BudgetMe

Software Design Document

Version 0.3

Contents

Section 1 - Project Description

1.1 Project
Budget Me

1.2 Description
Budget Me's mission is to help people build financial literacy. Budget Me is a mobile application that will provide the user a budget with little to no work on their part. Budget Me is geared towards a younger audience that is looking for guidance on how to budget their money.

1.3 Revision History

Date	Comment	Author
10/12/2023	Added the UML diagram	Omer Al Sumeri
10/12/2023	Added System Architecture	Jake Alvarado
11/10/2023	Added Data Tables / Collections	Jake Alvarado
11/10/2023	Updated Data Design and Domain Sections	Jake Alvarado
11/14/2023	Updated Project Purpose and Project Scope	Omer Al Sumeri
11/14/2023	Updated User Interface Design	Mohammad Skouti

Section 2 - Overview

2.1 Purpose

There is a knowledge gap about personal finance with people that are in high school and college which carries over to their adult life. The goal of Budget Me is to provide a way of learning on how to budget to a demographic that lacks knowledge in financial literacy

There are applications out there such as Mint that provide a budgeting service. Mint also crowds the user with other financial information that is geared toward someone who has more knowledge about financial space. The differentiator of Budget me is we will not crowd the application with other services that will ultimately get the user confused.

In theory, these topics may be great to introduce in a budgeting application, but in reality the user is bombarded with jargon and knowledge they don't understand. Budget Me believes in progress not perfection, and every step towards financial literacy is a milestone in of itself.

The mobile application Budget Me will provide the user with a budget after answering a questionnaire and then develop the user's understanding of budgeting and financial success by requiring them to enter their own spending.

2.2 Scope

Budget Me will be a mobile application whose main goal is to be the first step into financial literacy for young people. The benefits of Budget Me is to provide people that lack financial literacy a simple, easy to read budget. The main objective of Budget Me is to simplify the work for a user to create a budget. No complicated financial features or terms will be used.

2.3 Requirements

Your mileage may vary -- we typically break down the requirements to provide a ballpark estimate.

2.3.1 Estimates

#	Description	Hrs. Est.
1	Backend Database	20
2	UI/UX Design and Implementation	10
3	Frontend	20
	Total	50

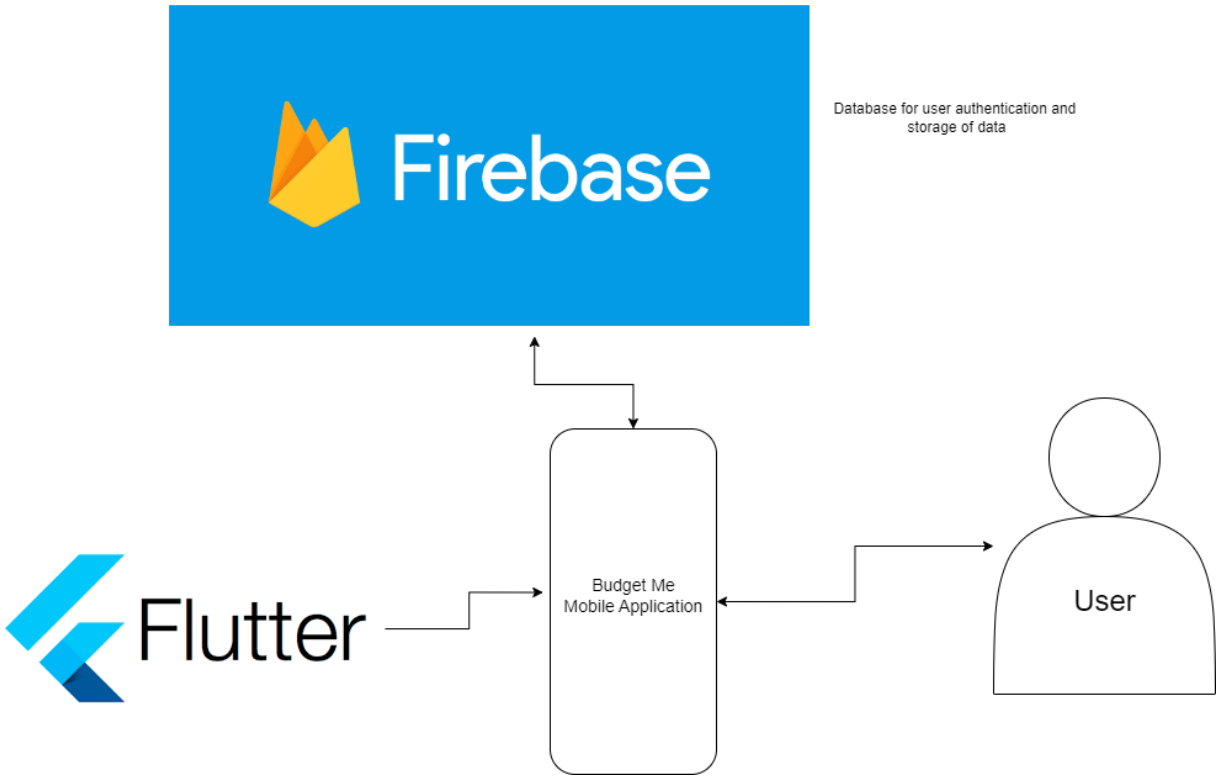
2.3.2 Traceability Matrix

Cross reference this document with your requirements document and link where you satisfy each requirement

SRS Requirement	SDD Module
Database for User account Creation	Section 3 and 4
Questionnaire intended to understand the users expenses and income	Section 7

Section 3 - System Architecture

Describe/include a figure of the overall system architecture (and where this module fits in)



System architecture includes the Flutter SDK managing back-end and front-end capabilities of the application. Using those packages and libraries to extend a minimal front-end user interface that allows user interaction to submit requests to an external Firebase. Communication from the users device to the firebase allows for features such as user authentication, real-time data storage of user information, and real-time visualization of such data on the application.

Section 4 - Data Dictionary

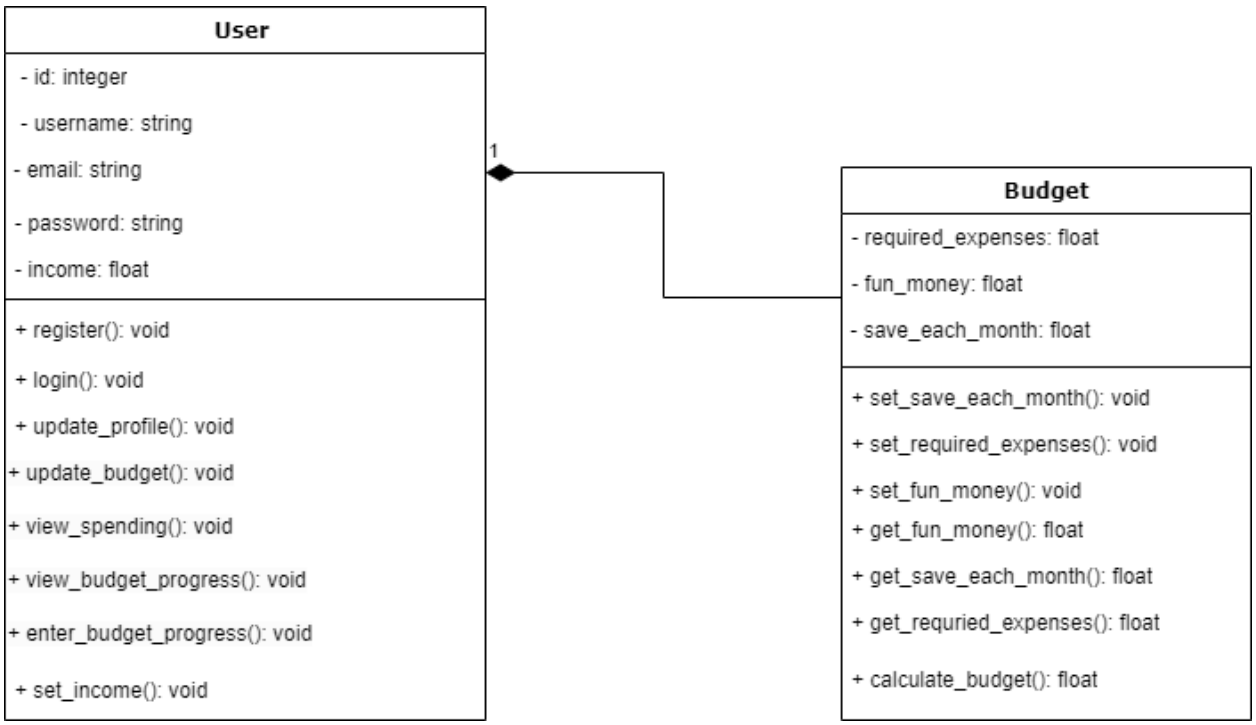
Brief description of each element in this module or a link to an actual data dictionary

Users		
Field	**Notes**	**Type**
ID	Unique Identifier from the Firebase Database	INTEGER
NAME	Username of the user's account	STRING
Email	Email of the user's account	STRING
Password	A variable of a pair along with NAME to allow access into the account	STRING
Income	Stores the users monthly income used in the budget creation	FLOAT

Budget		
Field	**Notes**	**Type**
required_expenses	Figure that will represent the total amount of required expenses a user has	FLOAT
fun_money	Figure that will represent the total amount of fun money allocated to the user	FLOAT
save_each_month	Figure that will represent the total amount that is set for the user to save each month	FLOAT

Section 5 - Software Domain Design

5.1 Software Application Domain Chart



5.2 Software Application Domain

A Comprehensive high level description of each domain (package/object wherever it is better to start) within the scope of this module (or within the greater scope of the project if applicable)

5.2.1 Domain X

Objects:

User class structured with Email, Username, Password.

- **FirebaseAuth**
- **Provides methods to the Firebase Authentication database in the form of methods:**
  - **CreateUserWithEmailPassword()**
  - **SignInWithEmailAndPassword()**

- **FirebaseStore**
  - **Provides methods to interact with collections that store the users information necessary for budget calculation**
    - **collections.doc.set('income': income, 'name' : name, 'expenses' : expense)**

5.2.1.1 Component Y of Domain X

User authentication is performed by the Firebase Authentication. The software won't use any email authorization links but will automatically register users and then perform a case match on the email and passwords.

Using the generated UID from the user authentication, we use those as a key to reference each user's specific collection. In this collection, the user's display name, income, expenses, and budget can be stored. This will allow for persistence throughout the life of the application

Section 6 – Data Design

Describe the data contained in databases and other shared structures between domains or within the scope of the overall project architecture

6.1 Persistent/Static Data

Describe/illustrate the logical data model or entity relationship diagrams for the persistent data (or static data if static)

6.1.1 Dataset

- Front-End login page credentials -> backend database authentication network -> collections of all expenses, user income etc.

6.1.2 Static Data

- Income is static, does not change throughout the life of the program unless the user specifically changes it.

6.1.3 Persisted data

- Budget is an integer that persists throughout the life of the application and will change as the user inputs their expenses.
    - Ideally this will persist even when the app closes and reopens.

6.2 Transient/Dynamic Data

- Expenses are dynamic, they change in accordance with the users transactions and overall shape the users spending.

6.3 External Interface Data

- External interfaces include any additional mobile devices, web viewing applications, and the firebase store. Firebase / Firestore collections and user authentication data tables are available as well.

6.4 Transformation of Data

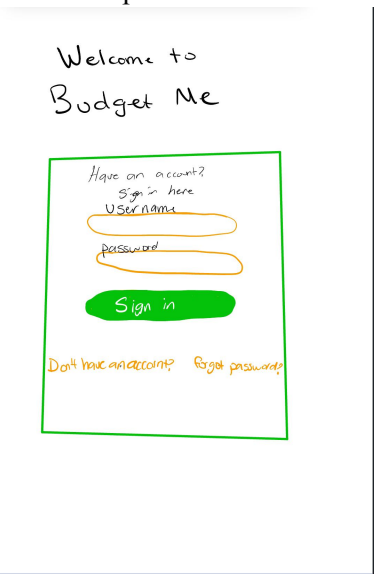
Describe any data transformation that goes on between design elements

- The application will transform the users budget, income, expenses into an easily digestible graph/chart.

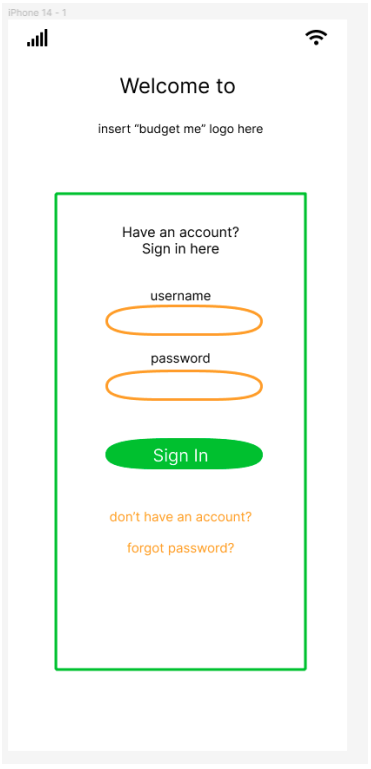
Section 7 - User Interface Design

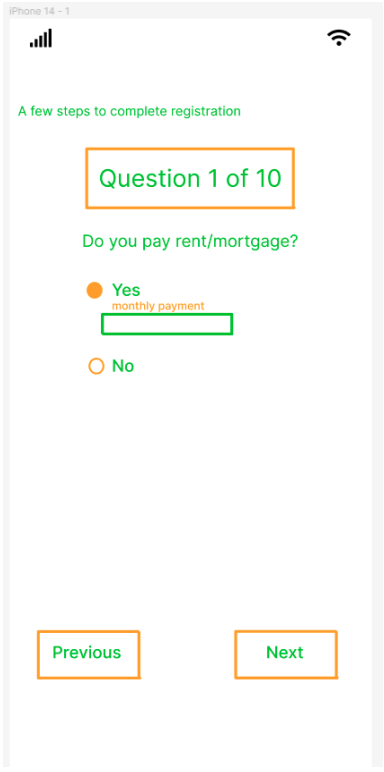
7.1 User Interface Design Overview

Pictures, high level requirements, mockups, etc.
Original mockups:

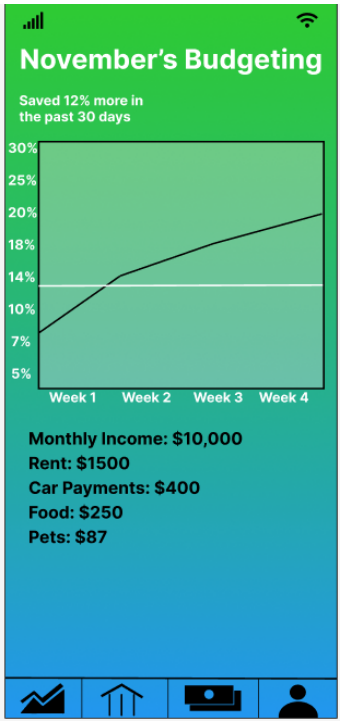


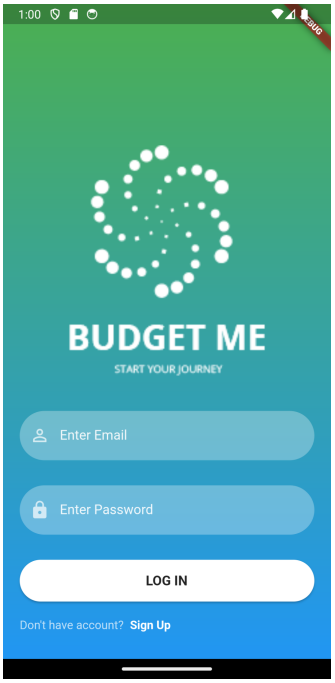
Figma mockups:

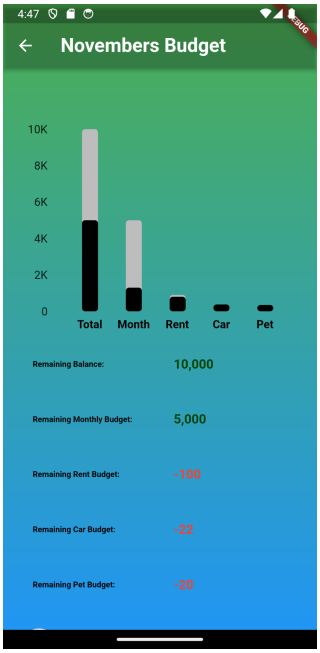


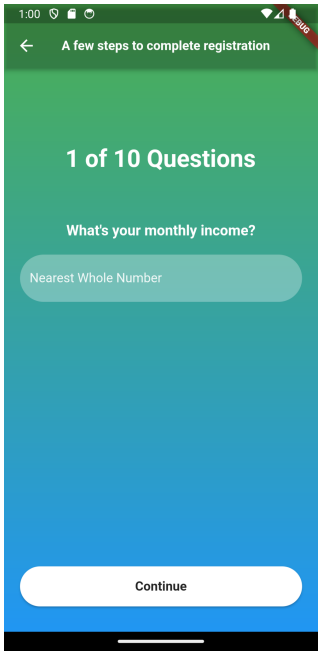


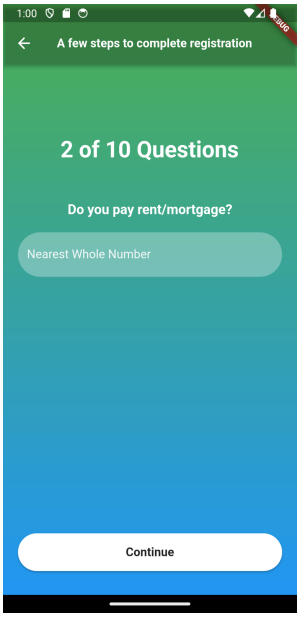
Frontend Implementations:



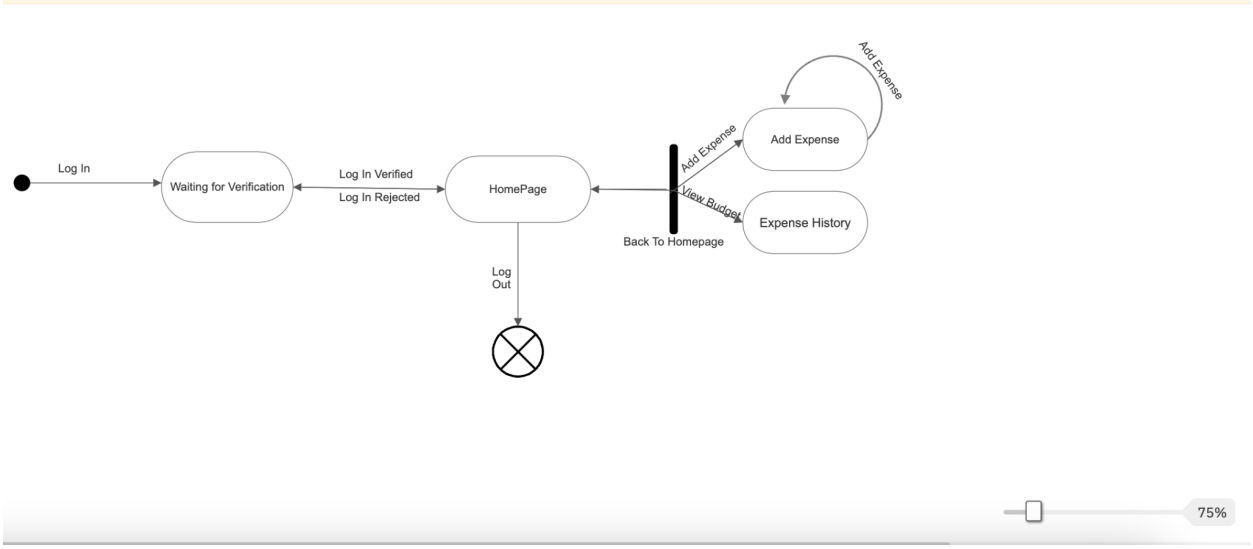


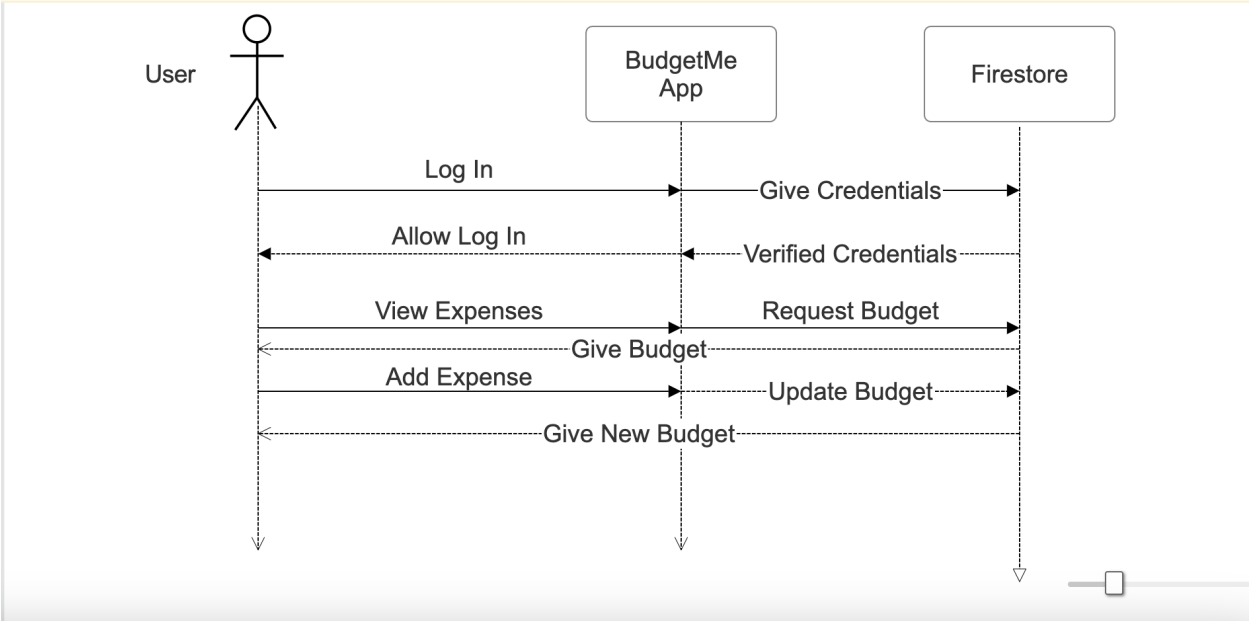






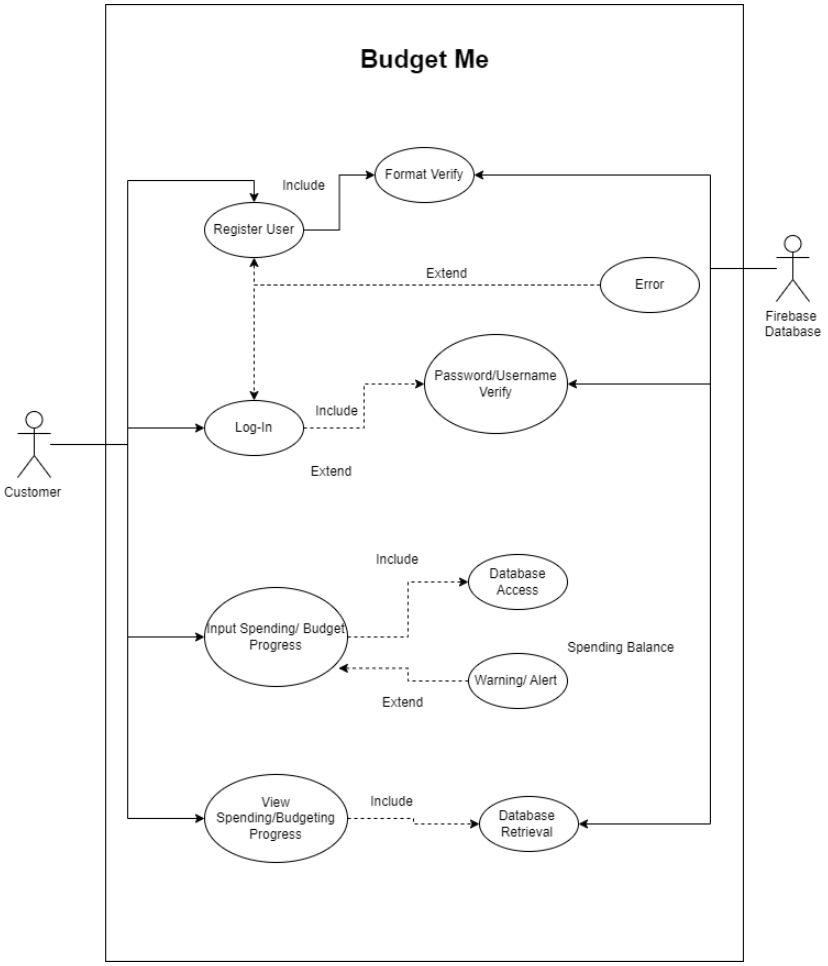
7.2 User Interface Navigation Flow





7.3 Use Cases / User Function Description

The sign in page will be used for the user's email and password, also has options in case they had forgotten their password or need to create an account. This page will not be used often, only for newcomers or users that need to sign in again.



Section 8 - Other Interfaces

8.1 Firebase Database Interface

The BudgetMe app utilizes the Firebase Database interface as the backend to store and retrieve user data. This interface is crucial for maintaining user profiles, budget information, and ensures the user a seamless experience. Also ensures that the user can view changes in real-time.

8.2 Detailed Overview

Interactions:

The interactions between the frontend and Firebase include CRUD operations(Create, Read, Update, Delete) through the integration of Flutter with Firebase Firestone.

Protocols:

Communication between the Flutter front end and Firebase backend relies on the following Flutter packages:

FirebaseAuth: Used for user authentication.
FirebaseAuthUI: Provides a ready-made UI for authentication.
FirebaseFirestore: Facilitates interactions with the Firestore database.

Message Formats:

Data exchanged between the Flutter frontend and Firebase Firestore conforms to the Firestore document structure, utilizing JSON-like data representation.

Failure Conditions:

Network connectivity is required to retrieve data from Firebase Firestone. Authentication failures can occur due to either invalid credentials or unauthorized access attempts.

Handshaking:

The FirebaseAuth package establishes a secure handshaking process to secure communication between Flutter and Firebase.

Section 9 - Extra Design Features / Outstanding Issues

9.1 Additional Design Features

9.1.1 Two-Factor Authentication

Explore the implementation of Two-Factor Authentication (2FA) to enhance user account security. Utilize the FirebaseAuth package to integrate 2FA seamlessly into the authentication process.

9.1.2 Bank API Integration for Seamless Budgeting

Integration with bank APIs to automatically import users' financial data for a more accurate budgeting experience. This requires extensive external research and collaboration with financial experts for an understanding of banking APIs and industry standards.

9.1.3 Customizable Notification System

Implementation of a customizable notification system to provide with personalized alerts and reminders related to their budget. Users can set preferences such as notification frequency, content, and delivery method.

9.1.4 Machine Learning for Budget Adherence Alerts

Integration of machine learning algorithms to analyze user spending patterns and recognize shifts from their budget. This will allow the user to get alerts when the system identifies overspending from the planned budget. For this, extensive research is needed in machine learning to develop and train an accurate model for budget predictions.

9.2 Outstanding Issues

9.2.1 Lack of Financial Expertise

The team has an absence of financial expertise. We need to seek external research and consultation to make informed decisions related to financial features. Collaboration with financial experts is needed as a guide to ensure the accuracy of the financial components of the app.

9.2.2 Privacy and Security Concerns

We anticipate potential privacy and security concerns associated with accessing and storing sensitive financial information. Implementation of encryption and data protection measures are needed to ensure the confidentiality and integrity of user financial data.