation between new transactions and existing categories through cosine similarity measures, high-quality embeddings that facilitate effective clustering algorithms for discovering new transaction categories, computational efficiency for real-time processing, and continuous learning capabilities that adapt and improve categorization accuracy through embedding-based similarity learning from user feedback and transaction patterns.

3.6 Technologies for Similarity Calculation and Clustering

3.6.1 Introduction

Following embedding generation from transaction descriptions, the personal banking assistant requires sophisticated clustering algorithms to automatically group semantically similar transactions into meaningful categories. This clustering phase achieves automated transaction categorization that adapts to diverse spending patterns without requiring predefined category structures.

3.6.2 Evaluation Framework

The clustering algorithm selection addressed unique challenges in financial transaction categorization: unknown cluster numbers, as the system cannot predetermine category quantities for different users; varying cluster shapes, as transaction categories may not follow uniform geometric distributions; noise handling to manage outliers and ambiguous transactions; scalability for processing large transaction volumes; and robustness to maintain consistent performance across different user spending patterns.

3.6.3 Technology Evaluation

K-Means Clustering Algorithm represents a centroid-based clustering algorithm that partitions data points into predetermined clusters by minimizing within-cluster sum of squares. The algorithm features computational efficiency with O(nki) complexity, assumes spherical cluster shapes, requires predetermined number of clusters (k), demonstrates sensitivity to initial centroid placement, and performs well with well-separated, spherical clusters. However, the algorithm's fundamental requirement for specifying cluster numbers

(k) in advance rendered it unsuitable for the dynamic nature of financial transaction categorization.

HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) extends traditional DBSCAN by building cluster hierarchies and extracting flat clustering based on stability across different density thresholds. Technical characteristics include hierarchical cluster detection capability, automatic cluster number determination, robust noise handling through outlier detection, variable density cluster support, and advanced stability-based cluster extraction. Despite theoretical advantages, HDBSCAN demonstrated high sensitivity to parameter variations during evaluation, resulting in inconsistent clustering results across different transaction datasets, making it less reliable for production deployment in financial applications requiring consistent categorization performance.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) identifies clusters based on data point density in feature space, grouping closely packed points while marking isolated points as noise. The algorithm requires two parameters: epsilon (neighbourhood radius) and minimum points per cluster. Technical characteristics include density-based cluster discovery, automatic cluster number determination, effective noise and outlier handling, capability to discover clusters of arbitrary shapes, and robustness to parameter selection within reasonable ranges. DBSCAN demonstrated superior performance in clustering transaction embeddings, successfully identifying natural groupings within high-dimensional embedding space while maintaining robustness across different transaction datasets.

3.6.4 Selected Technology and Justification

DBSCAN was selected as the optimal clustering algorithm for the personal banking assistant transaction categorization system based on several critical factors. The automatic cluster discovery capability addresses fundamental challenges in transaction categorization by identifying clusters without requiring predefined category numbers, essential as different users exhibit varying spending patterns requiring adaptive category discovery. The algorithm demonstrated robust performance with consistent results across different transaction datasets, unlike HDBSCAN's high parameter sensitivity, ensuring reliable categorization performance across diverse user bases.

The density-based clustering approach proves particularly well-suited for transaction categorization as similar transactions naturally cluster together in embedding space, effectively handling varying frequencies of different transaction types without bias toward larger categories. Noise handling capabilities identify and manage outliers crucial for financial applications where unusual or one-time transactions should not distort regular spending pattern categorization. Computational efficiency with O(n log n) complexity using proper indexing ensures scalable performance for real-time transaction processing, maintaining system responsiveness as transaction volumes grow.

3.6.5 Rejection of Alternative Algorithms

K-Means limitations stem from the fundamental requirement for predetermining cluster numbers (k), making it unsuitable for dynamic financial categorization where spending patterns vary significantly between users and new categories may emerge over time, requiring adaptive clustering solutions. HDBSCAN sensitivity issues pose significant challenges for production deployment despite theoretical advantages, as high sensitivity to parameter variations could lead to unreliable categorization results and undermine user trust in the system.

3.6.6 Integration with personal banking assistant

The selected DBSCAN clustering algorithm supports automated transaction categorization through adaptive category discovery that automatically identifies natural transaction categories based on spending patterns without predefined structures, enabling personalized financial insights. Similarity-based grouping leverages high-quality embeddings generated by GTR-T5-Large to group semantically similar transactions, ensuring accurate categorization based on transaction context and meaning. Outlier management effectively handles unusual or one-time transactions that don't fit established patterns, preventing distortion of regular spending category definitions, while maintaining computational efficiency for real-time processing and scalable architecture supporting growing transaction volumes with consistent categorization performance across diverse user spending patterns.

3.7 Time Series Forecasting Models

3.7.1 Introduction

Personal financial forecasting presents unique challenges that distinguish it from traditional time series prediction problems. The irregular nature of personal transactions, varying data volumes across users, and the need for interpretable predictions require careful consideration of appropriate modeling techniques. This evaluation examines eight prominent time series forecasting models against specific criteria relevant to personal financial prediction: data requirements, multitype forecasting capability, explainability, and uncertainty handling.

3.7.2 Model Evaluation Framework

The technology selection process was guided by four critical criteria essential for personal financial forecasting: data requirements, as personal finance applications often operate with limited historical data,especially for new users, multitype forecasting capability for predicting multiple financial categories simultaneously versus requiringseparate models, explainability where users need to understand the reasoning behind financial predictions for trust and decision-making, and uncertainty handling as financial predictions must communicate confidence levels and potential risks.

3.7.3 Technology Evaluation

LSTM (Long Short-Term Memory) networks are powerful for capturing long-term dependencies and modeling complex, irregular financial behaviours [8].However, they require separate models per financial category, offer limited interpretability, and need additional mechanisms for uncertainty quantification. These limitations increase system complexity and hinder user trust, leading to their exclusion from personal banking assistant.

Prophet, developed by Facebook, is a decomposable statistical model requiring minimal data and offering strong explainability via trend and seasonality components [9] . While it handles missing data well and includes built-in uncertainty intervals, it necessitates separate models for each category and assumes seasonal patterns that may not exist in all personal financial data, reducing its suitability for comprehensive personal finance applications.

N-BEATS-M, an extension of N-BEATS, supports multiple time series and offers a trade-off between accuracy and interpretability [10]. Although it has moderate data requirements and multitype forecasting capability, its generic variant lacks sufficient explainability, and overlaps with models selected for Personal banking assistant. Thus, despite its strong performance, it was not chosen.

Informer, leveraging sparse attention mechanisms, excels at long-sequence forecasting and multitype predictions but requires large datasets (over 12,000 points), excessive for personal financial data [11]. Its limited interpretability and high computational cost further reduce its feasibility in Personal banking assistant compared to alternatives like TFT.

N-BEATS (Neural Basis Expansion Analysis) provides high forecasting accuracy and interpretable decomposition into trend and seasonal components, making it ideal for aggregate financial metrics such as daily balance and total expenses [12]. It supports moderate data volumes (<1000 points), offers clear component breakdowns, and operates efficiently without complex feature engineering. For users seeking overall financial trends, N-BEATS delivers actionable insights while maintaining model transparency.

PatchTST utilizes a patching mechanism to improve transformer efficiency and supports multitype forecasting with moderate data needs [13]. However, its lower explainability and insufficient native uncertainty handling limit its appropriateness for financial applications where trust is critical.

Temporal Fusion Transformer (TFT) stands out for its capacity to model complex multivariate time series, providing category-wise and interdependent income and expense predictions [14]. It integrates attention mechanisms and variable importance scores for high explainability, supports uncertainty quantification, and handles multiple categories simultaneously. Despite higher data requirements and computational costs, TFT's advantages in multi-dimensional financial forecasting justify its inclusion for category-level predictions.

3.7.4 Selected Technologies and Justification

N-BEATS (Neural Basis Expansion Analysis) is an advanced neural network architecture specifically designed for univariate time series forecasting. It employs hierarchical structure with basis expansion that decomposes time series into trend and seasonal components, providing both high forecasting accuracy and model interpretability. Implementation in personal banking assistant includes daily balance prediction forecasting daily account balances based on historical balance patterns and daily total expenses prediction for aggregate daily spending amounts across all categories. Key advantages include high interpretability through decomposition of forecast components into trend and seasonal patterns, strong performance with moderate data requirements (less than 1000 data points), ability to identify and explain cyclical spending patterns, pure neural network approach without requiring complex feature engineering, and excellent performance for single-variable predictions with clear decomposition. N-BEATS is ideal for aggregate financial metrics where users need to understand overall financial trends, with decomposition capabilities allowing users to see whether predicted changes in balance or expenses are driven by trend shifts or seasonal patterns.

Temporal Fusion Transformer (TFT) represents state-of-the-art approach combining transformer architecture with temporal modeling capabilities, specifically designed for complex multi-variate time series forecasting with built-in explainability mechanisms. Implementation in Personal banking assistant includes category-wise expense prediction for forecasting spending across multiple expense categories simultaneously and income and expense prediction for joint modeling of income and expense patterns with their interdependencies. Key advantages include excellent multitype forecasting capability handling multiple financial categories simultaneously, high explainability through attention mechanisms and variable importance rankings, robust uncertainty quantification through prediction intervals, ability to model complex relationships between different financial categories, and superior performance on multi-variate financial datasets with interdependencies. Despite TFT's substantial data requirements, it is essential for category-wise predictions where relationships between different spending categories and income sources are crucial.

3.7.5 Strategic Implementation Approach

The dual-model strategy addresses different aspects of the personal financial forecasting problem. N-BEATS for aggregate predictions focuses on overall financial health indicators (balance, total expenses), provides clear trend and seasonal decomposition, remains optimal for users needing high-level financial overview, and maintains lower computational requirements for simple aggregated predictions. TFT for complex multi-dimensional predictions handles interdependent category relationships, provides sophisticated analysis of spending patterns across categories, remains essential for detailed budgeting and category-specific planning, and justifies higher computational cost through superior multi-variate insights.

This strategic approach ensures that simpler predictions (daily balance, total expenses) benefit from N-BEATS' interpretability and efficiency, while complex multi-dimensional predictions (category-wise forecasting) leverage TFT's advanced capabilities for handling inter-category relationships and multi-variate dependencies. These models collectively enable the personal banking assistant to predict multiple financial dimensions including projected account balances based on historical income and expenditure patterns, recurring expense predictions for subscription services, utilities, and regular payments, seasonal spending variations related to holidays, monthly cycles, and personal spending habits, and category-specific expenditure forecasts for budgeting and planning purposes.

3.7.6 Technical Considerations

The selection of N-BEATS and TFT addresses specific challenges in personal financial forecasting including data variability where personal financial data often exhibits irregular patterns requiring robust modelling approaches, interpretability requirements where users need to understand forecast reasoning for effective financial planning, computational efficiency where models must operate efficiently within microservice architecture, and scalability where the system must handle multiple users with varying data volumes and transaction frequencies.

3.7.7 Conclusion

This critical technology review demonstrates that the selection of N-BEATS and TFT for the personal banking assistant is based on systematic evaluation against specific criteria relevant to personal financial forecasting. The strategic decision to use different models for different prediction tasks reflects a nuanced understanding of the varying requirements within personal finance applications. The hybrid approach addresses the core problem of providing accurate, interpretable financial forecasts across different prediction complexities: N-BEATS for aggregate, interpretable predictions where trend and seasonal decomposition provide clear insights, and TFT for complex multi-dimensional predictions where inter-category relationships are crucial.

3.8 Conversational AI Technologies

3.8.1 Introduction

The integration of conversational AI into personal finance management fundamentally transforms user experiences by enabling natural language interactions with complex financial data. This chapter critically evaluates conversational AI technologies for the Personal Finance Management System personal banking assistant, addressing challenges including secure handling of sensitive financial data, precise domain knowledge integration, and effective user guidance. The primary objective is to empower users to query and interact with their financial information conversationally while ensuring privacy and delivering accurate, context-aware responses.

3.8.2 Technology Evaluation Framework

The technology selection process was guided by five essential criteria: natural language processing capability to handle diverse financial queries, domain knowledge integration to provide accurate advice, robust data privacy and security, contextual awareness to maintain conversation continuity, and seamless system integration with personal banking assistant microservices.

3.8.3 Technology Evaluation

LangChain Framework serves as the core orchestration technology, offering a modular architecture capable of integrating various AI components and maintaining conversation context via built-in memory. Its agent-based design allows specialized handling of different financial queries and supports extensibility through custom tools and data sources. LangChain's production-ready infrastructure reduces development complexity and ensures scalability. Its ability to orchestrate specialized agents makes it ideal for handling tasks like transaction retrieval, forecasting, and financial advice, aligning perfectly with personal banking assistant requirements [15].

Retrieval-Augmented Generation (RAG) Architecture addresses the critical need for factual accuracy and domain-specific knowledge in financial conversations. By combining the generative strengths of large language models (LLMs) with dynamic retrieval from trusted sources, RAG mitigates hallucination risks and allows access to updated information without retraining. The architecture incorporates financial education materials, personal banking assistant documentation, compliance guidelines, and user-specific data (handled securely), ensuring responses are grounded and explainable. This capability to integrate and dynamically update knowledge bases makes RAG indispensable for delivering reliable financial guidance [16].

3.8.4 Language Models

The personal banking assistant adopts a **hybrid LLM approach**, leveraging the OpenAI API for complex reasoning tasks and a local Ollama instance (Llama 3.2 model) to safeguard sensitive data. The OpenAI API provides advanced natural language understanding, superior reasoning abilities, and continuous improvements, enabling it to handle diverse and sophisticated financial queries. Meanwhile, the local Ollama instance ensures privacy by processing sensitive data on-premises, reducing API costs, enabling offline functionality, and allowing fine-tuning for domain-specific needs. This hybrid strategy balances high performance and strict privacy requirements, making it particularly suited to financial contexts where trust and compliance are critical [17].

3.8.5 Data Privacy and Security Implementation

Data privacy and security implementation within personal banking assistant employs comprehensive measures, including dummy value substitution to prevent exposure of actual financial data in external API calls, local analysis of sensitive data via the Ollama instance, anonymization techniques for training purposes, and secure communication protocols. A dedicated privacy layer processes financial data by substituting real values with semantically equivalent placeholders, ensuring that the system can still generate accurate responses while protecting user information.

3.8.6 Alternative Technologies Evaluated and Rejected

Alternative technologies evaluated and rejected include traditional rule-based chatbots, which lack the flexibility and natural language understanding necessary for diverse financial queries, making them inadequate for personal banking assistant. Standalone LLMs without retrieval mechanisms were also rejected due to their high hallucination risk and inability to incorporate up-to-date financial information. Similarly, purely cloud-based LLM solutions were excluded because of privacy concerns and regulatory compliance requirements, underscoring the need for local processing capabilities.

3.8.7 Chatbot Functionality Implementation

The chatbot functionality within personal banking assistant is designed as a virtual financial advisor, capable of retrieving transaction histories, providing financial predictions using integrated N-BEATS and TFT models, managing tasks via conversational commands, summarizing transactions for specified periods, offering financial education, and giving context-based financial advice tailored to individual user patterns. Tight integration with personal banking assistant microservices, such as the Financial Forecasting Service, Transaction Management Service, and User Management Service, enables seamless and accurate responses.

3.8.8 Conclusion

In conclusion, the selected conversational AI architecture represents a comprehensive, scientifically grounded solution that addresses the unique challenges of natural language interaction in personal finance. The synergistic combination of LangChain orchestration, RAG architecture, and a hybrid LLM deployment ensures accurate, secure, and context-aware financial guidance. This setup significantly enhances user experience by enabling intuitive interaction with financial data while maintaining robust privacy protections and delivering reliable, actionable advice.

3.9 Data Processing and Machine Learning Libraries

3.9.1 Introduction

Beyond the core AI components, the personal banking assistant leverages fundamental data processing and machine learning libraries to ensure robust data handling, analysis, and model development.

3.9.2 Pandas

Pandas [18] is an open-source data analysis and manipulation library for Python that provides comprehensive functionality for structured data processing [19]. The library offers specialized data structures, particularly Data Frame and Series objects, which enable efficient handling and analysis of tabular data analogous to database tables or spreadsheets [20]. Pandas serves as a fundamental component in data science and machine learning workflows due to its extensive capabilities in data cleaning, transformation, aggregation, and visualization operations [21].

Within the Personal banking assistant architecture, Pandas is employed across several critical data management processes:

Data Cleaning and Preprocessing: The library facilitates comprehensive data cleansing operations, including feature selection and transformation of raw financial transaction data [22]. This preprocessing ensures data consistency and quality, which is essential for accurate predictions and effective analysis by the system's AI models.

Data Aggregation and Feature Engineering: Pandas enables sophisticated data aggregation operations, allowing for the extraction of meaningful insights from unprocessed financial data through advanced functionalities such as group by operations and algorithms for computing rolling averages and cumulative sums [23].

Integration with Machine Learning Frameworks: The seamless interface between Pandas and other Python data science libraries, including Scikit-learn, TensorFlow, and Keras, supports the complete workflow from data preprocessing to model evaluation, making it an optimal choice for efficient handling and processing of large datasets in machine learning applications [24].

Through the implementation of Pandas, the personal banking assistant ensures that financial transaction data, which constitutes the foundation of all AI-driven insights, is optimally prepared for subsequent analysis and predictive modelling operations.

3.9.3 Scikit-learn

Scikit-learn is a comprehensive machine learning library for Python that provides efficient tools for data analysis and predictive modelling [25]. The library offers an extensive range of algorithms for classification, regression, and clustering tasks, along with utilities for model selection and evaluation. Within the personal banking assistant framework, Scikit-learn plays a pivotal role in several intelligent functionalities:

Predictive Model Development: The library is utilized to construct predictive models that generate reliable estimations by processing historical and contextual information [26]. This capability is fundamental to the personal banking assistant 's ability to forecast future financial trends, including projected account balances and recurring expenses, based on established user spending patterns.

Transaction Categorization: While the personal banking assistant employs natural language processing techniques and rule-based analysis for automated transaction categorization, Scikit-learn's clustering algorithms (including K-Means, DBSCAN, and HDBSCAN) are integral to automatically grouping similar transactions, thereby establishing the foundation for financial analysis and enabling structured understanding of spending behaviours [27].

Model Training and Optimization: Scikit-learn's functionalities encompass training machine learning models, evaluating various methodologies (such as decision trees, random forests, support vector machines, and gradient boosting), and optimizing performance through cross-validation techniques. This ensures continuous refinement and enhanced accuracy of the personal banking assistant's AI models.

The library's computational efficiency, straightforward implementation, and seamless integration with other Python libraries such as NumPy and Pandas make it an optimal choice for developing and implementing machine learning models within the personal banking assistant architecture [28].

3.9.4 Hyperparameter Tuning

To further optimize the performance and accuracy of the predictive models within the personal banking assistant, particularly the sophisticated AI-driven time series models used for financial forecasting (such as Temporal Fusion Transformers (TFT), LSTM, and N-BEAST), hyperparameter tuning constitutes a critical process [29]. Hyperparameters are configuration settings external to the model that cannot be learned from the data itself, and their optimal values significantly impact model performance.

While traditional methods such as grid search and random search exist for hyperparameter optimization [35], the personal banking assistant leverages advanced libraries for more efficient tuning:

Optuna Library: Optuna is an open source hyperparameter optimization framework that automates the search for optimal hyperparameter values [30]. The framework employs state-of-the-art algorithms that intelligently explore the hyperparameter space, often converging on superior solutions significantly faster than traditional grid search or random search methodologies [31]. Its flexibility enables compatibility with various machine learning frameworks, making it well-suited for tuning complex deep learning models such as TFT, LSTM, and N-BEAST that are central to the personal banking assistant's forecasting capabilities. Through efficient identification of optimal hyperparameters, Optuna contributes to the system's objective of continuously refining its predictions and ensuring more accurate and adaptive financial insights for users [32].

Chapter 4: Approach

4.1 Introduction

This chapter presents the comprehensive approach adopted for developing the personal banking assistant, an AI-powered mobile application designed to transform traditional personal banking from reactive to proactive financial management. The approach outlined in this chapter bridges the gap between the identified problem domain and the technological solutions, providing a detailed roadmap for implementing an intelligent financial assistant that addresses the critical challenges faced by bank account holders.

The chapter details the proposed solution framework that leverages cutting-edge artificial intelligence and machine learning technologies to deliver personalized financial insights. It encompasses the high-level solution architecture that prioritizes security and privacy, the sophisticated prediction flow that transforms raw banking data into actionable forecasts, and the comprehensive system components that enable seamless user interaction. The approach emphasizes privacy-first design through hybrid local and cloud processing, scalable microservice architecture, and intelligent automation to deliver proactive financial management capabilities.

The methodology presented integrates multiple advanced technologies including natural language processing for transaction categorization, time series forecasting models for financial prediction, and conversational AI for user interaction. This multi-faceted approach ensures that the system not only meets current financial management needs but also adapts to evolving user requirements and financial patterns.

4.2 Proposed Solution

The proposed solution is a comprehensive mobile application that serves as an intelligent personal finance management platform. The application integrates with users' banking systems to automatically retrieve their complete banking history and transaction data, providing a centralized hub for all financial activities across multiple bank accounts. At its core, the solution employs a sophisticated multi-model approach for financial prediction, utilizing advanced machine learning algorithms to analyze historical transaction patterns and generate accurate forecasts of future financial behavior. This predictive capability enables users to anticipate upcoming expenses, identify potential cash flow issues, and make informed financial decisions before problems arise. The application features an interactive agentic chatbot that serves as a personalized financial assistant, allowing users to engage in natural language conversations about their finances. Users can query their transaction history, request spending analysis, seek financial advice, and receive explanations for predictions through intuitive conversational interactions. A key strength of the solution is its multi-account integration capability, enabling users to connect and manage all their bank accounts within a single interface. This comprehensive view eliminates the need to navigate multiple banking applications and provides a unified perspective on personal financial health. The solution delivers full financial analytics through an intuitive dashboard that presents both historical summaries and future predictions with detailed explanations. Users can explore their spending patterns, income trends, category-wise breakdowns, and receive clear explanations for why certain predictions are made, ensuring transparency and building trust in AI-driven insights.

4.2.1 High-Level Solution Architecture

The personal banking assistant implements a comprehensive architecture that prioritizes security, privacy, and intelligent processing, as illustrated in the high-level solution diagram. The architecture follows a layered approach with clear separation of concerns and robust data protection mechanisms.

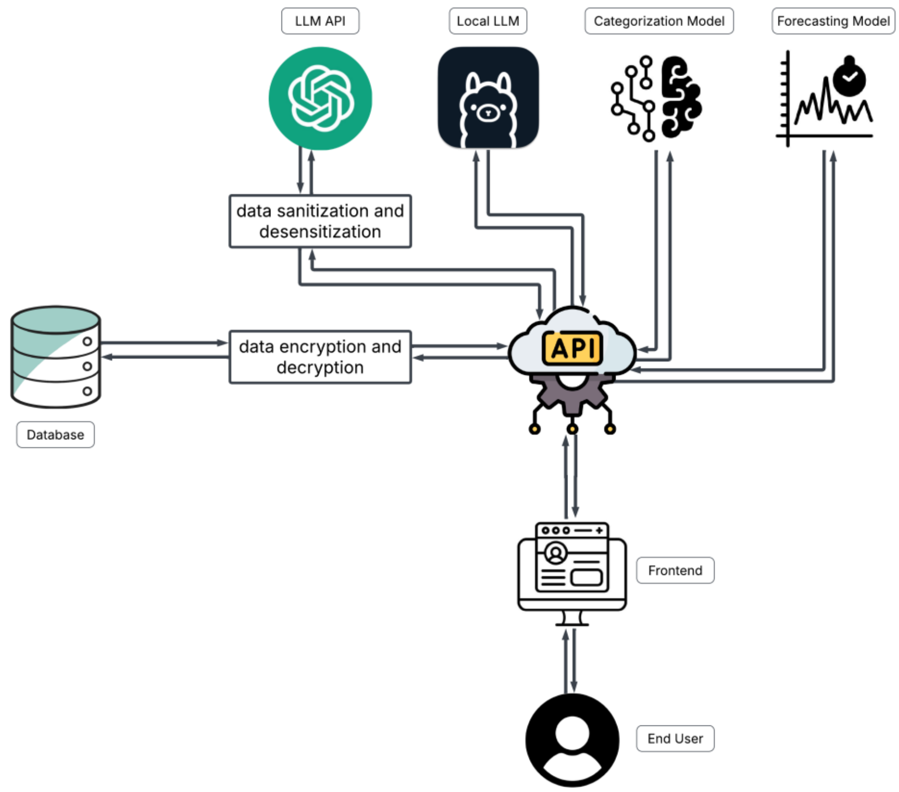


Figure 3.2.1.1 High-level Solution Architecture

At the user interface level, the Frontend serves as the primary interaction point for End Users, providing an intuitive mobile application interface for all financial management activities. The system employs a dual-layer security approach with data encryption and decryption mechanisms ensuring that all sensitive financial information remains protected throughout the entire data lifecycle.

The core processing architecture integrates multiple specialized components working in harmony. The Database serves as the central repository for all user financial data, maintaining encrypted storage with secure access controls. The Categorization Model processes raw transaction data to automatically classify financial activities into meaningful categories, while the Forecasting Model generates accurate financial predictions through sophisticated machine learning algorithms.

A critical privacy-preserving feature is the hybrid language model implementation, combining a Local LLM for sensitive data processing with an external LLM API for enhanced capabilities. This architecture ensures that sensitive financial information is processed locally through data sanitization and desensitization techniques before any external API communication occurs. The Local LLM handles privacy-critical operations, while the sanitized data can be safely sent to the LLM API for advanced natural language processing and response generation.

4.2.2 Prediction Flow

The system employs a sophisticated multi-stage prediction process that transforms raw banking data into actionable financial insights. The complete prediction workflow is illustrated in the system flow diagram, which demonstrates the sequential processing stages from raw data input to final predictions.

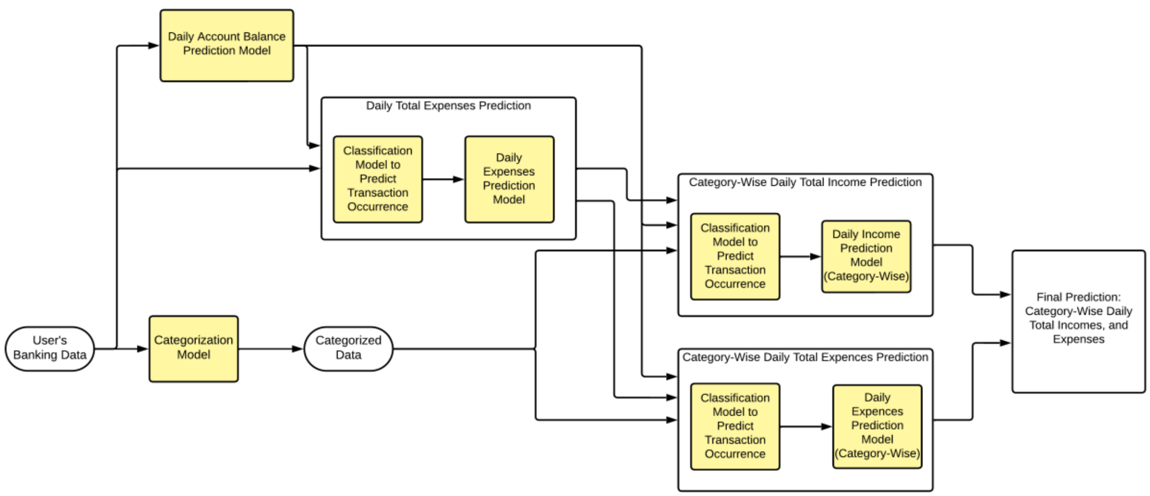


Figure 4.2.2.1 Prediction Flow

The prediction flow begins when user's banking data enters the categorization model, which automatically classifies transactions into meaningful categories using advanced natural language processing techniques.

This categorized data then flows through classification models that predict transaction occurrence, distinguishing between one-off transactions and recurring financial activities.

This critical preprocessing step addresses the challenge of sparse financial data where many categories contain zero values for extended periods.

The prediction pipeline branches into parallel processing streams that handle different aspects of financial forecasting. Daily income prediction models operate on a category-wise basis, analyzing historical patterns to generate category-wise daily total income predictions. Simultaneously, daily expenses prediction models process spending patterns to produce category-wise daily total expense predictions. A separate daily account balance prediction model forecasts future account balances by considering both income and expense predictions along with existing account states.

All these individual prediction streams converge to produce the final comprehensive prediction output, delivering category-wise daily total incomes and expenses. This multi-model approach ensures accuracy by specializing each model for specific prediction tasks while maintaining the ability to provide holistic financial forecasts.

4.2.3 Technology Implementation Strategy

The solution leverages cutting-edge technologies through strategic adaptation for personal finance management. The categorization system employs GTR-T5-large model for creating high-quality numerical representations from transaction descriptions, combined with DBSCAN clustering algorithm for automatic category discovery without requiring predefined cluster numbers.

For prediction modeling, the system implements a two-stage framework using Temporal Fusion Transformer (TFT) for category-wise predictions and N-BEATS for aggregate forecasting. This approach provides both detailed category-level insights and overall financial health indicators with high explainability and uncertainty quantification.

The conversational AI component utilizes LangChain-powered agents integrated with OpenAI's API and local Ollama instance running Llama 3.7 model. This hybrid approach balances performance, privacy, and cost by processing sensitive data locally while leveraging cloud capabilities for complex reasoning tasks.

The technical stack includes React Native with Expo for cross-platform mobile development, FastAPI for scalable backend services, and MongoDB for flexible data storage. The microservice architecture ensures scalability, maintainability, and efficient resource utilization while supporting real-time financial analytics and predictions.

Security is maintained through JWT (JSON Web Token) authentication for robust route protection, ensuring that all API endpoints and user sessions are properly authenticated and authorized. This token-based authentication system provides stateless security management while maintaining high performance and scalability across the distributed microservice architecture.

4.3 System Components

4.3.1 Users

The system is designed for Standard Users who have comprehensive access to all features of the personal banking assistant. These users can view and manage their personal financial information across multiple bank accounts, interact with the AI-powered chatbot for transaction information and system interaction, and set up and manage transaction categories according to their preferences. Users have full control over viewing both historical and predicted financial data, managing their bank accounts, and maintaining personalized to-do lists for financial tasks and commitments.

4.3.2 Inputs

The primary inputs to the system consist of users' past banking data, which includes complete transaction histories, account balances, and financial records from all connected bank accounts. Additionally, the system receives to-do list items manually added by users, representing their financial commitments, upcoming payments, and personal financial goals that require tracking and management.

4.3.3 Processes

The system employs a sophisticated multi-stage prediction process as illustrated in the system flow diagram. The process begins when user's banking data enters the categorization model, which automatically classifies transactions into meaningful categories. This categorized data then flows through classification models that predict transaction occurrence, distinguishing between one-off transactions and recurring financial activities. The prediction pipeline branches into parallel processing streams: daily income prediction models operating on a category-wise basis generate category-wise daily total income predictions, while daily expenses prediction models similarly produce category-wise daily total expense predictions. Simultaneously, a daily account balance prediction model forecasts future account balances. All these individual predictions converge to produce the final comprehensive prediction output, delivering category-wise daily total incomes and expenses that form the foundation for proactive financial management and early warning systems.

4.3.4 Outputs

The system generates comprehensive outputs including detailed income, expense, and balance summaries presented through intuitive visualizations. Users receive transaction category breakdowns displayed in interactive pie charts, along with historical and predicted financial graphs that illustrate trends and forecasts. The system provides specific transaction predictions for upcoming periods and accurate account balance predictions to prevent overdrafts. All transactions are organized into categorized lists for easy review and management. The to-do list feature delivers items and automated reminders for financial tasks, while the AI-powered chatbot provides intelligent responses containing information about past transactions, detailed explanations of predictions, transaction summaries, and personalized financial guidance. Critical to user financial health, the system generates proactive warnings when users are predicted to exceed their available balances, enabling preventive action before financial difficulties arise.

Chapter 5: Analysis and Design

5.1 Introduction

This chapter presents the architectural design and functional breakdown of the Personal Banking Assistant System. It begins with a top-level architecture diagram, followed by detailed descriptions of each module, their roles, and interactions. This modular structure ensures scalability, maintainability, and alignment with implementation goals, supported by design artifacts like use case diagrams.

5.2 Full system architecture

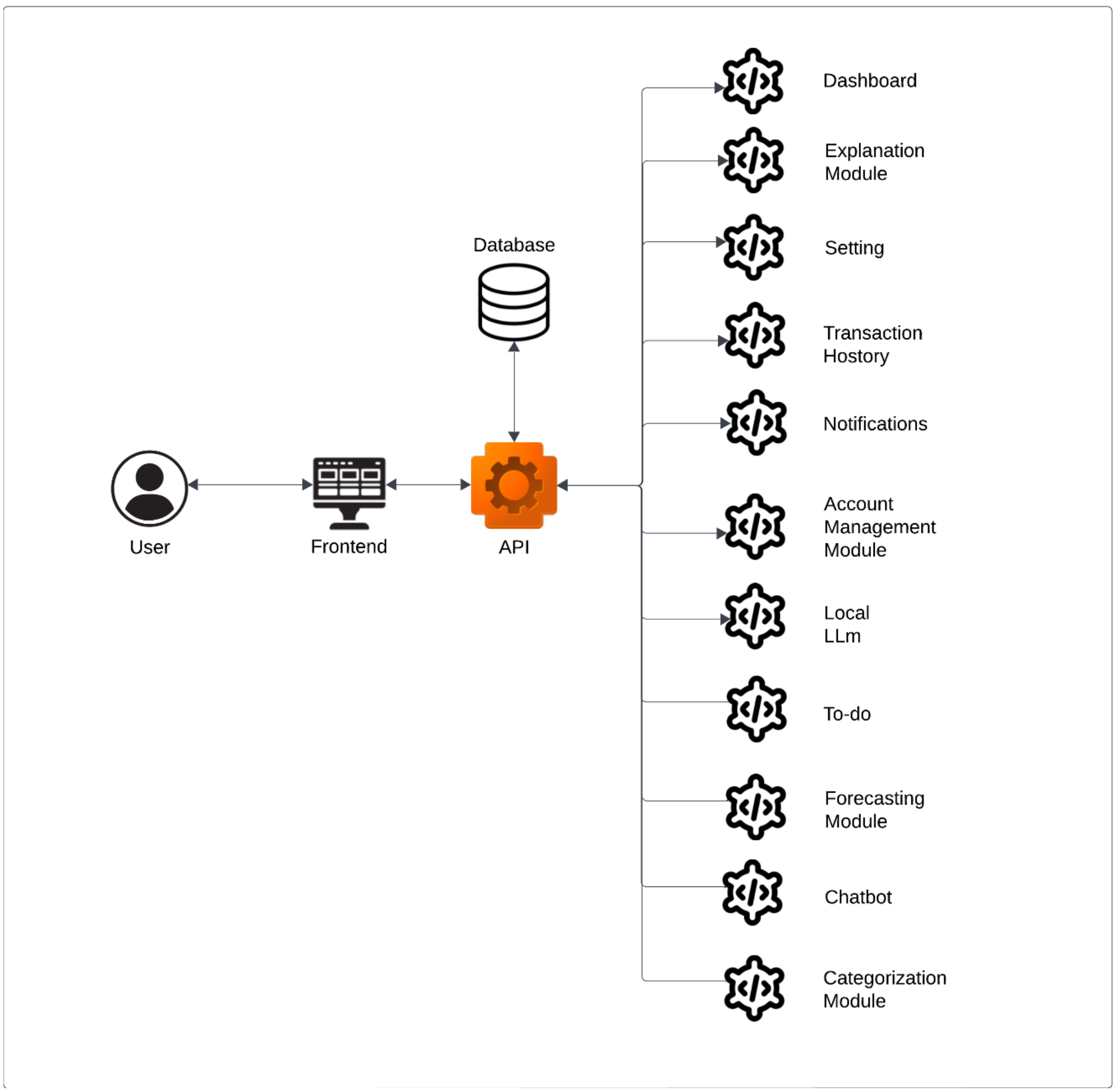


Figure 5.2.1 architectural design and functional decomposition of the Personal Banking Assistant System

5.3 User authentication module

The User Authentication Module is responsible for handling all tasks related to user identity and secure access. It ensures that only legitimate users can access the system by supporting secure registration and login functionality. This module has two main subcomponents:

- User Registration Module**:** Allows new users to sign up by entering their personal and contact details. The module stores this information securely in the backend database, applying encryption and validation as needed
- User Login Module: Authenticates existing users by verifying their credentials. It provides session management and token-based authentication to ensure secure access to user-specific features.

This module interacts with the Settings Module to handle user profile updates and with the Dashboard Module, which becomes accessible only after a successful login. It also connects with the Database Layer to store and retrieve authentication-related data securely.

5.4 To-do Module

The To-do Module enables users to manage their personal financial tasks effectively. It supports the core functionalities of adding new to-do items, editing or removing existing tasks, and marking tasks as completed. These tasks may include bill payments, budgeting activities, or custom financial reminders defined by the user.
The To-do Module consists of the core features such as adding tasks with optional details such as due dates and reminders, marking tasks as completed to update the task list, deleting existing tasks from the list, and editing tasks to modify their content or details as needed.

This module interacts closely with:

Notification Module: Sends alerts or reminders based on task due dates and user-configured notification preferences.

5.5 Dashboard module

The Dashboard Module serves as the system's primary visual interface, delivering key financial insights through concise, intuitive, and interactive displays. It enables users to quickly assess their financial status via graphical summaries and intelligent suggestions.

The module presents total and per-account income, expenses, and balances over user-defined timeframes. It highlights major spending categories and displays historical (past 100 days) and forecasted (next 30 days) trends using interactive charts. When the predicted balance for a selected account falls critically low, it suggests an alternative account with a surplus to support informed decisions. Notifications, including reminders and alerts, are shown via a bell icon.

To function effectively, the Dashboard Module integrates with the Forecasting Module for predictive data and the Notification Module for delivering alerts triggered by financial thresholds or events.

5.6 Setting Module

The Settings Module allows users to customize their profiles and preferences, enhancing personalization, security, and overall user experience within the financial forecasting system. It provides a user-friendly interface for managing account-level settings and maintaining control over system interactions.

Key sub-components include the Profile Update Sub-Module for modifying personal details, the Change Password Sub-Module for secure password updates integrated with the User Authentication Module, and the Notification Preferences Sub-Module for customizing alert and reminder delivery.

This module interacts with the User Authentication Module to validate users before executing sensitive actions, and with the Notification Module to apply user-defined preferences for system alerts.

5.7 Transaction History Module

Displays all historical financial transactions from both credit card and savings accounts, as requested by the user, account-wise. Users can filter transactions by date range and amount for savings accounts, or select a time period for credit cards, facilitating better financial tracking and analysis.

The Transaction History Module provides users with access to their complete financial transaction records. It supports viewing of past transactions from both credit card and savings accounts, allowing users to track and analyze their financial activity over time.

This module enables users to view all historical transactions in an account-wise manner, providing separate views for each linked account. It allows filtering of transactions by date range, such as the last week or last 30 days, and by amount range for savings accounts. For credit card accounts, users can filter transactions based on a selected time period.

5.8 Account management module

The Account Management Module allows users to manage their linked financial accounts, including adding new credit or savings accounts and removing existing ones. This ensures their financial profile remains current and accurately reflects all active income and expense sources.

When an account is added, metadata such as account type, name, and masked number is securely stored and linked to the user profile. Upon removal, all associated data is detached and excluded from future calculations or predictions.

Core features include adding new accounts, removing existing ones, and viewing a list of all linked accounts. The module ensures secure operations by integrating with the User Authentication Module to verify user identity before allowing account modifications.

5.9 Notification Module

The Notification Module is responsible for keeping users informed about important events and updates within the personal banking assistant. It delivers timely alerts such as reminders for upcoming tasks and low savings warnings to enhance the user's financial awareness and engagement.

These alerts help users take appropriate actions — for example, paying bills on time, switching to a different account when balance is low etc.

The Notification Module interacts with several other modules to deliver timely and relevant alerts. It communicates with the To-do Module to send reminders about pending financial tasks or due dates. It also works with the Forecasting Module to trigger alerts based on predicted expenses, incomes, or low account balances. The Settings Module defines user preferences for the types and delivery methods of notifications. Additionally, the Dashboard Module displays visual notification indicators and may highlight alerts within the financial overview.

5.10 Chatbot Module

The Chatbot Module functions as the primary conversational interface within the personal banking assistant. Its core objective is to provide users with real-time financial guidance by interpreting natural language queries and returning relevant information pertaining to spending patterns, savings, future financial forecasts, and general system functionalities.

This module is designed using an agent-based architecture, in which a Large Language Model (LLM) serves as the central decision-making component. The LLM analyzes user inputs and determines the most suitable internal tool or module to invoke in order to generate a contextually accurate and informative response. The chatbot not only answers direct queries but also provides explanations to enhance user understanding of system outputs.

By facilitating intuitive, language-based interactions, the chatbot significantly enhances the overall user experience and serves as a central access point for the system's core functionalities.

5.11 Categorization module

The Categorization Module is designed to automatically classify user financial transactions based on the description data found in their bank statements. Instead of relying on predefined manual inputs, this module intelligently analyzes textual patterns in the transaction descriptions to determine the most appropriate category for each entry.

When a user uploads their bank account to the system it takes a bank statement from the bank, each transaction is examined by the module. Through the use of keyword matching similarity calculations, the system determines the nature of the transaction and assigns it to a relevant category.

This automated classification enhances personalization by enabling more accurate financial forecasting for each category and providing insightful visualizations and trend analyses based on categorized data.

Module Interactions:

The Forecasting Module uses categorized transaction data to predict income and expenses at the category level. The Dashboard Module presents spending breakdowns, highlights top spending categories, and shows related trends. The Explanation Module leverages categorized data to generate clear, human-readable insights into financial behavior and forecasts.

5.12 Forecasting module

The Forecasting Module is a core component of the system, responsible for generating predictive insights related to a user's financial activity. By analyzing historical transaction patterns, this module forecasts future financial values, thereby enabling users to make informed financial decisions and engage in proactive financial planning.

This module leverages advanced time series forecasting techniques, incorporating two state-of-the-art deep learning models based on the availability and quality of historical data:

- Temporal Fusion Transformer (TFT): Applied in scenarios where sufficient historical data exists. TFT is used primarily for producing category-wise predictions of future expenses and incomes over the next 30 days.
- N-BEATS (Neural Basis Expansion Analysis for Time Series): Utilized for data-scarce situations or where more generalized forecasting is required. N-BEATS is used for:
    - Generating category-wise income and expense forecasts when data is limited.
    - Predicting total daily expenses.
    - Forecasting daily total account balances.

The combination of these models ensures that the forecasting component is both flexible and accurate across a variety of data conditions. And also these predictive outputs are critical for various system functionalities, such as alert generation, financial visualization, and explanation of projected financial trends.

Module Interactions:

The Forecasting Module collaborates with multiple components to deliver comprehensive insights. It receives labeled transaction data from the Categorization Module to enable category-level predictions. Forecasting results are used by the Notification Module to trigger alerts, such as low balance warnings. The Dashboard Module visualizes forecast outputs, including income, expenses, and balance projections. Additionally, the Explanation Module generates natural language summaries of the forecasts, enhancing transparency and user trust.

5.13 Local llm module

The Local LLM Module is responsible for ensuring data privacy and confidentiality by handling all sanitization and de-sanitization of sensitive user information within the system. Its primary function is to protect user data—such as names, monetary amounts, dates, and bank-related details—when interacting with the third-party language model integrated in the chatbot.

When a user submits a query, the Local LLM Module analyzes the input and detects any potentially sensitive details. These are temporarily replaced with generic placeholder values to anonymize the content. This sanitized version of the query is then forwarded to the third-party LLM or passed to a relevant agentic tool via the chatbot module.

Before any tool is executed, the placeholders are replaced with the original values to ensure accurate processing. Once a result is generated, the module again checks the output for sensitive information, applies sanitization if necessary, and only then sends it back for formatting. Finally, the module restores the original values before presenting the response to the user.

This structured process ensures that sensitive user information remains protected, tools and chatbot components receive properly formatted data, and users receive secure, complete, and contextually accurate responses.

Key Responsibilities:

The module identifies sensitive details in user inputs and outputs, anonymizes and restores data throughout chatbot interactions, and maintains privacy without compromising tool functionality.

Module Interactions:

All user-chatbot communications pass through this module for privacy control. It also interacts with the database to securely store and later restore original values during the sanitization process.

5.14 Explanation module

The Explanation Module is designed to enhance the transparency, interpretability, and trustworthiness of the financial predictions generated by the system. It provides users with clear and context-aware explanations for each type of forecast, such as category-wise expense predictions, category-wise income predictions, and overall account balance forecasts. Forecasting models such as TFT (Temporal Fusion Transformer) and N-BEATS generate not only the predicted values but also additional outputs—such as attention scores, feature importances, and confidence-related metrics—that give insight into how and why a prediction was made. These metrics are passed to this module as inputs for explanation generation. The Explanation Module interprets model outputs and generates descriptive summaries that clarify key aspects of the forecast. It identifies which past behaviors or categories contributed most to the predicted value, explains the model's confidence level based on variance or uncertainty, and highlights any anomalies or unusual spending patterns that may have influenced the prediction. To improve user comprehension, these technical metrics are converted into natural language explanations by invoking a language model, which formats the information in a user-friendly and readable form. This step ensures that explanations are both accurate and easily understandable to non-technical users. Each prediction generated by the system is therefore accompanied by a corresponding explanation. This empowers users to not only view their future financial trends but also understand the reasoning behind those trends.

Key Responsibilities:

The module interprets raw model outputs, including attention scores and uncertainty ranges, and transforms them into clear, natural-language explanations. It provides insights into category-wise income and expense forecasts, as well as total balance predictions. By making the system's decisions more transparent and auditable, the module helps build user trust.

5.15 Use case diagram

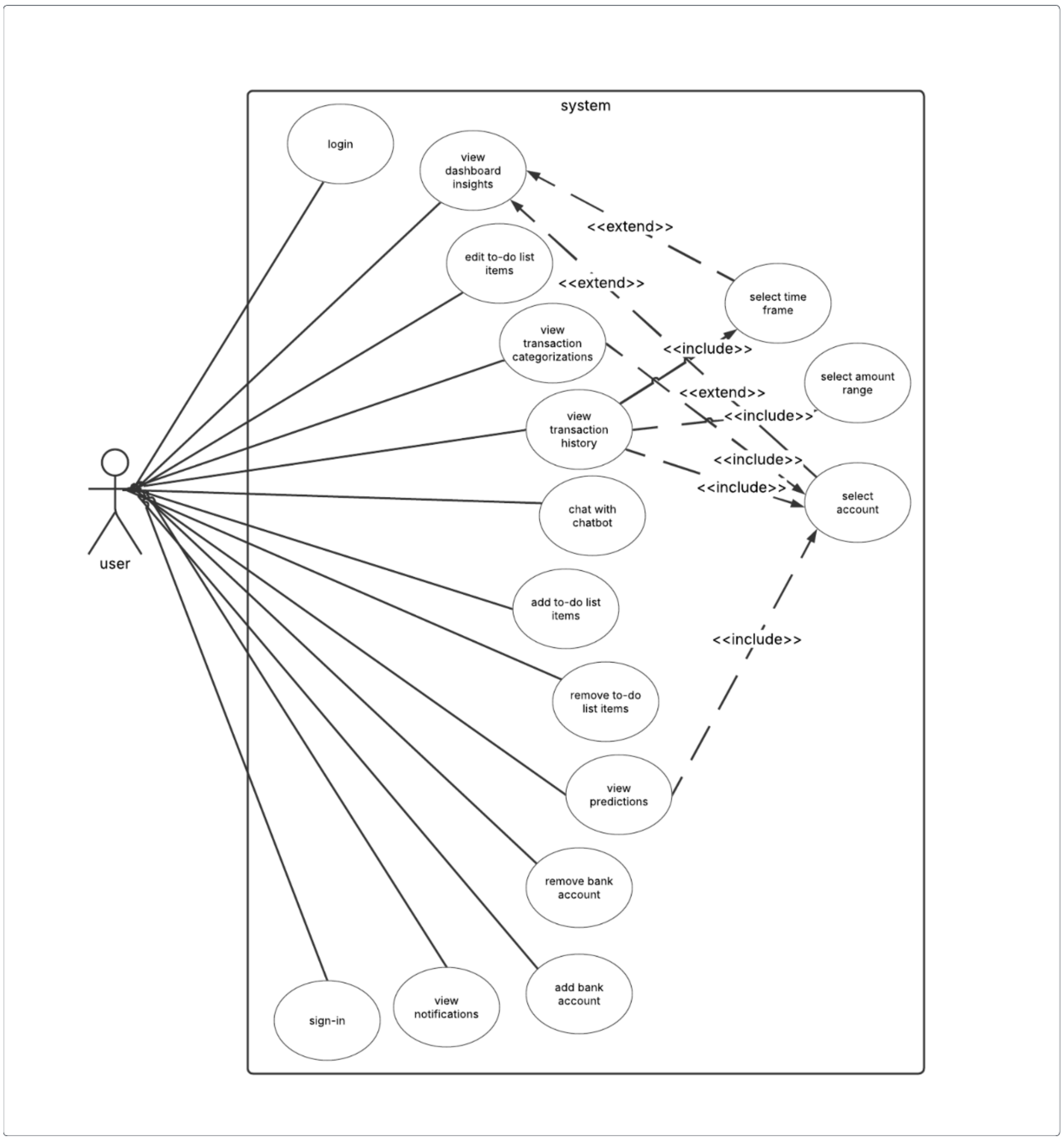


Figure 5.15.1 Use Case Diagram

5.16 Schema diagram

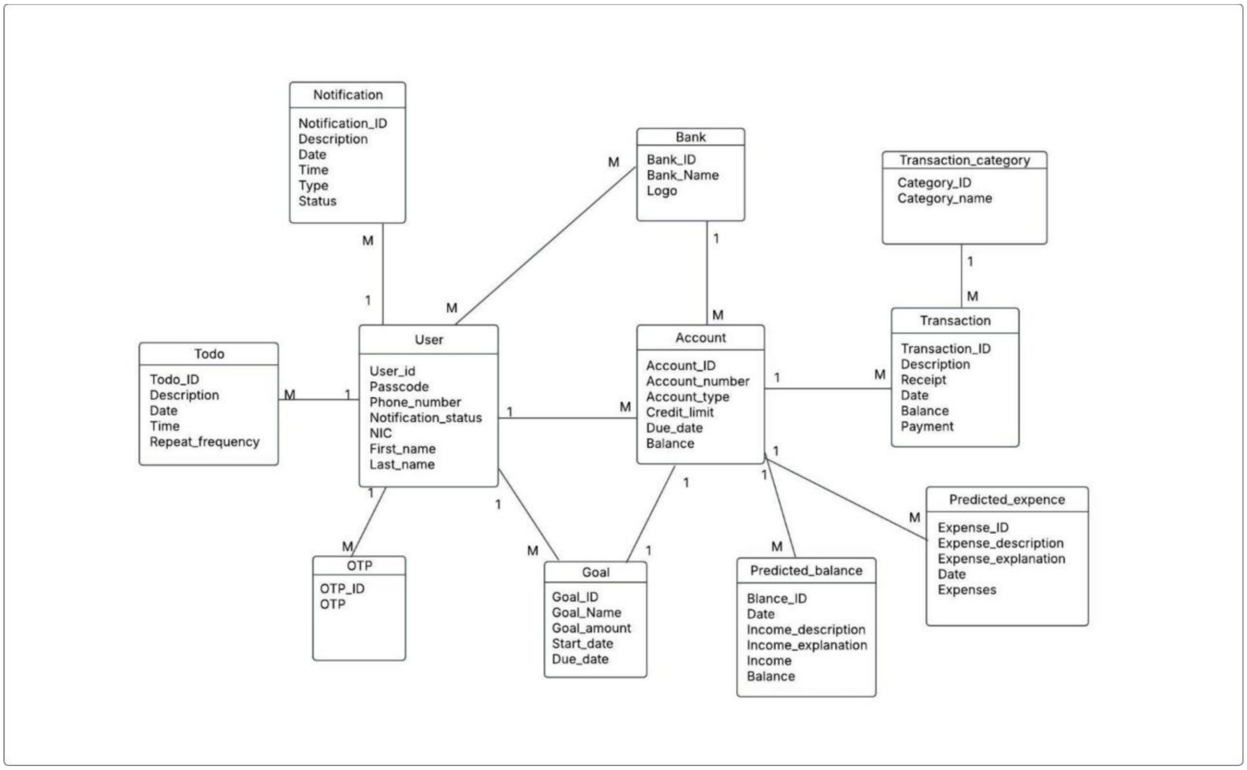


Figure 5.16.1 Schema Diagram

5.17 Summary

This chapter provides a structured overview of the system's architecture and design. The top-level architectural diagram was introduced, highlighting the key functional modules that make up the Personal Financial Forecasting System. Each module was described in detail, outlining its purpose, responsibilities, and interactions with other components in the system.

The modular design presented in this chapter ensures that the system is scalable, maintainable, and aligned with the overall functional objectives. Supporting diagrams were appended where necessary to aid in the understanding of complex processes. This design framework serves as the foundation for the implementation efforts described in the following chapter.

Chapter 6: Implementation

6.1 Introduction

This chapter outlines the implementation of each module described in the system's architectural design. It translates the conceptual framework into functioning software components that together build the personal banking assistant. The aim is to demonstrate how the system's design was practically realized, ensuring alignment with the intended architecture. Each module is discussed with its associated technologies, frameworks, and tools. Implementation details include internal logic, supported by flowcharts, pseudocode, and relevant diagrams.

The objective of this chapter is to demonstrate how the proposed design was realized in practice, ensuring consistency with the design phase, and providing clarity on the development process and operational behavior of the system. Detailed implementation artifacts such as screen captures, or extended listings are included in the Appendix to maintain readability.

6.2 User authentication module

The User Authentication Module is implemented to manage secure access to the system through user registration and login functionality. It ensures that only authorized users can access the application, maintaining data confidentiality and integrity. This module uses JWT (JSON Web Token)-based authentication to handle secure credential storage and session management.

The module is divided into two main subcomponents: the User Registration Module and the User Login Module. Each subcomponent has a clearly defined flow and interacts with the backend database and JWT service to provide authentication services.

6.2.1 JWT Authentication in the System

The system uses JWT (JSON Web Token) for secure, stateless user authentication. Upon registration, the user's NIC number and a securely hashed passcode are stored in the database. During login, credentials are validated, and if successful, a signed JWT containing user identity details is issued to the client. This token, stored client-side (e.g., in local Storage or cookies), must accompany subsequent requests and is verified by the server to authorize access.

6.2.2 User Registration Module

The below flow chart describes the sequence of processes happening inside the user registration module.

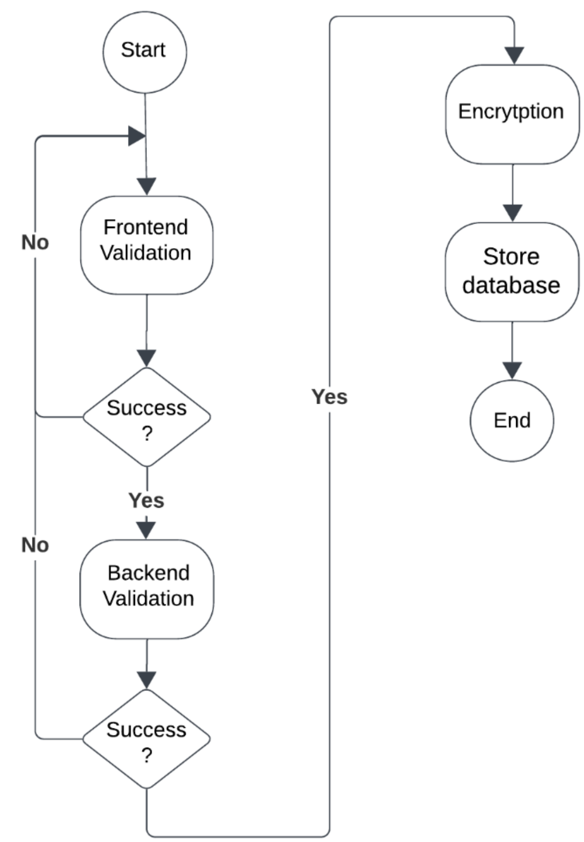


Figure 6.2.2.1 User Registration Module Flow

6.2.3 User login module

The below flow chart describes the sequence of processes happening inside the user login module

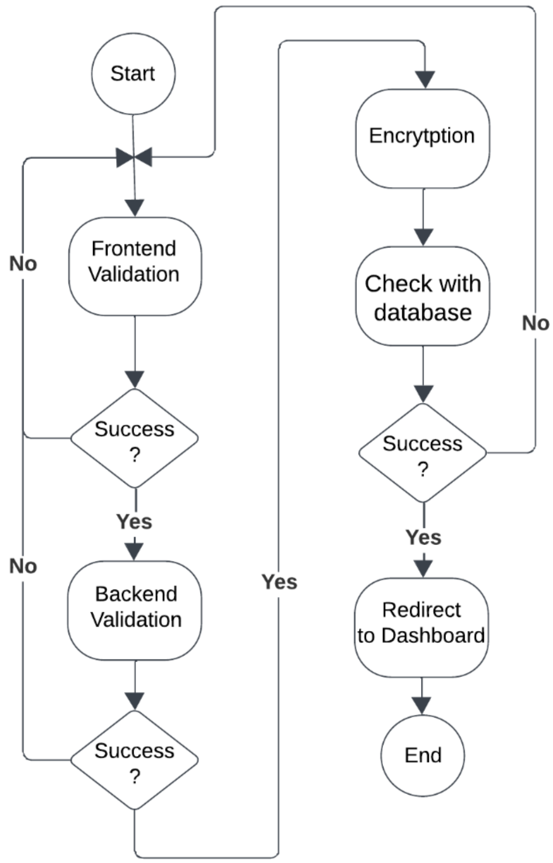


Figure 6.2.3.1 User Login Module flow

6.3 To-do Module

When a user adds a to-do item it is necessary to add a description. But it is optional to add due date, reminder and repeat frequency. Users can add one of them or all of them or none of them. When a user edit it is optional to edit to-do description and all other things.

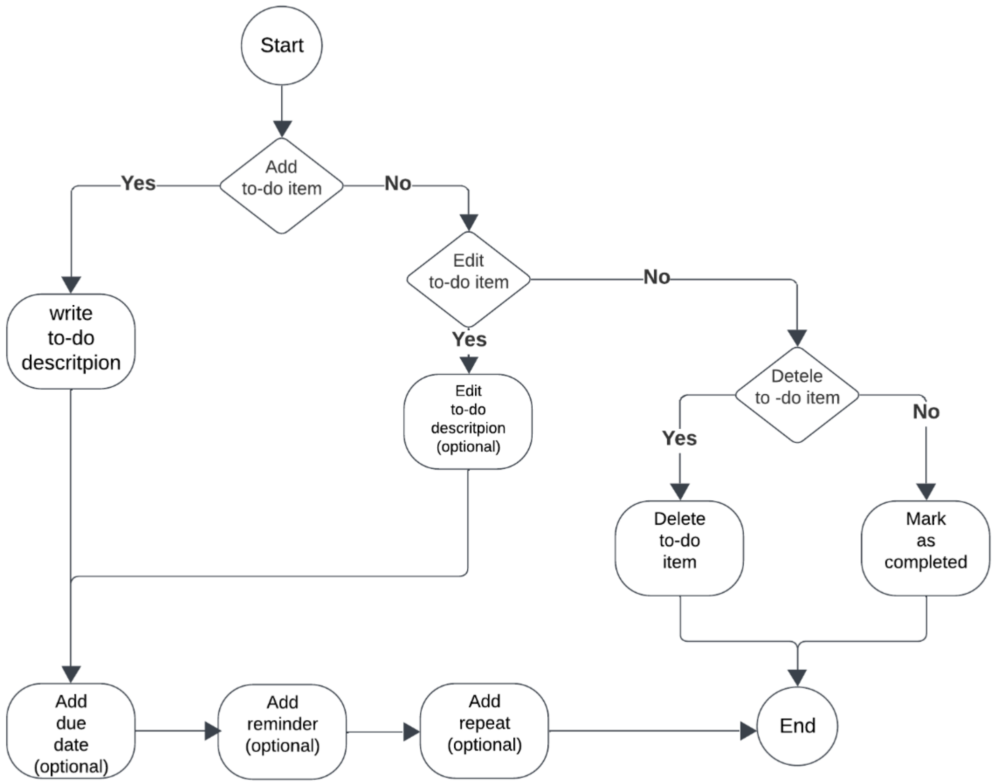


Figure 6.3.1 Todo Module Flow

6.4 Dashboard Module

The Dashboard Module was implemented using web technologies that support dynamic data visualization and real-time updates. It retrieves and displays both historical and predicted financial data, combining information from multiple backend modules.

Software Technologies Used:

- Frontend: React native expo, Chart.js / D3.js / Recharts for rendering graphs and charts
- Backend APIs: Fast API
- Database: MongoDB
- Styling: Tailwind CSS / Bootstrap
- Icons: FontAwesome or React Icons (for bell icon notifications)

Figure 6.4.1 Dashboard Module Flow

6.5 Setting Module

The Settings Module enables users to manage their personal information, security credentials, and notification preferences. It comprises three key sub-sections, each responsible for specific aspects of user customization and system interaction.

Technologies Used :

- Frontend: React native expo for UI forms
- Backend: FastAPI
- Database: MongoDB
- Authentication: JWT authentication
- Storage: AWS S3 Bucket (for profile picture upload)

Sub-sections and Functional Descriptions:

The Settings Module comprises three functional sub-sections. The Profile Update section allows users to modify their name, phone number, and profile image, with changes reflected in the database and profile pictures uploaded to AWS S3, storing the image URL accordingly. The Change Password section verifies user identity through the Authentication Module before securely updating credentials, including password strength validation. The Notification Preferences section enables users to toggle notifications on or off, with preferences stored in the database and accessed by the Notification Module during message generation.

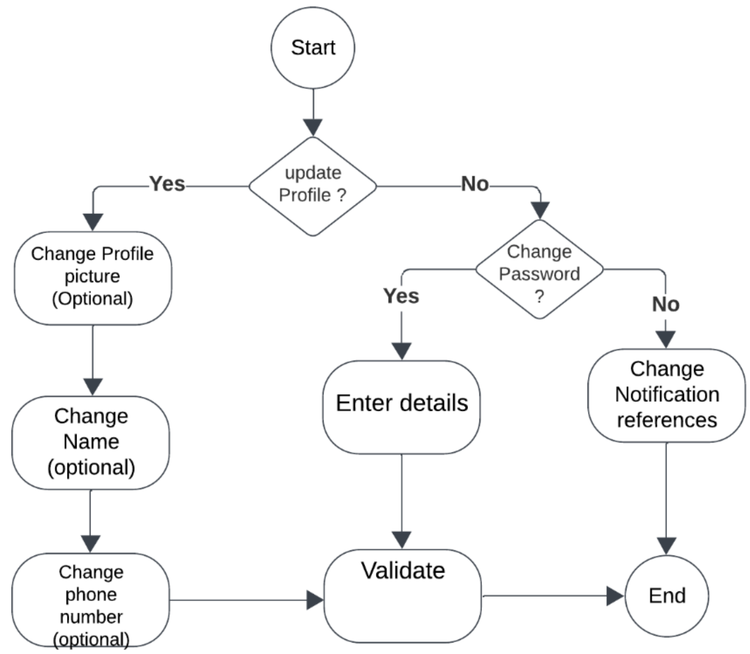


Figure 6.5.1 Profile Update Flow

6.6 Transaction History Module

The Transaction History Module is responsible for retrieving and displaying all past financial transactions linked to the user's credit card and savings accounts. It allows filtering by account, date range, and transaction amount to help users analyze their spending and income patterns effectively.

Technologies Used:

- Frontend: React native expo
- Backend: FastAPI
- Database: MongoDB
- Filtering: Handled via backend query logic and frontend filtering

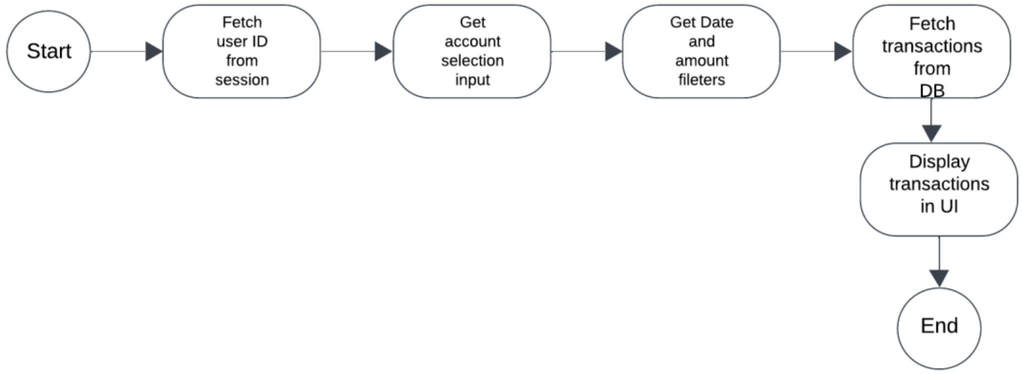


Figure 6.6.1 Transaction History Module Flow

6.7 Account management module

The Account Management Module is responsible for allowing users to add, remove, and view their financial accounts within the system. This module is critical for keeping the user's financial profile accurate and synchronized with real-world banking sources. Each account added by the user can be either a credit or savings account and is linked to the user profile stored in the backend. This uses **React Native Expo** for the frontend, **FastAPI** for the backend, **MongoDB** for the database, and **JWT authentication** for managing user authentication and access control.

Functional Features:

- Add Account: Allows users to enter new account details (Bank, account number, NIC) and save it securely.
- Remove Account: Deletes the selected account from the user's list.
- View Accounts: Lists all currently linked accounts in the user's profile, with account type indicators, account numbers and available balance

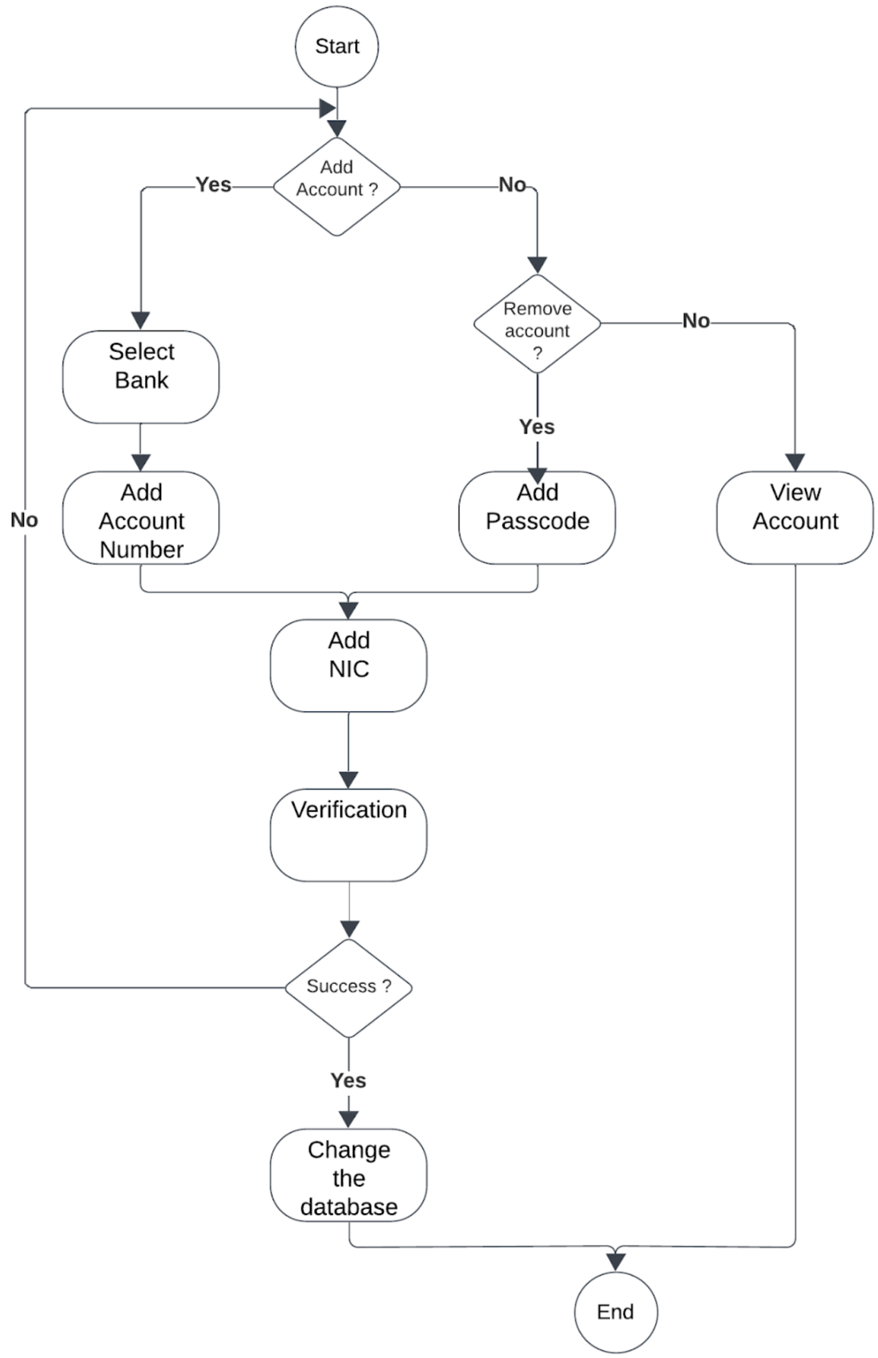


Figure 6.7.1 Account Management Module Flow

6.8 Notification Module

The Notification Module delivers real-time and scheduled alerts to users about critical financial events, such as low balance warnings and upcoming tasks. It integrates WebSocket-based in-app notifications and optional mobile push alerts. Implemented using FastAPI for API and WebSocket handling, MongoDB stores notification data and enables real-time updates via Change Streams. Expo Push Notification Service supports mobile alerts, while the React Native Expo frontend displays notifications and badge icons, ensuring timely and user-friendly communication.

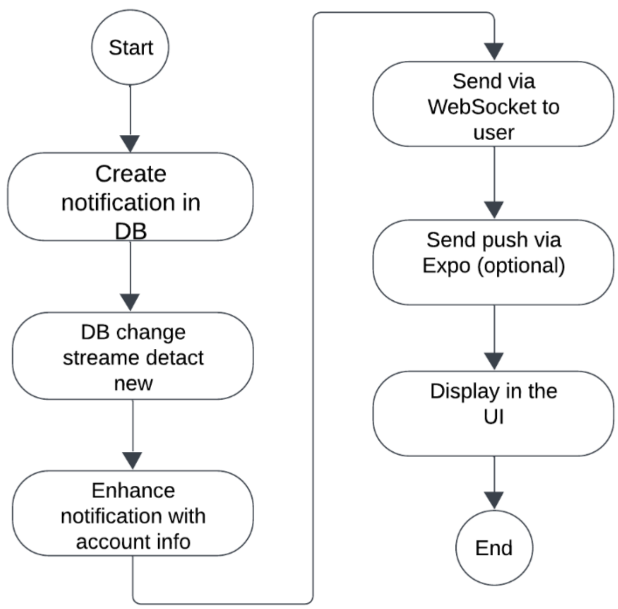


Figure 6.8.1 Notification Module Flow

6.9 Chatbot Module

The Chatbot Module functions as a single-agent system equipped with specialized tools to autonomously handle user queries. Built using the LangChain framework, it integrates a Large Language Model (LLM) for reasoning and tool invocation. The chatbot interprets user intent, routes tasks to appropriate tools, and generates natural language responses. Gemini powers the LLM, LangChain enables tool orchestration and memory handling, and ChromaDB serves as a lightweight vector database for storing and retrieving contextual embeddings, ensuring accurate and context-aware interactions.

The chatbot includes the following tools, each responsible for a specific task:

Tool Name	Functionality
Get_total_spendings_for_given_time_period	Calculates total spending during a user-specified time period.
Get_total_incomes_for_given_time_period	Retrieves total income received within a specified duration.
Get_last_transaction	Returns the most recent transaction for a given user.
Get_monthly_summary_for_given_month	Summarizes income and expenses for a specific month.
Get_all_transactions_for_given_date	Lists all transactions that occurred on a given date.
Get_next_month_total_incomes	Predicts the total income expected for the next month using forecasting data.
Get_next_month_total_spendings	Predicts the total spending expected for the next month.
Get_next_income	Identifies the next predicted income transaction.
Get_next_spending	Identifies the next predicted expense transaction
Get_greeting_response	Handles casual interactions such as greetings (e.g., “Hello”, “Good morning”).
Get_bank_rates	Responds to queries about current bank interest or loan rates.

Table 6.9.1 Chatbot Tools and Tasks

Implementation Flow:

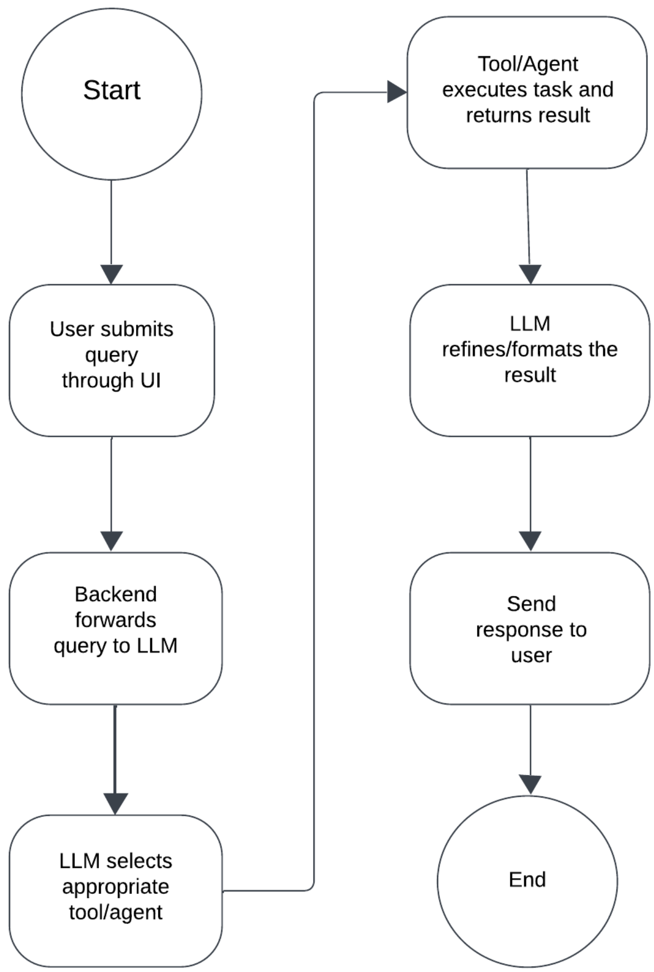


Figure 6.9.1 Chatbot Flow

6.10 Categorization module

The Categorization Module automatically classifies user financial transactions into meaningful categories by analyzing transaction descriptions from bank statements. Instead of relying on predefined rules, it uses unsupervised machine learning to detect patterns and group similar transactions. GTR-T5-Large transforms descriptions into semantic embeddings, capturing contextual meaning, while DBSCAN clusters these embeddings without needing predefined category counts. This adaptive approach ensures accurate categorization even when transaction formats vary across banks or vendors, enhancing the system’s flexibility and effectiveness in understanding diverse financial data.

Implementation Flow:

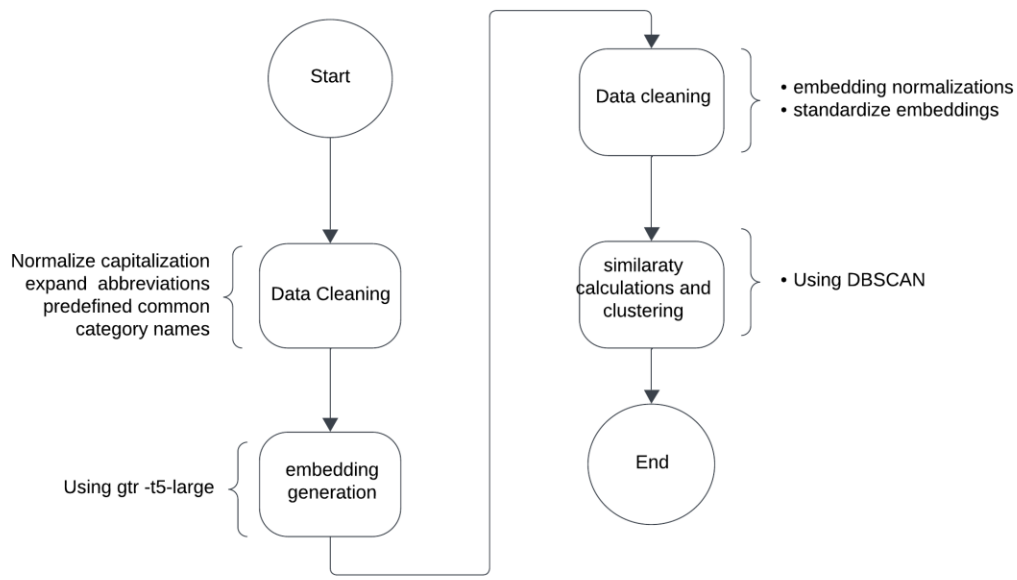


Figure 6.10.1 Categorization Module Flow

6.11 Forecasting module

The Forecasting Module delivers predictive insights into user financial behavior, including income, expenses, and account balance trends. It uses advanced time series models like Temporal Fusion Transformer (TFT) and N-BEATS, trained on historical transaction data. Input is preprocessed through categorization pipelines, transformed, and fed into the models based on context. Forecasts are stored in MongoDB and accessed via FastAPI for visualization, chatbot responses, and notification generation. PyTorch and PyTorch Lightning support model development, while Pandas and NumPy handle data preparation. This module ensures users receive accurate, personalized financial forecasts to support proactive decision-making.

Model Selection and Use Cases:

Forecast Type	Model Used	Description
Category-wise income forecast (30 days)	TFT	Used when sufficient historical data per category is available.
Category-wise expense forecast (30 days)	TFT	Supports personalized predictions for high-volume users.
Category-wise forecast (low data)	N-BEATS	Applied when category-specific data is limited.
Total daily expense forecast	N-BEATS	Used for generating system-level daily expenses.
Total account balance forecast (daily)	N-BEATS	Predicts future total balances across all user accounts.

Table 6.11.1 Prediction Models Selection

Implementation Flow:

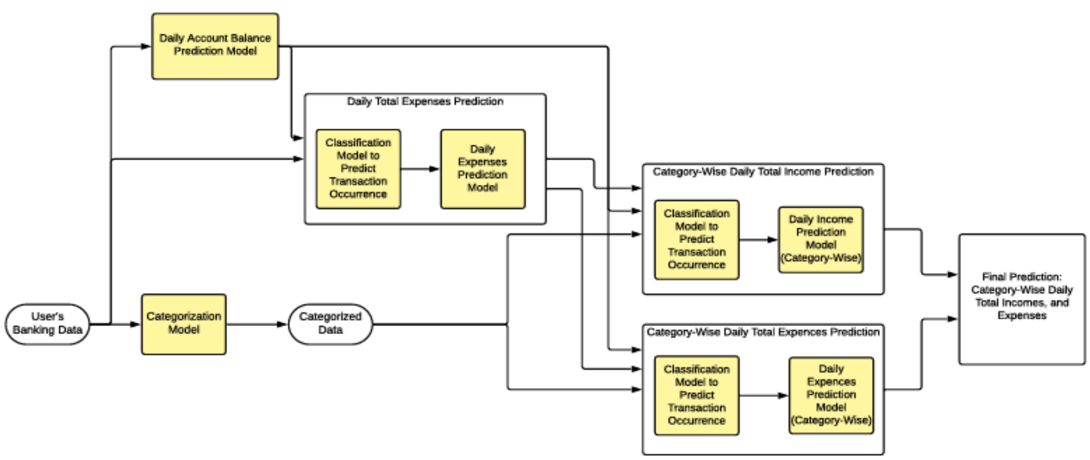


Figure 6.11.1 Prediction Flow

6.12 Local llm module

The Local LLM Module ensures privacy by securely handling sensitive information during chatbot interactions. It performs sanitization by replacing personal data with placeholders and de-sanitization to restore original values before presenting responses. This protects user confidentiality when communicating with third-party LLMs. The module uses LLaMA 3.2 for prompt-based entity extraction, MongoDB to store placeholder mappings, and FastAPI to expose internal endpoints. Custom regex patterns complement the LLM by detecting structured or ambiguous data formats. Together, these technologies ensure robust data privacy and secure handling of sensitive financial and personal details throughout the system.

Functional Workflow:

The sanitization and de-sanitization process follows a structured multi-stage workflow to ensure data security throughout system interactions. First, user input is processed by the local LLM to identify sensitive entities, which are replaced with placeholders (e.g., [amount_1], [bank_1]) while the original values are securely stored in the database. When the third-party LLM selects a tool for execution, these placeholders are de-sanitized to restore accurate inputs. Following tool execution, any sensitive data in the response is re-sanitized before being passed to the LLM for natural language generation. Finally, the output is de-sanitized once more before being presented to the user, ensuring both privacy and contextual integrity.

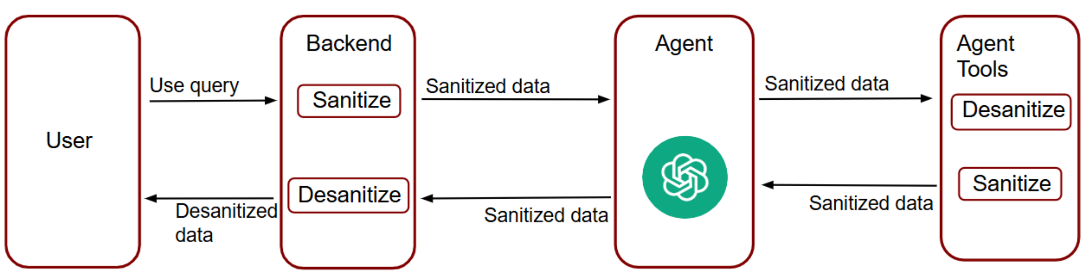


Figure 6.12.1 Sanitization Flow

6.13 Explanation module

The Explanation Module is implemented to provide users with human-understandable justifications for the financial predictions generated by the system's forecasting models. It ensures that each forecast—whether for income, expense, or account balance—is accompanied by an explanation that highlights why and how that forecast was made. This is achieved by processing the interpretability-related outputs of forecasting models and converting them into natural language through a locally controlled language model.

Technologies Used:

Temporal Fusion Transformer (TFT) – Produces attention weights and interpretable components used in explanation generation. Outputs attention scores and most important 3 features.

N-BEATS – Outputs residual, seasonal, trend components and prediction confidence over time.

- Confidence Interval & Attention Score Extractors – Custom utilities that derive model uncertainty, most influential features, and category contributions.
- Gemini LLM – Used to convert structured explanation metadata into user-friendly natural language.
- FastAPI – Hosts endpoints that generate and serve explanations based on user queries or forecast requests.
- MongoDB – Stores generated explanations along with their associated predictions for reuse and reference.

Functional Workflow:

The functional workflow begins with forecast retrieval, where the Forecasting Module generates predictions—such as category-wise expenses, incomes, or balances—along with metadata like attention scores or uncertainty indicators. For each prediction, a tailored prompt is constructed using the predicted values and relevant metrics to guide the generation of human-readable explanations. These prompts are then sent to a third-party LLM, which returns natural language interpretations of the structured data. The resulting explanations are stored in the database alongside their corresponding predictions and are later retrieved for display in interfaces such as the prediction view and account balance prediction view.

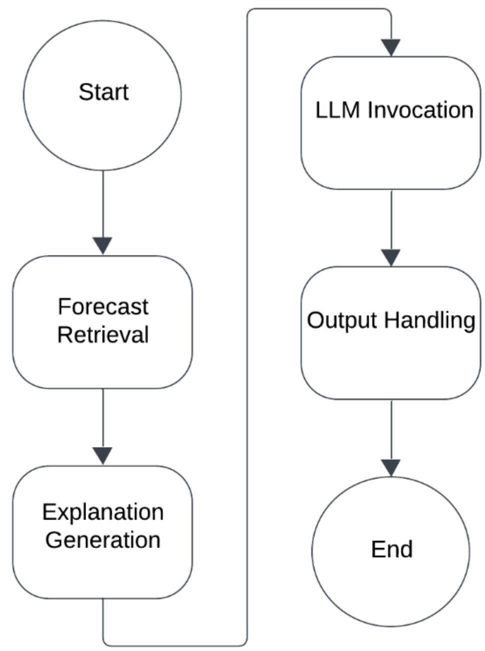


Figure 6.13.1 Explanation Flow

6.14 Summary

This chapter detailed the practical realization of the system modules introduced in the design phase. For each module, the implementation process was explained using appropriate technologies, libraries, and tools. Where relevant, internal workflows were illustrated using flowcharts and pseudo code.

The implementation strictly followed the design outlined in the previous chapter, ensuring consistency and traceability between the design and development phases. The use of modern technologies and secure coding practices helped ensure that the system meets performance, privacy, and usability requirements. Additional implementation artifacts are provided in the Appendix to support further understanding.

Chapter 7: Evaluation

7.1 Introduction

This chapter focuses on assessing the application`s performance, effectiveness, and achievements of its objectives. It outlines the methodologies and metrics used to evaluate functionality, usability, and overall impact on users. By conducting comprehensive testing, we aim to identify strengths and areas for improvement, ensuring the application meets its intended requirements and delivers a high-quality user experience.

7.2 Testing process and result

User login and registration

Test	Expected Result	Test Result
Successful user registration	user is successfully registered	passed
Failed user registration due to incomplete/invalid data input	user registration failed with appropriate message displayed.	passed
Successful user login	user is authenticated and granted access to the system	passed
Failed user login due to incomplete data input	user login failed with appropriate message displayed.	passed
Failed user login due to wrong credential input	user login failed with appropriate message displayed.	passed
Successful user logout	user session is terminated and redirected to login page.	passed

Table 7.2.1 Testing Process and Result of User Login and Registration

Dashboard module

Test	Expected Result	Test Result
Account selection dropdown works or not	successful switching between accounts	passed

Filter transaction summaries based on selected date range	change past transaction summaries according to the selected time frame.	passed
Accurate graph data update based on the selected account	graph data update successfully based on the selected accounts	passed
Display account specific alerts	Display warnings if account is about to face an insufficient balance situation	passed

Table 7.2.2 Testing Process and Result of Dashboard Module

Spending suggestions module

Test	**Expected Result**	**Test Result**
Surplus balance suggestions	Display surplus balances of each account if surplus available	passed

Table 7.2.3 Testing Process and Result of Spending Suggestions Module

Bank account management module

Test	**Expected Result**	**Test Result**
Add savings account	A savings account is added successfully and displayed in the account list.	passed
Add credit card account	credit card account is added successfully and displayed in the account list.	passed
Remove a savings account	authentications happen first and then account is removed	passed
Remove a credit card account	authentications happen first and then account is removed	passed

Table 7.2.4 Testing Process and Result of Bank Account Management Module

JWT authentication

Test	**Expected Result**	**Test Result**

Token expiring test by staying in a page without any interaction for 5 minutes	JWT token expires after 5 minutes and redirect to log in page	passed

Table 7.2.5 Testing Process and Result of JWT Authentication

Todo module

Test	**Expected Result**	**Test Result**
Add a Todo with a date	added Todo is displayed in pending section	passed
Add a Todo with a reminder	added Todo is displayed in pending section	passed
Add a Todo with just description	added Todo is displayed in pending section	passed
Add a Todo with a date and a reminder	added Todo is displayed in pending section	passed
Add a Todo with a date and a repeat	added Todo is displayed in pending section	passed
Add a Todo with a repeat without date	display an alert that ask to add a date	passed
Edit a Todo	Edited Todo is displayed in the page	passed
Remove a Todo	removed Todo is not displayed in the page	passed
Mark a Todo as completed	Todo is displayed in completed section	passed

Table 7.2.6 Testing Process and Result of Todo Module

Transaction categorization module

Test	**Expected Result**	**Test Result**
Account selection dropdown works or not	successful switching between accounts	passed
Correct initial page view	no account data displayed just initial view	passed

Remove a transaction from its category	after pop-up response, transaction is removed and added into uncategorized category	passed
Add an uncategorized transaction to a category	after pop-up response, transaction is removed from uncategorized category and added to the requested category	passed
Rename a category	new name for the category is displayed until next change	passed

Table 7.2.7 Testing Process and Result of Transaction Categorization Module

Agentic Chatbot

Test	Expected Result	Test Result
Awareness of next possible income	using predictions in database, agent respond with next predicted income	passed
Awareness of next possible expense	using predictions in database, agent respond with next predicted expense	passed
Ability of giving transaction summaries	using transaction history in database, agent respond with transaction summaries for asked time range	passed
Awareness of how to use the application	using RAG (retrieval augmented generation) tool, agent respond with instructions to use the application	passed
Avoid of answering unrelated questions	The agent responds respectfully saying it is not allowed to answer such questions	passed
Awareness of bank interest rates	using interest rates in database, agent respond with requested rates	passed
Ability of adding Todos	The agent adds Todo to the database and in Todo page, it is displayed	passed
Awareness of Todos	The agent retrieves Todos from the database, and respond	passed

Ability of answering financial related queries	The agent responds for asked queries	passed

Table 7.2.8 Testing Process and Result of Agentic Chatbot

Notification module

Test	**Expected Result**	**Test Result**
Insufficient balance notifications	successfully retrieve notifications for insufficient balances based on predictions	passed
Todo task reminder notifications	reminder Todo tasks via notifications at asked reminder times	passed
Alert via notification bell icon	when a new notification retrieve bell icon at the header in every page will be change and new notification count will be shown on top of the bell icon	passed

Table 7.2.9 Testing Process and Result of Notification Module

Settings module

Test	**Expected Result**	**Test Result**
Change profile picture	successfully change profile picture	passed
Change password	successfully change password with authentication steps	passed
Change notification on/off status	turn on/ turn off notifications based on the selection	passed

Table 7.2.10 Testing Process and Result of Settings Module

Transaction history view module

Test	**Expected Result**	**Test Result**
Account selection dropdown works or not	successful switching between accounts	passed
Correct initial page view	no account data displayed just initial view	passed

Filter transactions of a savings account based on selected date range and amount	change of available past transactions according to the selected time frame and amount.	passed
Necessary of time range and amount selection before searching in a savings account	if one of time range or amount selection is not selected before searching, display the alert and ask to select	passed
Switch between value and range in amount selection in a savings account	correctly switch between range and value selection options	passed
Necessary of time frame selection before searching in a credit card account	if a time frame is not selected when searching, display the alert and ask to select	passed

Table 7.2.11 Testing Process and Result of Transaction History View Module

Transaction prediction view module

Test	Expected Result	Test Result
Account selection dropdown works or not	successful switching between accounts	passed
Correct initial page view	no account data displayed just initial view	passed

Table 7.2.12 Testing Process and Result of Transaction Prediction View Module

Account balance prediction view module

Test	Expected Result	Test Result
Account selection dropdown works or not	successful switching between accounts	passed
Correct initial page view	no account data displayed just initial view	passed

Table 7.2.13 Testing Process and Result of Account Balance Prediction View Module

7.3 Model evaluation

Datasets used

- Dataset 1 **-** A real world dataset with 731 datapoints of a savings account from people`s bank.
- Dataset 2 - A real world dataset with 587 datapoints of a savings account from commercial bank.
- Dataset 3 - A real world dataset with 68 datapoints of a savings account from people`s bank.
- Dataset 4 - A real world dataset with 56 datapoints of a savings account from commercial bank.
- Dataset 5 - A real world dataset with 105 datapoints of a credits card account from DFCC bank.
- Dataset 6 - A real world dataset with 44 datapoints of a credits card account from DFCC bank.

Metrics used

We used RMSE (root mean squared error), MAE (mean absolute error) and MSE (mean squared error) for regression tasks and Precision, recall, F1 score and AUC score for binary classification tasks.

Account balance prediction by N-BEATS

Dataset	RMSE	MAE
dataset 1	1350.3021	737.8777
dataset 2	1823.4038	1137.6800
dataset 3	2455.5006	2028.6489
dataset 4	2350.21	1737.0077
dataset 5	1023.403	637.1100
dataset 6	955.0006	828.1182

Table 7.3.1 Evaluation Results of Account Balance Prediction by NBEATS

Total expense prediction by N-BEATS

Occurrence prediction – binary classification

Dataset	precision	recall	f1	AUC
dataset 1	0.8667	0.6842	0.7647	0.7488
dataset 2	0.8000	0.6154	0.6957	0.7714
dataset 3	0.5834	0.9231	0.6789	0.9842
dataset 4	0.6750	0.7058	0.6881	0.9456
dataset 5	0.7350	0.6900	0.7118	0.8124
dataset 6	0.6790	0.8125	0.7392	0.8477

Table 7.3.2 Evaluation Results of Total Expense Prediction By NBEATS: Occurrence Prediction

Amount prediction – regression

Dataset	MAE	RMSE
dataset 1	1420.8377	1792.3668
dataset 2	1595.5006	1628.6489
dataset 3	2551.89	2823.4566
dataset 4	1000.8612	1278.7378
dataset 5	1389.61	1659.1211
dataset 6	827.1288	930.33

Table 7.3.3 Evaluation Results of Total Expense Prediction By NBEATS: Amount Prediction

Category wise expense prediction by Temporal fusion transformer

Occurrence prediction – binary classification

Dataset	precision	recall	f1	AUC
dataset 1	0.5641	1.0000	0.7213	0.9937
dataset 2	0.8003	0.5842	0.6649	0.6981
dataset 5	0.8667	0.6842	0.7647	0.7488

Table 7.3.4 Evaluation Results of Category Wise Expense Prediction by TFT: Occurrence Prediction

Amount prediction – regression

Dataset	MAE	RMSE	RMSE (non-zero targets)

dataset 1	191.9271240	451.442410	363.4733698
dataset 2	211.832222	487.241032	320.1869788
dataset 5	143.129273	252.67777	198.2433111

Table 7.3.5 Evaluation Results of Category Wise Expense Prediction by TFT: Amount Prediction

Category wise income prediction by Temporal fusion transformer

Occurrence prediction – binary classification

Dataset	precision	recall	f1	AUC
dataset 1	0.8000	1.0000	0.8889	1.0000
dataset 2	0.0602	0.8333	0.1124	0.9506
dataset 5	0.5667	0.6842	0.5647	0.6488

Table 7.3.6 Evaluation Results of Category Wise Income Prediction by TFT: Occurrence Prediction

Amount prediction – regression

Dataset	MAE	RMSE	RMSE (non-zero targets)
dataset 1	211.83	487.34	274.81
dataset 2	198.64	287.55	210.99
dataset 5	134.44	283.68	250.11

Table 7.3.7 Evaluation Results of Category Wise Income Prediction by TFT: Amount Prediction

Category wise expense prediction by N-BEATS

Occurrence prediction – binary classification

Dataset	precision	recall	f1	AUC
dataset 3	0.7667	0. 7730	0.7647	0.8488
dataset 4	0.0543	0.8333	0.1020	0.8822
dataset 6	0.5667	0.6842	0.5647	0.6488

Table 7.3.8 Evaluation Results of Category Wise Expense Prediction by NBEATS: Occurrence Prediction

Amount prediction – regression

Dataset	MAE	RMSE	RMSE (non-zero targets)
dataset 3	191.92740	451.442410	363.4733698
dataset 4	211.8322	487.241032	320.1869788

dataset 6	143.1273	252.67777	198.2433111

Table 7.3.9 Evaluation Results of Category Wise Expense Prediction by NBEATS: Amount Prediction

Category wise income prediction by N-BEATS

Occurrence prediction – binary classification

Dataset	precision	recall	f1	AUC
dataset 3	0.8000	0.6154	0.6957	0.771493
dataset 4	0.5834	0.9231	0.6789	0.9842
dataset 6	0.6750	0.7058	0.6881	0.9456

Table 7.3.10 Evaluation Results of Category Wise Income Prediction by NBEATS: Occurrence Prediction

Amount prediction – regression

Dataset	MAE	RMSE	RMSE (non-zero targets)
dataset 3	211.8322	487.241032	320.1869788
dataset 4	143.1273	252.67777	198.2433111
dataset 6	112.3455	212.8999	200.8993

Table 7.3.11 Evaluation Results of Category Wise Income Prediction by NBEATS: Amount Prediction

7.4 Summary

This chapter presented front-end and back-end test cases, evaluating time series models. Testing confirmed alignment with project goals. Time series models performed reasonably well on complex, sparse data, with Optuna-based hyperparameter tuning improving accuracy. The pipeline proved robust for various challenging data conditions. Note that due to computational limitations, the number of trials conducted using the Optuna library was restricted to only 10. However, the recommended range for effective hyperparameter tuning is typically between 50 and 100 trials. Therefore, increasing the number of trials is expected to further improve the model's performance.

Chapter 8: Conclusion and further work

8.1 Achievements

A user friendly and user driven mobile application

We developed a user-friendly, user-driven mobile application using React Expo for the frontend and FastAPI for the backend. By integrating the latest Android features, the system stays current with mobile advancements. Users can adjust transaction categories, manage accounts, handle to-dos, view time range-based summaries, view amount-based summaries, update profiles, and customize functionality like notifications. These features demonstrate that the application not only supports bank account management but also meets personalized user needs. Thus, we successfully delivered a solution tailored to what users truly expect from such an application.

Detailed data analysis dashboard

We developed a detailed dashboard for analyzing user bank accounts, using visual aids like pie charts and graphs for an enhanced experience. It helps users understand their spending and gain an abstract overview. Features like user-selected time ranges and amount-based transaction summaries further enrich the dashboard, making it both informative and personalized for improved financial insight.

Advanced time series forecasting pipeline capable of handling diverse datasets

We developed an advanced time series forecasting pipeline using industry best practices, capable of handling diverse datasets that are sparse, imbalanced, small, complex, and non-seasonal. Account-wise hyperparameter tuning ensured model generalization across users. Forecasting was enabled for both savings and credit card accounts without restrictions. Custom loss functions were created to enhance prediction accuracy. Overall, we successfully delivered an industry-grade forecasting solution for bank transaction prediction.

Alert users for upcoming potential insufficient balances

We implemented a notification system to alert users about potential insufficient balances using predictive analysis. By forecasting future financial states, the app detects possible low-balance situations and notifies users with detailed alerts. This proactive feature helps users identify and address such issues early, preventing financial difficulties and ensuring better control over their account activities.

Enhanced transaction categorization based on both symbolic AI and non-symbolic AI approaches

We were able to provide an advanced clustering pipeline that categorizes user account transactions with high accuracy. It provides the user with a view of what categories they are engaging with in day-to-day life.

Privacy secured agentic chatbot

We developed a privacy-secured agentic chatbot that interacts intelligently while ensuring no personal data is shared with external APIs. This guarantees user privacy, allowing for free and confidential interaction.

Allowing management of multiple bank accounts from different banks

We were able to provide a solution, huge advancement in Sri Lankan mobile banking applications by introducing a mobile banking application that can handle multiple bank accounts from different bank accounts. It makes ease for users as they do not need to switch between different mobile applications to analyze their bank accounts.

Secure user authentication and user data

We secured user authentication using JWT and encrypted all private data, ensuring no raw sensitive data is stored. Only the user can access their bank information, eliminating privacy concerns and potential data leaks.

Explanations for predictions

Without just providing user upcoming predictions and letting them have no idea where those predictions come from, we were able to provide them with user friendly explanations for each of the predictions given to the users. It makes more sense for users to deal with those predictions and make accurate financial decisions.

8.2 Problems encountered

Handling imbalanced datasets

Handling imbalanced datasets was a key challenge due to the sparse nature of bank transaction histories, especially in regions like Sri Lanka where card usage is limited. This caused time series data to be heavily biased toward zero values. Although intermittent time series forecasting methods like TSB (Teunter-Syntetos-Babai) were explored, they didn't meet our complexity and interpretability needs. To address this, we designed custom loss functions tailored to handle data imbalance effectively. These functions helped us achieve more accurate and balanced forecasting, allowing the system to learn meaningful patterns even from highly imbalanced and sparse transaction datasets.

Handling small datasets

Handling small datasets was a challenge, especially for users with newly created bank accounts and limited transaction history. Our initial transformer-based models struggled with such cases. After researching alternatives, we built a new pipeline capable of handling complex, non-seasonal datasets while offering interpretability and uncertainty. This solution now works with as little as three months of data. For accounts with less than three months of data, the application limits forecasting but still offers all other features, ensuring functionality for both new and long-term users effectively.

Generalize forecasting models for every user

To make our application meet industry standards, we aimed to generalize forecasting models for all users. Initially, we tried building a single model for everyone, but this approach failed due to user-specific transaction categories and unique data characteristics. Techniques like model fine-tuning, freezing, and meta-learning also posed challenges, especially with complex, sparse datasets. Additionally, category-wise relationships require overly complex pipelines. After analyzing industry practices, we adopted account-wise hyperparameter tuning. This approach allowed us to personalize models efficiently while maintaining accuracy and scalability, making our forecasting pipeline more adaptable to each user's individual transaction behavior.

8.3 Limitations

Need to have at least 3 months of data to have prediction

Current forecasting models are not capable of performing well if there is not at least 3 months of transaction data. Therefore, the user must wait up to 3 months to get a prediction facility for that account. But other features will be still working for them.

Dependency on Gemini API

Agentic chatbot uses Gemini API as its LLM (large language model). Therefore, there is a risk of failing API connection due to server crashes or due to any other possible cautions. So currently the application is running assuming these API`s will not face any trouble.

Dependency on LangChain framework

The agentic chatbot is built using LangChain, an open-source framework that can change over time. These changes may introduce compatibility issues, require updates, or necessitate adjustments to the system. Staying up to date with the latest developments in LangChain and adapting the system accordingly can be challenging and resource intensive.

No data entry mechanism

Not yet integrated with banks to automate data entry mechanism daily. It is required to have partnerships with banks and have a data entry mechanism to do daily predictions like tasks.

8.4 Further work

Integrating the application with banking systems.

Application`s data entry happens manually now. But it is required to integrate this with banks and get user data through banks directly.

Deploy forecasting pipeline with azure ML studio

Forecasting pipeline needs to be deployed in azure ML studio to ensure smooth operation of the pipeline. Concepts like model drift and data drift need to integrate with pipeline to ensure cost reduced, efficient model operations like time to time retraining or hyper parameter tuning. Daily data entry automation after bank integration is a prerequisite for this to ensure full pipeline is working.

Two separate databases to store user data and user`s transaction histories

This will ensure more security for the privacy of user data.

8.5 Conclusion

Our AI-Powered Mobile Banking Assistant marks a major advancement in delivering smart, user-centric financial tools. Key achievements include a customizable mobile interface, detailed transaction analysis, advanced forecasting, multi-account support, secure authentication, and a privacy-aware agentic chatbot. These features offer users better control and insights into their finances. However, limitations exist—reliance on APIs like Gemini and LangChain poses maintenance risks, and the lack of direct bank integration restricts automated data entry. Moreover, forecasting needs at least three months of transaction history, excluding new users. Future improvements include direct bank integration for real-time data, deploying forecasting with Azure ML Studio, handling model drift and retraining, and using separate databases for privacy. These enhancements will evolve the assistant into a secure, intelligent, and fully integrated financial partner.

References

[1] S. D. Talasila, “AI-Driven Personal Finance Management: Revolutionizing Budgeting and Financial Planning”.

[2] M. J. Sluijmers, “Predicting Future Incoming Bank Transactions By Detecting Recurring Transaction Sequences,” Eindhoven University of Technology, 2021.

[3] WONGA, “WONGA: The Future of Personal Finance Management – A Machine Learning-Driven Approach for Predictive Analysis and Efficient Expense Tracking”.

[4] G. M. G. F. D. K. Dimitrios Kotios, “Deep learning enhancing banking services: a hybrid transaction classification and cash flow prediction approach,” *IEEE International Conference on Artificial Intelligence for Cyber Security (AICS),* pp. 1-8, 2022.

[5] S. K. P. M. Devendra Prakash Jaiswala, “Customer Transaction Prediction System,” *2020 International Conference on Computational Performance Evaluation (ComPE),* p. 619–623, 2020.

[6] F. O. &. K. Lindh, “Deep Learning Techniques for Bank Transaction Categorization: A Collaborative Study with Kreditz,” KTH Royal Institute of Technology, 2021.

[7] G. Petnehazi, “Recurrent Neural Networks for Time Series Forecasting,” Doctoral School of Mathematical and Computational Sciences, University of Debrecen, 2019.

[8] S. &. S. J. Hochreiter, “Long short-term memory,” *Neural Computation,* vol. 9, no. 8, pp. 1735-1780, 1997.

[9] S. J. &. L. B. Taylor, “Forecasting at scale,” *The American Statistician,,* vol. 72, no. 1, pp. 37-45, 2018.

[10] K. G. C. C. G. F. R. M. M. M. D. A. .. &. S. F. Olivares, “Neural basis expansion analysis with exogenous variables: Forecasting electricity prices with NBEATSx.,” *International Journal of Forecasting,* vol. 39, no. 2, 2023.

[11] H. Z. S. P. J. Z. S. L. J. X. H. &. Z. W. Zhou, “Informer: Beyond efficient transformer for long sequence time-series forecasting,” *Proceedings of the AAAI Conference on Artificial Intelligence,* vol. 35, no. 12, pp. 11106-11115, 2021.

[12] B. N. C. D. C. N. &. B. Y. Oreshkin, “N-BEATS: Neural basis expansion analysis for interpretable time series forecasting.,” *International Conference on Learning Representations. ,* 2019.

[13] Y. N. N. H. S. P. &. K. J. Nie, “A time series is worth 64 words: Long-term forecasting with transformers,” *International Conference on Learning Representations. ,* 2022.

[14] B. A. S. Ö. L. N. &. P. T. Lim, “Temporal fusion transformers for interpretable multi-horizon time series forecasting.,” *nternational Journal of Forecasting,* vol. 37, no. 4, pp. 1748-1764, 2021.

[15] H. Chase, “LangChain: Building applications with LLMs through composability,” 2022. [Online]. Available: https://github.com/hwchase17/langchain .

[16] P. P. E. P. A. P. F. K. V. G. N. .. &. K. D. Lewis, “Retrieval-augmented generation for knowledge-intensive NLP tasks.,” *Advances in Neural Information Processing Systems (NeurIPS conference proceedings),* vol. 33, pp. 9459-9474, 2020.

[17] H. M. L. S. K. A. P. A. A. B. Y. .. &. S. T. Touvron, “Llama 2: Open foundation and fine-tuned chat models.,” 2023.

[18] W. McKinney, “Data structures for statistical computing in Python.,” *Proceedings of the 9th Python in Science Conference (SciPy),* vol. 445, no. 51-56, 2010.

[19] Pandas Development Team, “pandas: powerful Python data analysis toolkit.,” 2023. [Online]. Available: https://pandas.pydata.org/docs/.

[20] W. McKinney, “Data wrangling with pandas, NumPy, and Jupyter.,” in *Python for data analysis:* , O'Reilly Media, 2022.

[21] J. (. VanderPlas, “ Essential tools for working with data.,” in *Python data science handbook*, O'Reilly Media, 2016.

[22] A. Géron, Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems., O'Reilly Media, 2019.

[23] S. &. M. V. Raschka, Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2., Packt Publishing, 2019.

[24] A. C. &. G. S. Müller, Introduction to machine learning with Python: A guide for data scientists., O'Reilly Media, 2016.

[25] F. V. G. G. A. M. V. T. B. G. O. .. &. D. E. Pedregosa, “ Scikit-learn: Machine learning in Python.,” *Journal of Machine Learning Research,* vol. 12, p. 2825–2830, 2011.

[26] T. T. R. &. F. J. Hastie, The elements of statistical learning: Data mining, inference, and prediction., Springer Science & Business Media, 2009.

[27] D. &. T. Y. (. Xu, “A comprehensive survey of clustering algorithms.,” *Annals of Data Science,* vol. 2, no. 2, p. 165–193, 2015.

[28] T. E. Oliphant, A guide to NumPy., Trelgol Publishing, 2007.

[29] J. &. B. Y. Bergstra, “Random search for hyper-parameter optimization.,” *Journal of Machine Learning Research,* vol. 13, no. 2, p. 281–305, 2012.

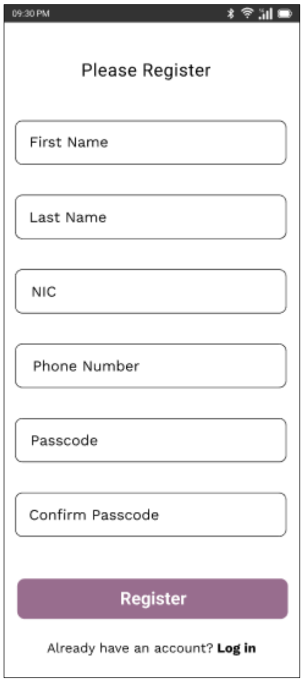
[30] T. S. S. Y. T. O. T. &. K. M. Akiba, “Optuna: A next-generation hyperparameter optimization framework.,” *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD),* p. 2623–2631, 2019.

[31] S. K. A. &. H. F. Falkner, “BOHB: Robust and efficient hyperparameter optimization at scale.,” *International Conference on Machine Learning (ICML),* p. 1437–1446, 2018.

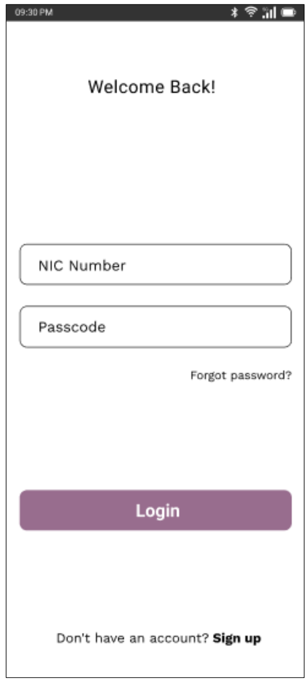
[32] J. Y. D. &. C. D. Bergstra, “Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures.,” *International Conference on Machine Learning (ICML),* p. 115–123, 2013.

Appendix A – Front end pages

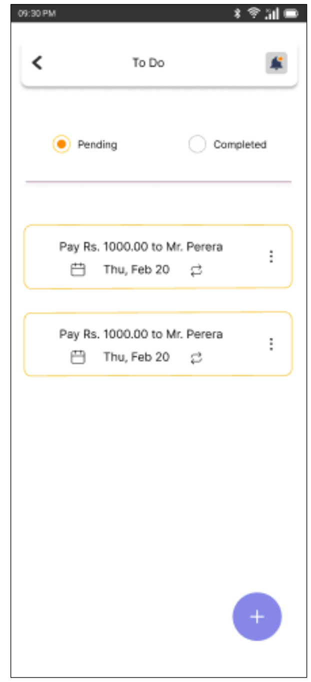
User registration:



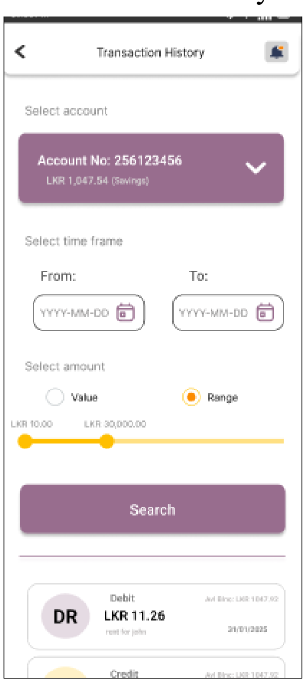
User login:



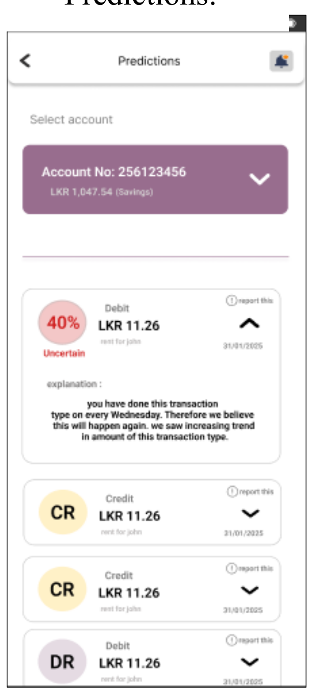
To-do list:



Transactions history:

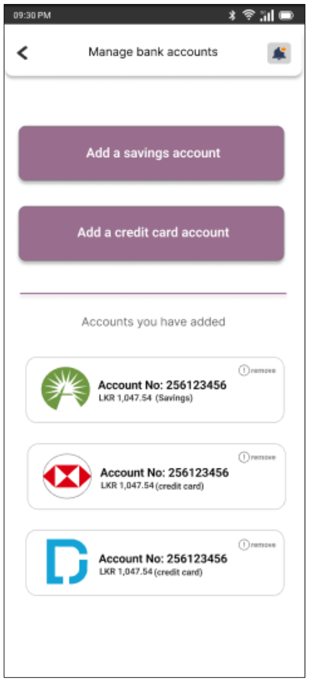


Predictions:



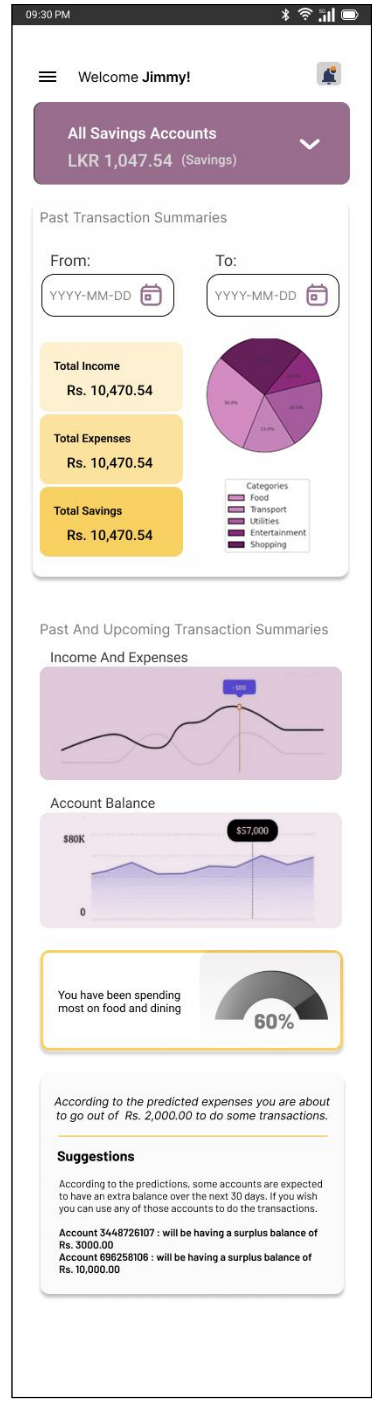
Notifications:

Account management

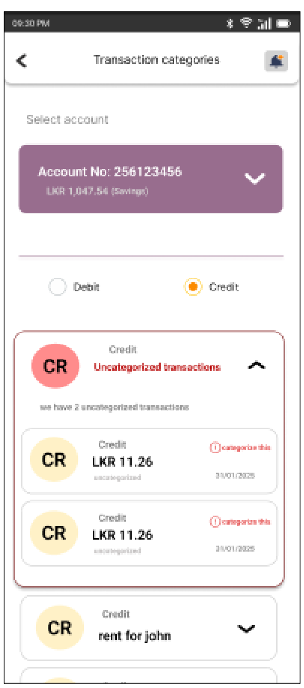


Chatbot:

Dashboard:



Transactions categories:



Settings:

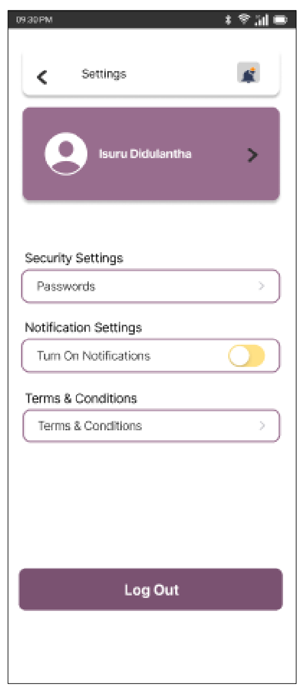


Figure Appendix A Frontend Pages

Appendix B – individual contribution

Individual contribution

N.D. Siriwardhana -225543D 1. Category wise expense prediction using TFT 2. Implement chatbot agent 3. Deploy local LLM 4. Backend development - User authentication, Bank account management, Transaction history view, OTP sending, View account balance prediction tab 5. Frontend development - Dashboard, To-do, Settings, Manage transaction categories tab, View account balance prediction tab	T.S.F Fernando -225515U 1. Total expense of a day prediction using N-BEATS 2. Integrate RAG application into the chatbot 3. Explanation generation for predictions 4. Backend development – Dashboard, To-do, Settings, Manage transaction categories tab, View account balance prediction tab 5. Frontend development - User authentication, Bank account management, Transaction history view, OTP sending, View account balance prediction tab
T.W.M.K.C Thilakarathna -225545K 1. Category wise income prediction using TFT 2. Data sanitization for privacy using Llama model 3. Category wise income prediction using N-BEATS 4. Backend development - User authentication, To-do, Manage transaction categories tab, Notifications, View prediction tab 5. Frontend development - Bank account management, Dashboard, Settings, Transaction history view, View prediction tab	P.K.I Didulantha -225510B 1. Total balance of a day prediction using N-BEATS 2. Transaction categorization 3. Category wise expense prediction using N-BEATS 4. Backend development - Bank account management, Dashboard, Settings, Transaction history view, View prediction tab 5. Frontend development - User authentication, To-do, Manage transaction categories tab, Notifications, View prediction tab

Figure Appendix B Individual Contribution