Shivam Missar

7052CEM Mobile Platforms and Application Development

Expense tracker

Table of Contents

[List Of Figures 2](#_Toc162265636)

[List Of Tables 3](#_Toc162265637)

[1 Introduction 4](#_Toc162265638)

[1.2 Aims and objectives 4](#_Toc162265639)

[2 Analysis 5](#_Toc162265640)

[2.1 Design patterns 5](#_Toc162265641)

[2.2 Requirements 5](#_Toc162265642)

[3 Design 7](#_Toc162265643)

[3.1 UML Use Case 7](#_Toc162265644)

[3.2 UML Entity Relational Database 8](#_Toc162265645)

[4 System Architecture 9](#_Toc162265646)

[4.1 Data involved 9](#_Toc162265647)

[4.2 System layers 10](#_Toc162265648)

[5 Application Design 12](#_Toc162265649)

[5.1 Main Menu 12](#_Toc162265650)

[5.2 Homepage 13](#_Toc162265651)

[5.3 Log Expense/View Expenses 13](#_Toc162265652)

[5.4 Manage Account 15](#_Toc162265653)

[References 16](#_Toc162265654)

# List Of Figures

[Figure 1.Use case diagram (Lucidchart, n.d.). 7](#_Toc162265628)

[Figure 2.ERD diagram for expense application (Lucidchart, n.d.) 9](#_Toc162265629)

[Figure 3. System Architecture (Lucidchart, n.d.). 11](#_Toc162265630)

[Figure 4.Main page for when the user first loads in. 12](#_Toc162265631)

[Figure 5.Homepage which summarises the user's spendings. 13](#_Toc162265632)

[Figure 6. Log expenses page. 14](#_Toc162265633)

[Figure 7. Shows a list of logged transactions in cardview. 14](#_Toc162265634)

[Figure 8.Manage account page. 15](#_Toc162265635)

# List Of Tables

[Table 1. Functionality table. 5](#_Toc162265622)

[Table 2. Details about the type of data involved. 9](#_Toc162265623)

# 1 Introduction

Many people in today’s world fail to track their expenses in an effective manner. According to Statista (2019), it states that 38% of users have never used an app to help manage their budget within the UK. As the world is increasing becoming digitalized, there needs to be more awareness of financial tracking and help people become more aware about their spendings.

This is where the proposal of an expense tracking application comes in allowing users to track their expenses. The application will be designed for users **between the ages of 18-60** and prioritizes being **user friendly, interactive and easy to use**.

## 1.2 Aims and objectives

However, to meet the aim of this project which is to **successfully** develop an application for managing expenses, the objective(s) of the application is to:

***OBJ1.*** Follow a **design pattern** which ensures right design choices are made creating a nice visual design

***OBJ2.*** Identify key data and sperate data into separate tables within the database

***OBJ3.***  Create **dynamic pages** which update regular based on user interactions

***OBJ4.*** Create a secure way for user authentication when **logging in** and **updating personal information**

# 2 Analysis

This section focuses on the design pattern which needs to be followed in order to get the most appropriate design thinking about all the users such as the sort of colours and layouts. This section also includes focusing on the sort of functionalities needed for the application to run.

## 2.1 Design patterns

To ensure the application is well designed, easy to use and user-friendly, it will require following a design pattern to ensure ‘***OBJ3****’* is met. The application will be following [Google’s design](https://m2.material.io/design) style to ensure all the boxes are ticked such as the use of colour, font, layout and vice versa. Following this design pattern ensures that the design language of this application is met to the highest standard.

## 2.2 Requirements

Seen in table 1 is a list of the functionalities which **aim** to be implemented within the application.

Table 1. Functionality table.

|  |  |  |
| --- | --- | --- |
| Requirement No | Functionality name | Description |
| R1 | Register | To use the application, user must be registered so that content can be tailored to them. The stored data will be securely stored with Google’s firebase authentication module. |
| R2 | Login | Login functionality is required to confirm the user’s identification and to meet this functionality, this will be done via Google’s authorisation inbuilt functionality to confirm the user’s identification. |
| R3 | Total spendings | The purpose of this feature is to display a summary of how much the user has spent in total which is calculated through the spendings that the user has logged. |
| R4 | Log expenses | To implement the functionality described in ‘R3’ it requires a method for the user to log each expense made which is the purpose of this functionally. The user will need to input their expense made which requires, the **amount**, the **date of purchase** and the type of purchase made.  **Once they have logged this payment, the user will not be able to remove this payment.** |
| R5 | View all expenses | This feature will be used to display all the expenses which the user has logged using a linear or list view containier. |
| R6 | Set a budget | The purpose of this feature is to allow the user to set a budget and keep their spendings in check.  If the user has set a budget, it will give the user constant reminders displayed on the homepage to inform them of their total spending and to see if the spendings is within their budget or has exceeded their budget. |
| R7 | Update password | User can update their password using Google’s firebase inbuilt functions to update their password. |
| R8 | Update currency type | This functionality allows the user to update their currency type in case of moving to a different country. |
| R9 | Delete account | If the user no longer wants to use the application, they can then delete their account which in result will remove all their data from the database. |

# 3 Design

## 3.1 UML Use Case

As seen in figure 1 it shows a use case diagram which represents the scope of what the user can do within the application.

It shows a visiting user who is only restricted to register as to access the in-app features, the application requires the user to register. Meanwhile, a user who is already registered would have access to all the in app features where they can carry out the tasks/interactions they need to do.

A diagram of a user flow

Description automatically generated

Figure 1.Use case diagram (Lucidchart, n.d.).

## 3.2 UML Entity Relational Database

Figure 2 shows the **entity relational database diagram** which shows the structure of the database. The choice of database used for the application is **Google’s Firebase/Authentication** and as seen in figure 2 the database will be made up of 3 different collections.

The ‘UserData’ collection stored in **Firebase** focuses on storing additional data for each user. Data such as email and password will be directly stored into the ‘User’ table which represents **Google’s** **Authentication** module as that is used for authentication hence, why it is stored separately. However, within ‘UserData’ collection it contains documents which are identified by the userId when registering the user. The ‘UserData’ table then directly links to the ‘Expense’ collection which has a relationship of ‘one to many’ as one **user can log one or more expenses**.

The ‘Expenses’ collection then focuses on data based on the expense which has been logged by that particular user hence, why userId is needed within this collection. It stores the amount, date the expense was logged to be later used as a filter to display the most recent transaction on the summary page, the item type i.e., the purchase title. This is then later used to display a list of the expenses made by that particular user and even used for calculating the user’s total spendings displayed on the summary page.

The ‘Expense’ table then links with the ‘ItemType’ collection that has a relationship of ‘one to one’ as **one** **expense only has one item type**. The ‘ItemType’ collection contains a list of documents which will store names of different type purchase options such as if the user has spent money on a music subscription it will offer the user that option. This table is mainly used to categorise expenses leading to a more structured approach.

A diagram of a server

Description automatically generated

Figure 2.ERD diagram for expense application (Lucidchart, n.d.)

# 4 System Architecture

This section introduces the sort of data which is being handled and how it is involved within the application. Furthermore, this section also highlights the system architecture i.e., how the data is processed and resulted back to the user display.

## 4.1 Data involved

Table 2 highlights each type of data which has a direct corelation within the application.

Table 2. Details about the type of data involved.

|  |  |
| --- | --- |
| Data type | How it contributes to the application |
| User registration and sign in | This data contributes to the application in terms of registering the user and authenticating them to gain access to within the application. This would include a mix of data such as email, password, currency type and vice versa. This data is quite important to the application as it is needed in various of the application such as the manage account page. |
| Cloud data | Cloud data plays an important role within this application enabling data to be stored, retrieved and manipulated. The application makes effective use of this data for example, retrieving and displaying expenses made from Firebase, making use of Google’s authentication module, checking that the user exists and having other involvement in various parts of the application. |
| Expense data | Expense data is a fundamental part of the application as it used in various aspects within the application such as displaying the user’s total spendings, transaction history comparing budgets with total spendings and much more. |
| Item type data | This includes pre-defined data from within a created collection stored in firebase that categorizes the expenses each time a expense is logged. |
| UI data | This data includes data such as containers for displaying content such as linear and relative layouts, textviews, edittexts, buttons and so forth. |

## A close-up of a text Description automatically generated4.2 System layers

A diagram of a software application

Description automatically generated

Figure 3. System Architecture (Lucidchart, n.d.).

# 5 Application Design

The figures below show a few screenshots of the application and the outcome of how the application has turned out to look. It highlights the main functionalities on the pages.

## 5.1 Main Menu

Figure 4 shows the main menu of when the user first loads in. It uses a simple design using clear buttons labelled for the user to know which to click. It gives the option to register if they have not already done so or login if they have already registered with a link to reset their password if they have forgotten their password which sends a reset password link to the user’s inbox.

A screen shot of a phone

Description automatically generated

Figure 4.Main page for when the user first loads in.

## 5.2 Homepage

Figure 6 highlights the homepage i.e., a summary how much the user has spent, their most recent purchase and if they over bounds in terms of their budget. This page is very dynamic and will update based on the total spendings and any recent expense that has been logged. Both recent purchases and budget view are displayed through the help linear layout containers for better structure.

A screenshot of a phone

Description automatically generated

Figure 5.Homepage which summarises the user's spendings.

## A white background with black text Description automatically generated5.3 Log Expense/View Expenses

A screenshot of a phone

Description automatically generated

Figure 6. Log expenses page.

A black rectangular device with a white screen

Description automatically generated

Figure 7. Shows a list of logged transactions in cardview.

## A white background with black text Description automatically generated5.4 Manage Account

A screenshot of a phone

Description automatically generated

Figure 8.Manage account page.

# References

Material Design. (n.d.). *Design.* [*https://m2.material.io/design*](https://m2.material.io/design)

Lucidchart.(n.d.). *Documents.* [*https://lucid.app/documents#/documents?folder\_id=recent*](https://lucid.app/documents#/documents?folder_id=recent)

Statista. (2022). *Frequency of budget* apps *used by individuals in the United Kingdom*. <https://www.statista.com/statistics/1039358/frequency-of-budgets-apps-used-by-individuals-in-the-united-kingdom/>