

Design Documents

Wallee - Personal Finance Mobile App

Team Members

Emma Bahr ebahr2022@my.fit.edu
Matteo Caruso mcaruso2023@my.fit.edu
Joshua Cajuste jcajuste2022@my.fit.edu
Kyle Gibson kgibson2021@my.fit.edu

Client & Faculty Advisor

Dr. Siddhartha Bhattacharyya sbhattacharyya@fit.edu
Doug Gibson

TABLE OF CONTENTS

1.0....	INTRODUCTION.....	
1.1....	Purpose.....	
1.2....	Scope.....	
1.3....	Overview.....	
1.4....	Definitions and Acronyms.....	
2.0....	System Overview.....	
3.0....	System Architecture.....	
3.1....	Architectural Design.....	
3.2....	Design Rationale.....	
4.0....	DATA DESIGN.....	
4.1....	Data Description.....	
5.0....	HUMAN INTERFACE DESIGN.....	
5.1....	Overview of User Interface.....	
5.2....	Screen Images.....	
5.3....	Screen Objects and Actions.....	
6.0....	Compent design.....	
7.0....	Appendices.....	

1.0 Introduction

- 1.1 Purpose

This software design document describes the early conceptualized internal architecture of the Walle app. Along with this, this document attempts to outline some of the early visual designs and features for the application.

- 1.2 Scope

The Software is intended to be a mobile application designed to help the user budget their finances to allow them to make bigger purchases easier. We intend for the app to be used by a wide range of users in order to better manage their spending.

- 1.3 Overview

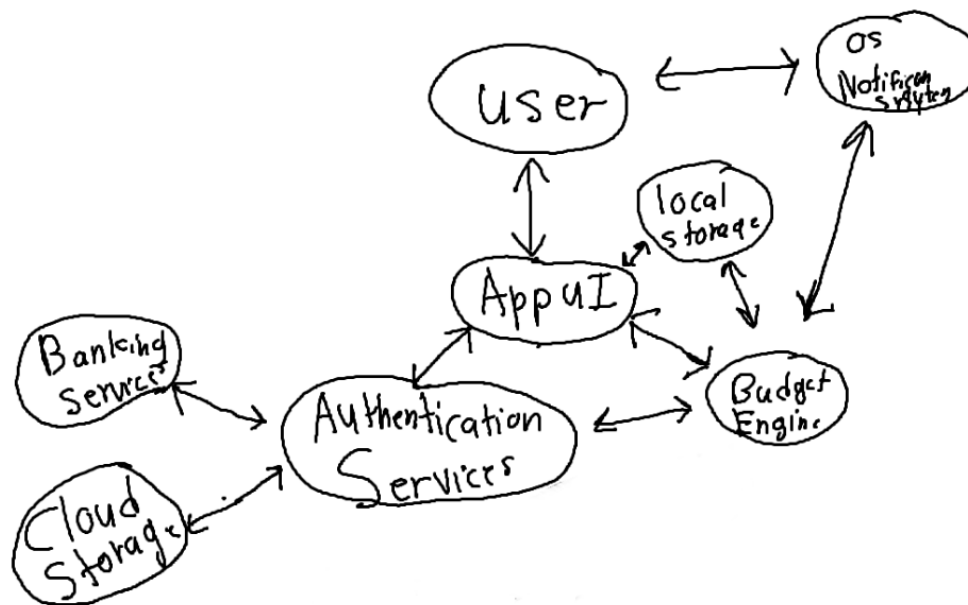
This documents describes provides a generalized overview of the underlying architecture of Walle, mostly focussing on our current concept for a user interface/interactable features

2.0 System Overview

Our application's main goal is to help improve a user's ability to manage their finances. We plan to achieve this by giving users access to features that allow them to set goals on the purchase they want to make. We also intend for the user to be able to link their bank account/s in order to monitor spending, track their income, and have a better visualization on how their spending and income affect their goal.

3.0 System Architecture

- 3.1 Architectural Design



The architecture describes our system as everything working together behind the UI.

- 3.2 Design Rationale

We believe that a user's account information should be protected at all times hence why we will be using cloud storage to store the account information and having it so that all interactions that involve user data will be forced to go through authentication services. The budget engine is relatively simple and should be light weight and fast enough to where it can live on the user phone. The engine can then periodically pull the user data through the authentication services and perform the calculations so that it is working with accurate information. This then ties into the notification system as the budget engine will then be able to send the user an alert they are closing or have approached the spending limit on a certain category. We also plan to store app settings like themes locally so that can always be in effect.

4.0 Data Design

- 4.1 Data Description

The data we store will be unique from user to user. Some of the information we store will include User profile information such as Name, email, phone and number. We will also intend to store the user bank account record which includes transaction history and balances. Almost all this data will be stored securely in the cloud.

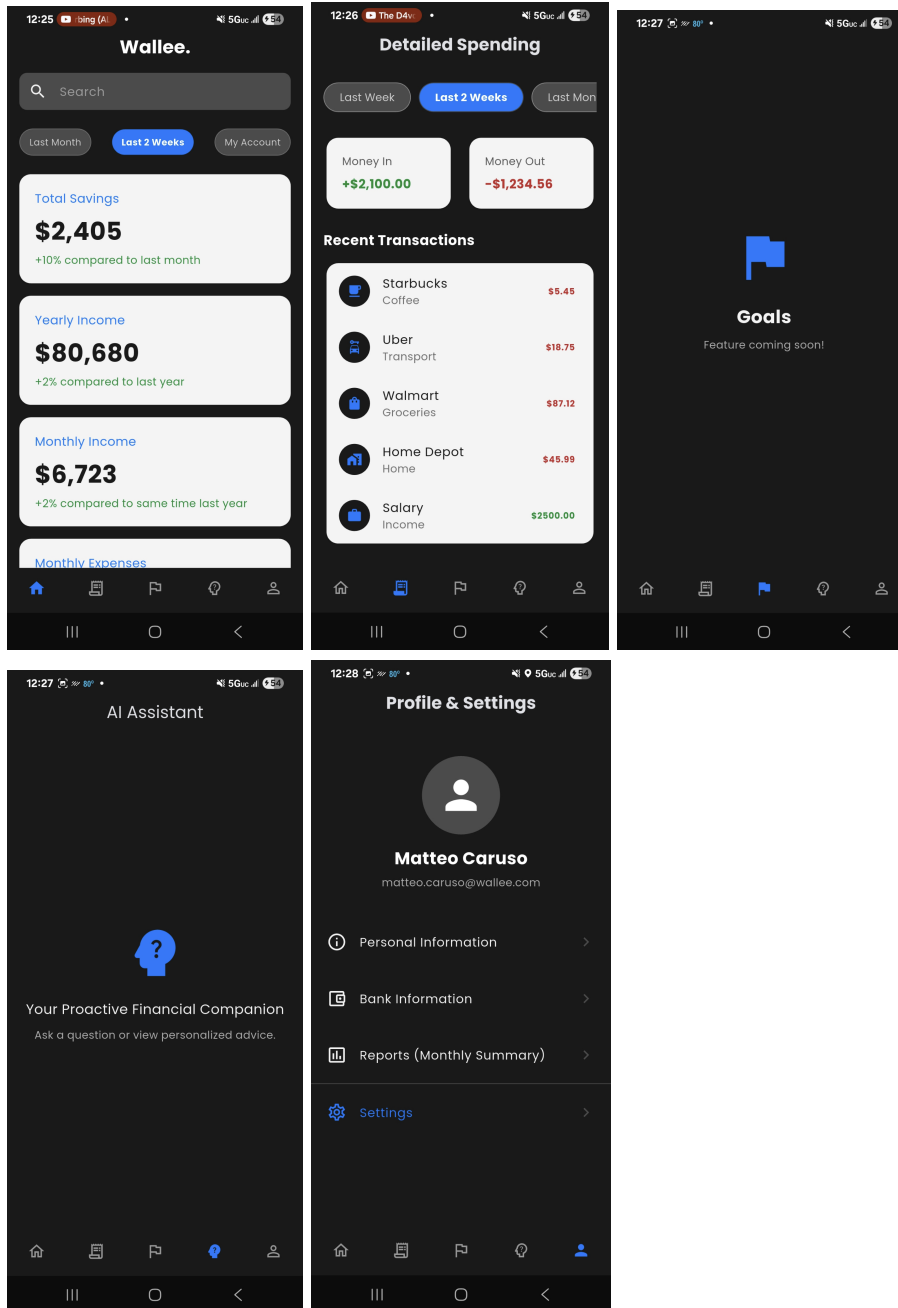
5.0 Human interface Design

- 5.1 Overview of User Interface

The Walle user interface has been designed with a focus on clarity and simplicity to help users manage their finances efficiently. The home dashboard provides a summary of total savings, income, and expenses, allowing users to quickly assess their financial health at a glance. Navigation is facilitated through a bottom tab bar, which provides access to the core features of the application, including the dashboard, detailed spending, goals, AI assistant, and profile settings. The main budgeting feature will be housed under the Goals tab, where users will be able to set financial targets and track their progress over time.

As we continue to develop our app, features will be refined and expanded upon, with new functionality layered on top of the existing prototype. The core of Walle centers around connecting accounts, reviewing financial activity, and comparing spending against budgets, while additional tools such as the AI assistant and detailed insights will extend and enhance the experience.

- 5.2 Screen images



- 5.3 Screen Objects and actions

The primary objects within the Walle interface are designed to make financial information both accessible and actionable. On the home dashboard, users will interact with summary cards that display total savings, yearly income, monthly income, and expenses. Each of these cards provides a quick snapshot of financial health, while the tab controls at the top allow users to filter results across different time periods.

In the detailed spending view, the main objects are transaction entries, which list the merchant, category, and amount. Each entry is color-coded to distinguish between income and expenses, and users may tap a transaction to view further details. This list serves as the record of financial activity, and filtering between “Money In” and “Money Out” helps users analyze their cash flow. The Goals tab allows users to set and monitor budgeting targets. While it is currently marked with a placeholder, it will be the main area for financial planning and budgeting within the application.

Additional interaction is available through the AI Assistant screen, which allows users to ask questions and receive personalized financial advice. This object is designed to enhance decision-making by providing context-driven suggestions. Finally, the profile and settings menu contains navigational objects for managing personal information, bank details, and financial reports.

6.0 Component Design

We currently have several components to the app in order for it to function properly. These are :

- The **Authentication Component** manages secure access to the application. It uses Firebase Authentication to support login and registration with email or third-party providers such as Google. This component also handles session management, ensuring that users remain securely logged in and that tokens are refreshed when necessary. By controlling identity verification, it serves as the foundation for all other modules.
- The **Account Linking Component** is responsible for securely connecting user bank accounts through Plaid. When a user initiates the account linking process, the component generates a Plaid link token, facilitates the linking flow, and stores account metadata. This enables Walle to securely fetch transaction data without exposing sensitive banking credentials.
- The **Transaction Management Component** handles the retrieval, storage, and display of user transactions. It communicates with the Plaid API to download transaction history, normalizes the data for consistent formatting, and stores it in Firestore. Within the app, this component powers the transaction feed, enabling users to browse their financial history, search by merchant, and filter results by date or category.
- The **Budgeting Component** allows users to create and monitor budget profiles across spending categories such as dining, entertainment, and groceries. It tracks the user’s progress against predefined limits, provides visual indicators of usage, and generates alerts when spending approaches or exceeds a set threshold. This ensures that users remain aware of their financial boundaries in real time.
- The **Insights and Analytics Component** transforms raw transaction data into meaningful financial intelligence. It generates spending summaries over different time periods, produces interactive visualizations such as pie charts and bar graphs, and highlights unusual spending activity. This component empowers users to understand patterns in their behavior and adjust their financial decisions accordingly.

- The **Notifications Component** ensures that important information reaches users at the right time. It delivers alerts when budgets are exceeded, sends reminders about financial goals, and can provide daily or weekly summaries of financial activity. By proactively engaging users, this component increases the app's usefulness beyond passive monitoring.

While we have these are components we currently have planned for app we may add other to accommodate other features that we plan for the app

7.0 Appendices

- API & Integration Details

Currently the Apis we have integrated are

- Nodejs
- Fast Api
- Nestjs
 - Also hosting Plaid
- Python Worker

We are working on implementing

- PostgreSQL
- Celery
- Redis
- Spacy

These are all integrated and hosted on google cloud project using cloud run or the apis that needed a container to run