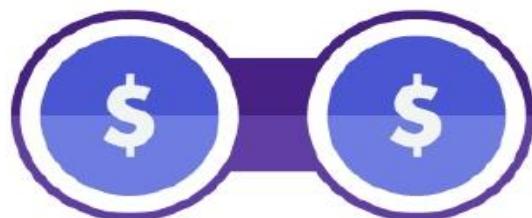




Cairo University
Faculty of Computers and Artificial Intelligence
Department of Computer Sciences



MoneyWatcher

Smart Financial Tracker

Supervised By

*Dr. Manar EL-Kady
TA. Basmaa Mokhtar*

Implemented By

20190537	Mostafa Abdelnasser Mohamed
20190448	Muhammad Salah Abdallah
20190451	Mohamed Adel Sayed
20190305	Abdalla Fadl Shehata

*Graduation Project
Academic Year 2022-2023
Final Documentation*

Table of Contents:

Chapter 1: Introduction	5
Motivation (Abstract)	5
Background	6
Problem Definition	6
Our Solution	7
Family Expenses Prediction Model	8
Overview	8
Objective	8
Data	9
Filipino Income and Expenditures Data	9
Data Preprocessing	10
Feature Selection	10
Correlation Matrix	11
Normalize and Splitting the Data	12
Model and Experiments	13
Deployment	15
Gantt Chart	17
Project Development Methodology	18
Waterfall	18
The Used Tools	19
Frontend	19
Backend	19
Machine Learning	19
Report Organization	20
Chapter 2: Related Work	21
Existing Similar targeted apps	21
1Money	21
Money Manager	21
Wallet	21
El-Masareef	22
Difference in our application	22

Chapter 3: System Analysis	23
Project Specification	23
System Architecture	23
Stakeholders	23
Functional Requirements	24
Non-functional Requirements	26
Use Case Diagrams	27
Use Case Tables	28
System Components Diagram	31
System Class Diagrams	32
Sequence Diagrams	35
Sign UP	35
Login	35
Add Budget	36
Add Expense	36
Export Report	37
Show Statistics	37
Project ERD	38
System GUI Design	39
Chapter 5: Implementation and Testing	41
Implementation	41
Frontend	41
Backend	41
Machine Learning	41
Testing	42
Conclusion	46
References	47

List of Tables

Table 1: Use Case 1 --> Sign Up	28
Table 2: Use Case 2 --> Login	28
Table 3: Use Case 5 --> Add Expense	29
Table 4: Use Case 6 --> Add Installment	29
Table 5: Use Case 7 --> View Statistics	30
Table 6: Use Case 9 --> Predict Expenses	30

List of Figures

Figure 1: Average Expenses on different categories of Egyptians, 2017/2018	9
Figure 2: Correlation Matrix	12
Figure 3: Comparison of Algorithm Performance	14
Figure 4: example of Prediction Model Flask API	15
Figure 5: Example of Using the Prediction model from the real App	16
Figure 6: Analysis Gantt Chart	17
Figure 7: Implementation Gantt Chart	17
Figure 8: System Architecture	23
Figure 9: Use Case Diagram	27
Figure 10: Component Diagram	31
Figure 11: Class Diagram 1	32
Figure 12 : Class Diagram 3	33
Figure 13: Class Diagram 2	33
Figure 14: Class Diagram 4	34
Figure 15: Sequence 1 (Sign Up)	35
Figure 16: Sequence 2 (Login)	35
Figure 17: Sequence 3 (Add Budget)	36
Figure 18: Sequence 4 (Add Expense)	36
Figure 19: Sequence 5 (Export Report)	37
Figure 20: Sequence 6 (View Statistics)	37
Figure 21: System ERD	38

Chapter 1: Introduction

Motivation (Abstract)

The goal of our graduation project is to develop a mobile app that helps a user manage his money, expenses, bank account, and monthly salary.

The basic idea of our application **MoneyWatcher** is to track finance and know where your money goes without needing any special skills and knowledge of organizing money management, we will provide the plans for you, just with a few buttons pushes you will able to track your spending, budgets, and debts and save your money as much as possible.

It's a finance manager, that will be developed to help to save your money, plan for the future, and see all your payments in one location.

The main goal of this project is to help you by managing personal finances smartly and easily, by recording your personal and business financial transactions, generating spending reports and reviewing your daily, weekly, and monthly financial data, and managing your assets with MoneyWatcher.

The project will introduce an easy way of expense tracking and introduce economic plans for the user to achieve the best way to spend his money. The people behind the project are very enthusiastic and eager to learn more about economics and money management to introduce the best experience to the user under the supervision of a more experienced person, **Dr. Manar**, and we reviewed some applications that provide the same idea.

The Technologies that will be used in our project are:

- Desktop Application (JavaFX)
- Mobile Application (Flutter)
- Back-end (Java Spring)
- Database (MySQL)
- Machine Learning Model (Python)

Background

Most of us have difficulty managing monthly expenses and the budget. We may spend a large amount that exceeds our income, and then wonder where this income went?

We will provide a smart solution, which is a mobile application that manages and organizes your monthly income and bank balance through careful monitoring of what you spend and what you pay for your monthly and even weekly bills, and then you get a report showing the value of what you spent, as well as reminding you of the dates of paying your monthly bills in order to avoid drowning in accumulated debts.

So, your best decision is to include **MoneyWatcher** on your smartphone to help you organize your expenses better, and we will help you to enjoy many features that we will discuss in the functional requirements part.

Problem Definition

Money, being the primary objective for individuals, institutions, and states, necessitates considerable effort and patience to acquire. However, effectively managing, spending, and tracking money in alignment with one's social standing requires further dedication and a sound understanding of money management systems. In this document, we present a proposed solution that leverages current technology to assist individuals in monitoring their finances and developing plans to optimize wealth preservation.

Many individuals aspire to reduce expenses, save more money, and establish a budget to avoid the distress of running out of funds before the month's end. Our application aims to address this issue comprehensively. By utilizing our app, individuals can significantly enhance their financial management capabilities, gaining the ability to track their expenditures and develop strategies for optimal financial well-being.

Our Solution

MoneyWatcher is a comprehensive financial management application designed to address the challenges individuals face in monitoring and controlling their finances.

With MoneyWatcher, users can effortlessly track their expenses, create personalized budgets, and implement effective strategies to optimize their money management. By providing a user-friendly interface and utilizing advanced technology, our solution empowers individuals to save more, make informed financial decisions, and achieve their financial goals with ease.

The MoneyWatcher application will be accessible across multiple platforms, including mobile devices and desktop computers. Whether you prefer to manage your finances on the go or from the comfort of your home or office, our application offers seamless compatibility with both mobile and desktop environments. This ensures that you can access and monitor your financial data anytime, anywhere, providing convenience and flexibility in managing your money effectively.

Family Expenses Prediction Model

Overview

The Total Expenses Prediction Model is a key component of the Smart Money Watcher. This model aims to forecast the total expenses based on a set of selected features and characteristics of the user's household. By leveraging machine learning techniques, specifically regression modeling, the model provides users with valuable insights into their expected expenses, enabling them to make informed financial decisions.

In this section, we will provide an overview of the Total Expenses Prediction Model, highlighting its purpose, objectives, and significance within the Smart Money Watcher. We will talk about the main aspects of building our Prediction model like: data preparation, feature selection, model training, evaluation metrics, and integration aspects of the model.

Objective

Based on the whole objective of the App, one of the primary key points in the App by helping the user with valuable insights about his financial situation and make informed financial Decisions in budgeting and goals, leads us to the need to provide powerful tool for budgeting and expense management that predict the expenses of the month.

this prediction is very valuable for the financial life of the user in several aspects

- Financial Planning and Threshold: By having a prediction of his total expenses, user can plan his finances more effectively. Putting the predicted expense as a threshold that he can plan according to it, user can allocate his resources wisely and set realistic budgets.
- Goal Setting and Monitoring: by knowing the predicted Expenses, user can align his savings targets, user can plan to save part of the predicted expense to achieve his financial objective, also user can track his progress, and makes informed decisions to stay on track towards achieving his goals.
- Proactive Decision Making: By Providing The model providing Future Insights into user's excepted expenses which is significant for any Financial Decision.

Data

After Heavy Search to find an Egyptian Data we found that "The National Agency for Public Mobilization and Statistics" not providing raw data (.csv file) to manage us build our intended model instead it provides only statistics, like the below chart:

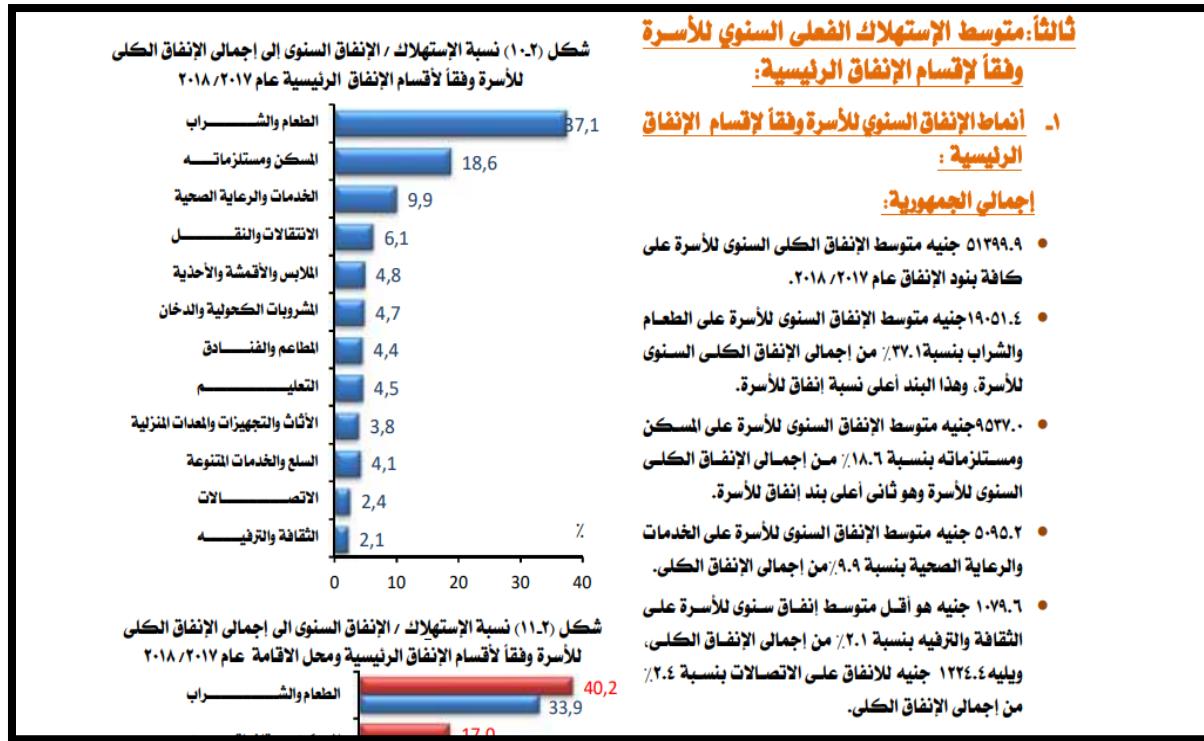


Figure 1: Average Expenses on different categories of Egyptians, 2017/2018

Filipino Income and Expenditures Data

Happily, we found “Philippine Statistics Authority” runs a survey called the Family Income and Expenditure Survey (FIES) every three years to gather information family income and expenditure, including, among others, levels of consumption by item of expenditure, sources of income in cash, and related information affecting income and expenditure levels and patterns in the Philippines,

The dataset, composed of (41,544) rows and (60) features for each household, including significant columns which matched our prediction model’s needs like: “Total Household Income”, “Main Source of Income”, “Total Income from Entrepreneurial Activities”, “Household Head Sex”, “Household Head Age”,

“Household Head Marital Status”, “Household Head Highest Grade Completed”
“Family Members” “Family Members Employed”, “number of phones”

Data Preprocessing

Convert the annual values to monthly values: as the model aims to predict the total expenses of the month not the year, we did it by dividing on 12.

Finance Conversion from Philippine’s Peso to Egyptian pound: just for convenient and readability only, we did this conversion by multiplying by 0.55, as the 1 peso = 0.55 EP.

As it will be valuable for future work to unify Expenses Categories to be the common categorization of expenses among surveys of different countries (like Egypt), we did the following:

Combined the two columns “Housing and Water Exp” + “Imputed House Rental Value” in one columns called “Housing Exp” to concern about the total Housing Exp in only one column “Tobacco and Alcoholic Exp”

Combined the two columns “Tobacco Exp” + “Alcohol Exp” to be

Converted the name of the columns “Special Occasions Exp” to be “Entertainment and Other Exp”, as the features aren’t including entertainment field, and the word other enable the user to put in this field the unmentioned category Exp if exist

Created a new column “total expenses” which is the sum of all expenses = Food Exp + Housing Exp + Medical Care Exp + Transportation Exp + Clothing and Wear Exp + Tobacco and Alcoholic Exp + Restaurant and Hotels Exp + Education Exp + Miscellaneous Goods and Services Exp + Communication Exp + Crop Farming and Gardening Exp + Entertainment and Other Exp.

Feature Selection

The target feature is the Total Expenses

The independents will be as the follows:

Dealing with 60 features, forced us to make feature selections to determine which features will affect our prediction, with (46) features as input (independents) is not reasonable and affects user experience which needs to enter heavy input fields and data, we will do 2 experiences, one with the 46 features and the second with the choosing suitable number of features as an input, and we will watch the small difference.

Feature number	Feature name
1	'monthly_income'
2	'Total_Income_from_Entrepreneurial_Acitivites'
3	'Household_Head_Job_or_Business_Indicator'
4	'Total_Number_of_Family_members'
5	'Main_Source_of_Income',
6	'Household_Head_Age'
7	'Number_of_Cellular_Phone'
8	'Household_Head_Marital_Status'
9	'Type_of_Household'
10	'Household_Head_Sex'
11	'Total_Number_of_Family Members Employed'
12	'Members_with_age_less_than_5_year_old'
13	'Members_with_age_from_5_To_17_years_old'
14	'Agricultural_Household_indicator'
15	'Household_Head_Class_of_Worker'
16	'Household_Head_Occupation'

First we can determine with experience the persuading features that offcourse affects the expenses of any household, the following are 16 features that we will filter them in the next stage, these features listed below:

Feature 'Agricultural_Household_indicator': Indicate which if the household has Agricultural activities that affects the total income of the household, it variates from 0(high activity), 1 (moderate activity), 2 (low activity)

Correlation Matrix

To check the importance of the features, we have to check the correlation values of each feature with the target feature.

Encoding the Categorical Features with Label Encoder: we need first to encode the categorical features with Label encoder which give each value in the column a unique numerical value, this will manage us to draw the correlation matrix for it and for fitting the model, below is the correlation matrix:

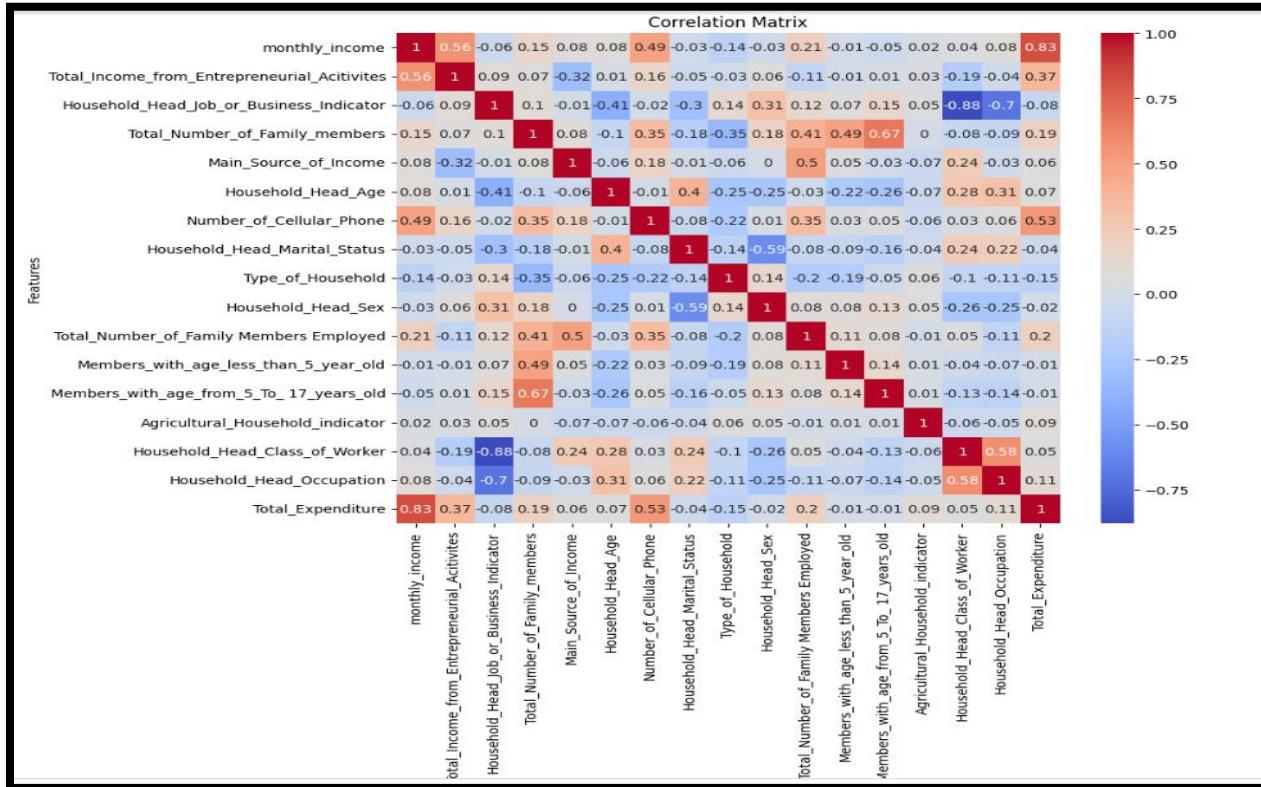


Figure 2: Correlation Matrix

From the correlation matrix, we can see that Total Expenditure has some reasonable correlation value with 6 features which are:

- "monthly_income"
- "Total_Income_from_Entrepreneurial_Activities"
- "Agricultural_Household_indicator",
- "Total_Number_of_Family_members"
- "Total_Number_of_Family_Members_Employed"
- "Number_of_Cellular_Phone"

Normalize and Splitting the Data

We used **Min Max Scalar** Algorithm to normalize the data with the equation:

$$X = (X - \text{minValue}) / (\text{maxValue} - \text{minValue})$$

Splitting the Data to xTrain, xTest, yTrain, yTest with 75% of the data for training and 25% of the data for Testing.

Model and Experiments

Our Task is Obvious that it is a Supervised Regression Problem so we need to apply and compare different and suitable Algorithms for getting the best result

while we Train our models with our ready data, we will use two important techniques for helping getting better insights and improving the result:
cross-validation: by train the models in 5 different splitting of the data
hyper parameter tuning: by setting for each hyper parameters list of values and fit and test the model and get the best combination of hyper parameters

Experiments and Evaluate Different Algorithms:

Let us try each algorithm and test it's score with the R2 evaluation:

Linear Regression:

We conducted an experiment using Linear Regression to predict total expenses.

Here are the results obtained:

- Train Score: 0.7184075450654188
- Test Score: 0.7459760248446135
- Cross-Validation Scores: [0.43590468, 0.73765008, 0.75460475, 0.72977389, 0.74232499]

Gradient Boosting:

We also experimented with Gradient Boosting for total expense prediction. The following are the results:

- Train Score: 0.8783771523436342
- Test Score: 0.7804163394482927
- Cross-Validation Scores: [0.75787768, 0.78540468, 0.82808379, 0.78068901, 0.74620636]

XGBoosting:

Another algorithm we employed is XGBoosting. Here are the results:

- Train Score: 0.8703387778909394
- Test Score: 0.7792336403300275
- Cross-Validation Scores: [0.76937294, 0.78677582, 0.8196059, 0.77618916, 0.72668308]

Random Forest Regressor:

Additionally, we experimented with the Random Forest Regressor algorithm for total expense prediction. Here are the results:

- Train Score: 0.9440054731137455
- Test Score: 0.7637319095749504

- Cross-Validation Scores: [0.76664361, 0.77617444, 0.80573696, 0.78263124, 0.7183109]

The train and test scores represent the performance using the R2 metric, a higher score indicates a better fit to the data. The cross-validation scores provide an estimate of the model's generalization performance across different folds of the data.

According to the above results we can summary them in the below chart:

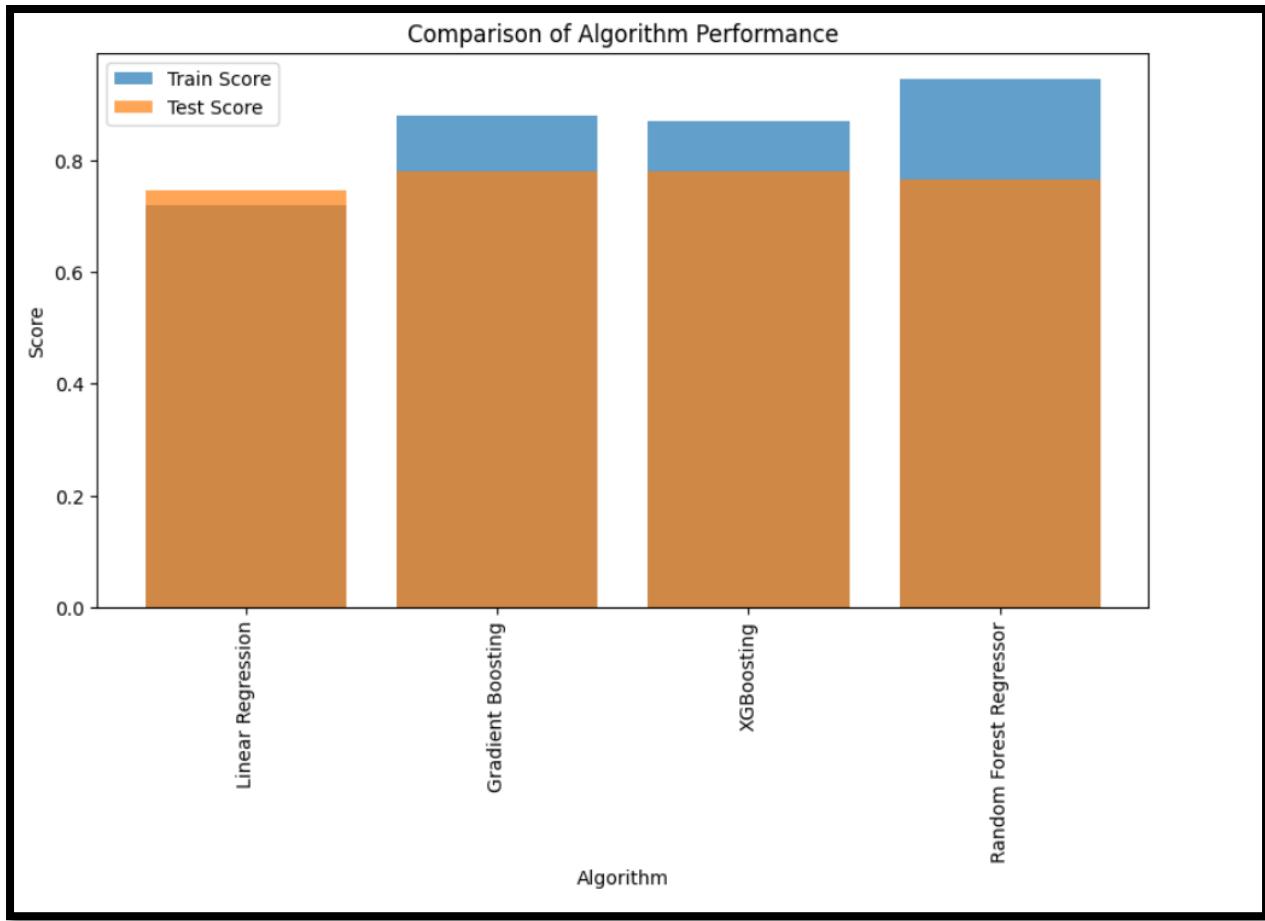


Figure 3: Comparison of Algorithm Performance

We can see that the best test scores R^2 is coming from the Gradient Boosting Algorithm with $R^2 = 0.78$, so now we are ready with the Prediction Model

Deployment

To enable other parts of the App to user the model we need to provide an API which takes the input (features) and provide the output (the total expenses)

Flask API that takes the 6 feature, the API form:

(<http://127.0.0.1:5000/predict/v1/v2/v3/v4/v5/v6>)

Where:

V1 = "monthly_income"

V2 = "Total_Income_from_Entrepreneurial_Acitivites"

V3 = "Agricultural_Household_indicator"

V4 = "Total_Number_of_Family_members"

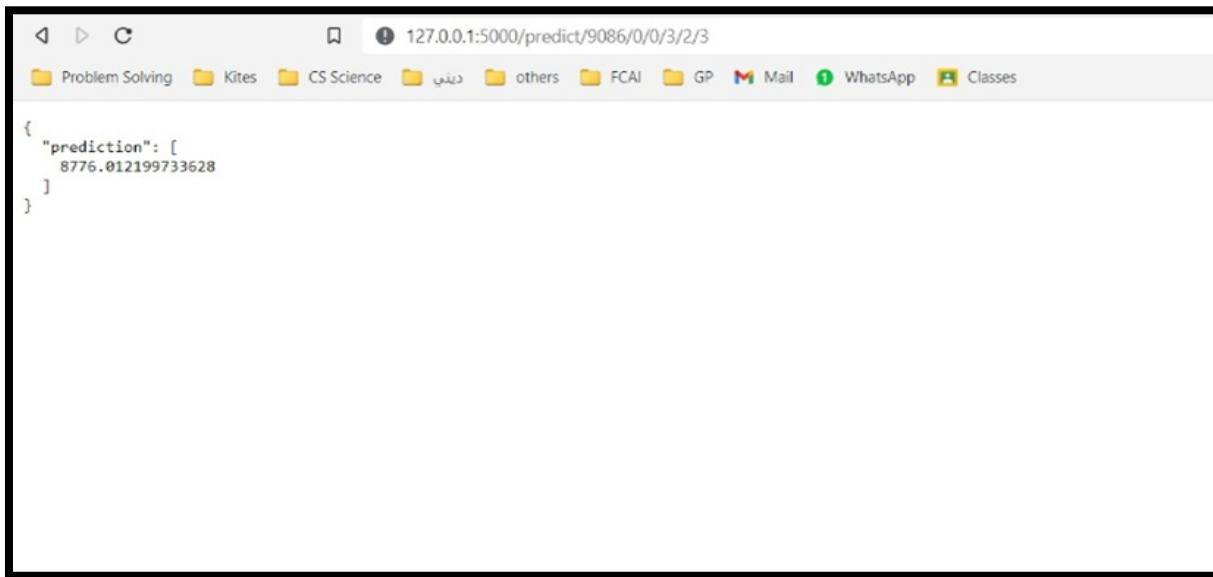
V5 = "Total_Number_of_Family Members Employed"

V6 = "Number_of_Cellular_Phone"

taking the features values from the URL and normalize them with the same algorithm we used before fitting and then predict the total expenses, here is an example of the API calling

the actual total expenses = 9712

the predicted total expense = 8776



The screenshot shows a browser window with the URL `127.0.0.1:5000/predict/9086/0/0/3/2/3`. The page content displays a JSON response:

```
{ "prediction": [ 8776.012199733628 ] }
```

Figure 4: example of Prediction Model Flask API

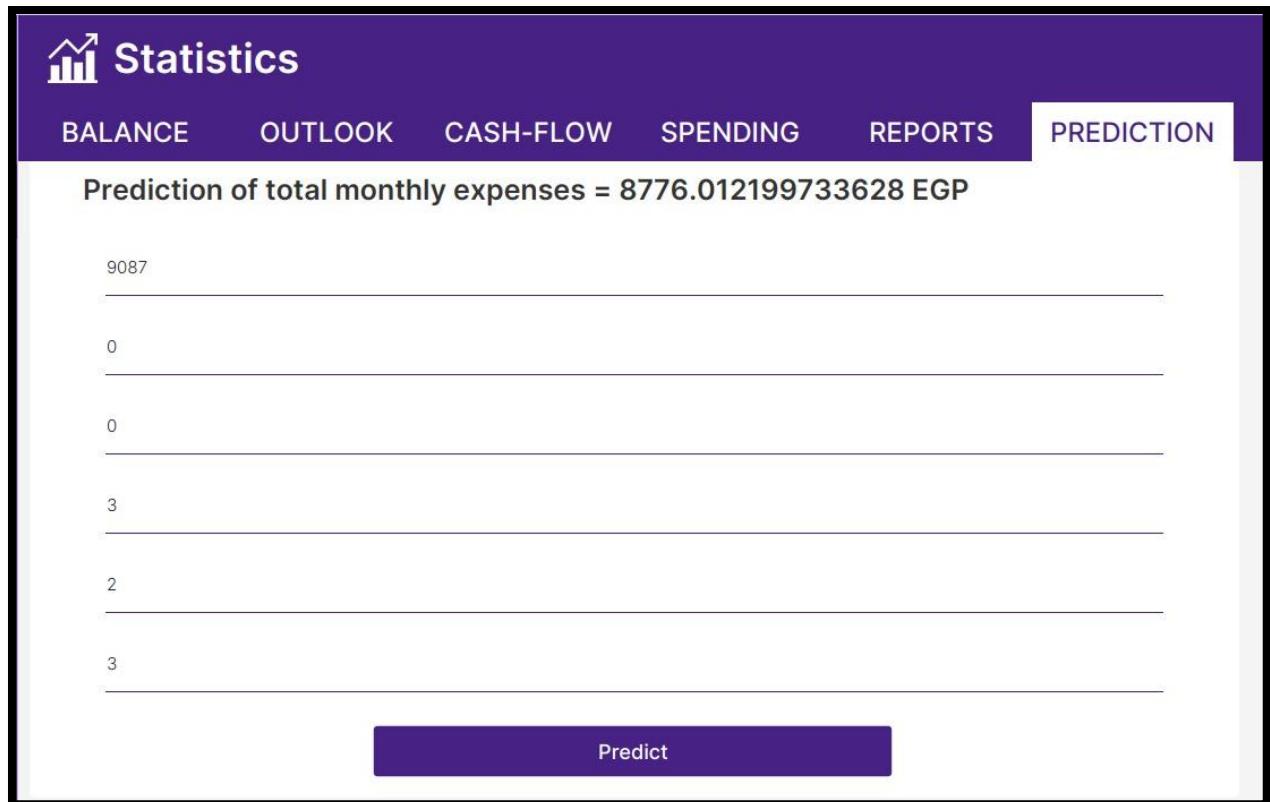


Figure 5: Example of Using the Prediction model from the real App

Gantt Chart

Project Gantt chart is divided into two parts to be more clear and these two parts are (analysis and implementation).

1. Analysis Gantt chart

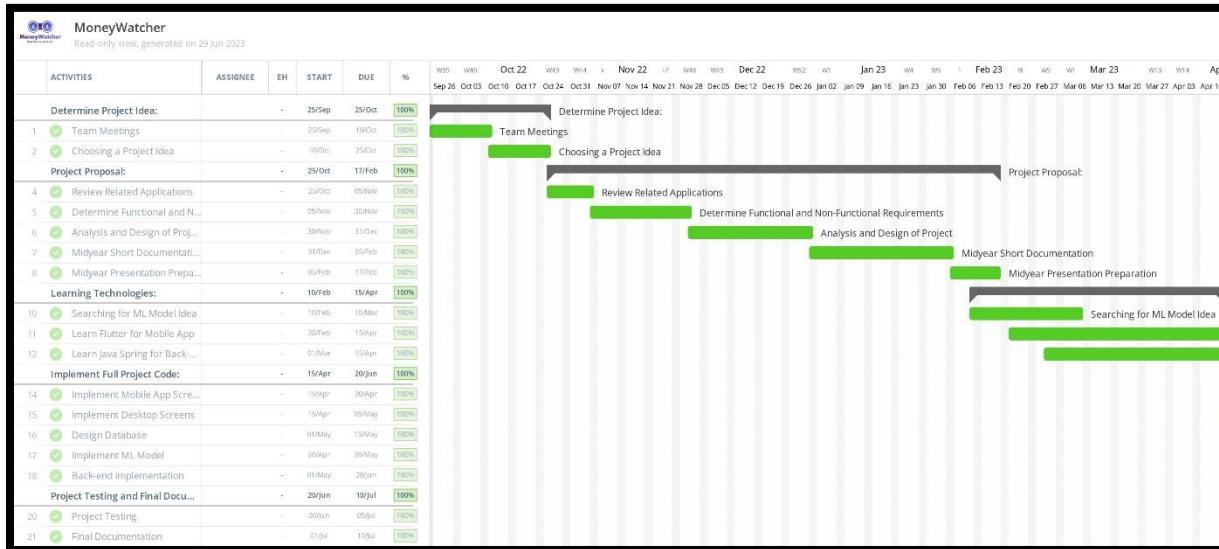


Figure 6: Analysis Gantt Chart

2. Implementation Gantt chart

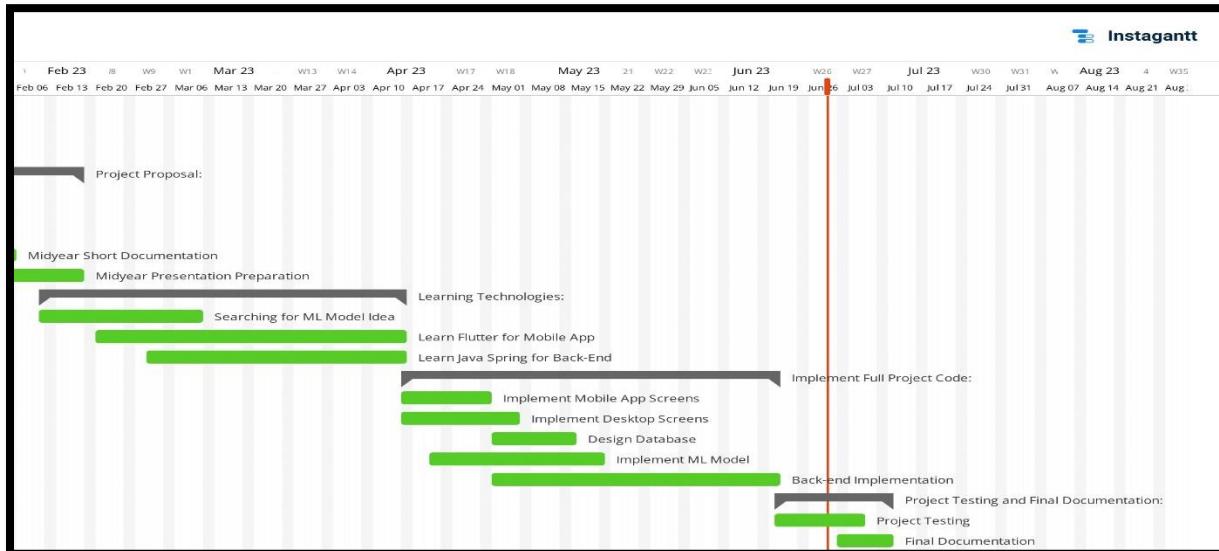


Figure 7: Implementation Gantt Chart

Project Development Methodology

Waterfall

The Waterfall methodology is a linear and sequential approach to software development. It is characterized by a series of distinct phases that are followed in a strict order, with each phase building upon the results of the previous one. The phases typically include requirements gathering and analysis, system design, implementation, testing, deployment, and maintenance.

Requirements

In the requirements phase of the software development process, the main goal is to gather a thorough understanding of the user's needs and define the specifications of the software product. This phase is crucial as it lays the foundation for the entire development process.

Design

In the design phase of software development, the goal is to translate the requirements gathered in the previous phase into a detailed design that will guide the implementation of the software. The design phase focuses on defining the structure, architecture, and components of the final solution.

Implementation

In the implementation phase of software development, the focus is on translating the design specifications into actual working code. This is where the development team takes the detailed design and builds the software system.

Testing

Is a critical phase in the software development process, aimed at identifying and fixing any defects or issues in the system before it is deployed to the users. A test plan will be followed. Different test cases will be executed according to three types: unit testing, integration testing, and usability testing.

Deployment and Maintenance

are two essential stages in the software development process. Deployment involves releasing the developed software system and making it available for use by the intended users. It includes activities such as configuring the system, setting up servers, and ensuring a smooth transition to the production environment. On the other hand, Maintenance focuses on ensuring the ongoing operation and performance of the software system. It involves monitoring, bug fixing, applying updates, and providing technical support to users. Together, these stages ensure that the software system is successfully deployed and maintained to deliver reliable and efficient functionality to its users over time.

The Used Tools

Frontend

- *Flutter*

Our main technology for the front-end of our mobile app was Flutter, we choose flutter because it offers the following benefits:

- Flutter has a fast development cycle, which means that you can build and release your app quickly.
- Flutter allows you to build apps that can run on both Android and iOS platforms using a single codebase.

- *JavaFX*

Provides a set of powerful UI controls, layout managers, and styling options that allow you to create visually appealing and interactive user interfaces for desktop application.

Backend

- *Java + Spring Boot*

Our main technology for the back end of our application was Spring Boot, we chose Spring because it offers the following benefits:

- Helps greatly in building loosely coupled applications.
- Easy separation of different layers and packages.
- Minimum boilerplate code for configuration.
- Can be easily integrated with other applications and servers.

Machine Learning

- *Python*

Python was the language of choice for the Machine Learning model because almost all ML frameworks and libraries are supported in Python.

- *SKLearn Pandas*

Pandas was used to read data in the form of Data Frames in order to easily apply SKLearn techniques on our data.

- *Flask*

Flask was used to build a REST API that is used by the back-end application to find suggested peers based on personality.

Report Organization

The Next Chapters Will Discuss **MoneyWatcher** in More Details

Chapter 2

Some Similar Solutions to the Same Problem and the Difference from our Project

Chapter 3

Details of the Requirements in the System and Use Cases

Chapter 4

More About the Design of **MoneyWathcer** in Details:

- System Component Design
- System Classes UML
- Sequence Diagrams
- System ERD
- System GUI

Chapter 5

Implementation and Testing of the Application

Chapter 2: Related Work

Existing Similar targeted apps

1) 1Money

An easy application to keep track of expenses, quick budget, and personal finance manager.

- Features

- Gives Statistical Histograms for your expenses and incomes per day, month, year, and lifetime.
- Adding, and deleting categories as you want.
- Passcode for the application for security purposes.
- You can add saving goals for your budget.
- Exporting data as a CSV file.

- Weaknesses

- Backup in paid version only.
- You need a subscription to enjoy the full features.

2) Money Manager

Another android application for financial tracking which contains the easiest way to manage personal finances.

- Features

- Three different money wallets (Cash, Accounts, Card).
- You can add a photo description for the transaction.
- Provide a statistical chart for incomes and expenses in a daily, monthly, yearly, or period manner.
- Have a passcode for security, and reset the application.
- You can repeat a certain transaction every period of time.

- Weaknesses

- Not easy to find what you're looking for or organize something.

3) Wallet

The best mobile application we found, it has extra terrific features.

- Features

- You can add expenses and incomes or transfer from different money sources and categories.
- The transactions have a date, time, place, labels, and receipt.
- Have a different statistical chart for incomes and expenses.
- You can schedule your payments with the Planned Payment feature.
- Managing your debts, track what you lent and borrowed.

4) El-Masareef

Android application developed by Arab developers and it is the best application for the Arab audience.

- Features
 - The transactions have a category, date, time, place, labels, and picture for receipt.
 - Have a different statistical chart for income and expenses.
 - Adding new categories, or deleting existing ones.
 - Search for a transaction on a range of dates.
- Weaknesses
 - Not all features in the free version.
 - In the free version, advertisements appear during use in a very exaggerated way.

Difference in our application

While there are existing apps that target personal finance management and expense tracking, our application stands out due to several key differences:

1. Our application works seamlessly on both mobile devices and desktop computers, allowing users to manage their finances from anywhere, whether they're on the go or in the comfort of their own space.
2. Our app provides comprehensive reports on expenses, income, budgets, and savings. These reports offer detailed insights into users' financial situations, enabling them to make informed decisions and take steps toward their financial goals.
3. We use machine model to predict the total monthly expenses which provides user with valuable insights about his expected expense, user budgeting and planning for his financial situation, user also can treat this prediction as a threshold which he can't exceed, also helping in goal tracking and savings for achieving goal.

By incorporating these unique features and functionalities, our application aims to provide a comprehensive and user-centric approach to personal finance management. We believe that these differences set our app apart from existing solutions and offer users a more efficient, insightful, and personalized experience in managing their finances.

Chapter 3: System Analysis

Project Specification

System Architecture

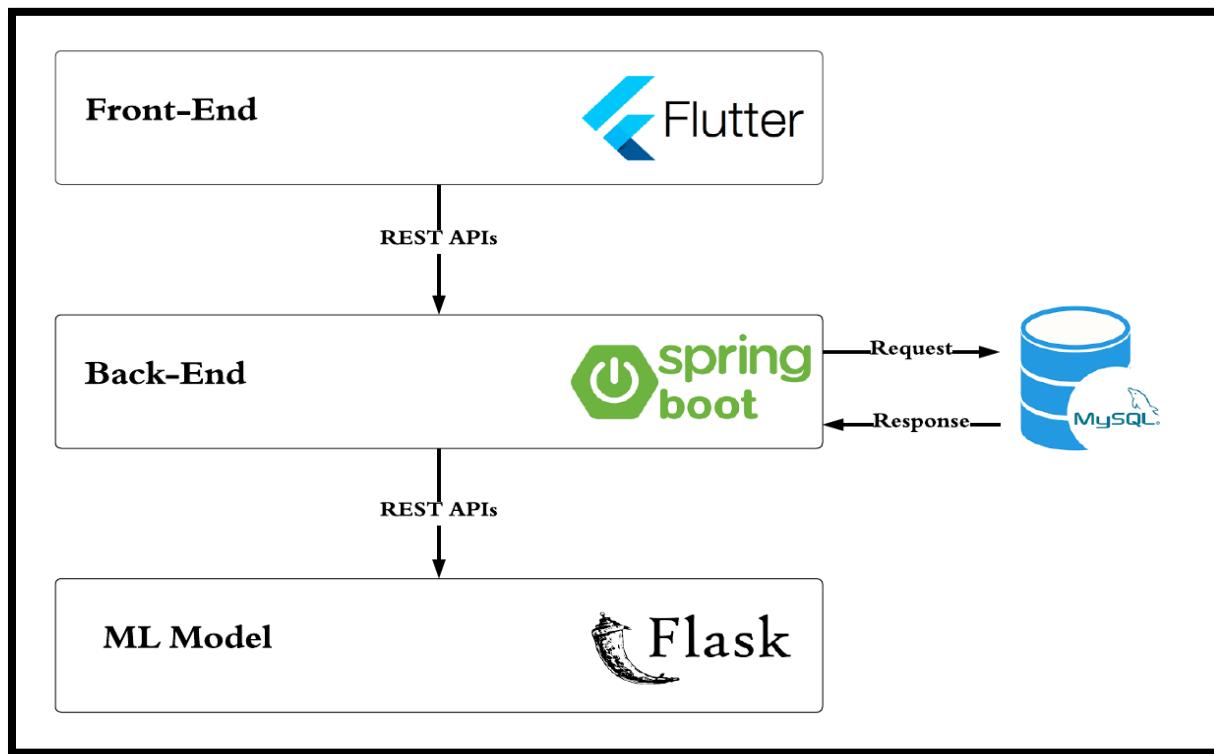


Figure 8: System Architecture

Stakeholders

1. Anyone interested in managing their money and expenses: This is a broad category that includes employees and anyone who wants to organize and manage their personal finances.
2. People who need to control and manage their expenses: This category includes individuals who tend to spend their money on unnecessary purchases and payments. Through the **MoneyWatcher** app, these individuals can prioritize their expenses, reduce overspending, and achieve savings.
3. Anyone can use the app as it has a friendly UI and nice UX to make the experience easier.

Functional Requirements

1. Basic Requirements

1.1. Login and Registration

1.1.1. The user should be able to register to the system with his email and password.

1.1.2. The user should log in to the system if he is already registered.

1.2. User main operations

1.2.1. User should be able to add expenses, incomes, and debts in a good convenient way.

1.2.2. User should be able to add categories for his expenses, and income.

1.2.3. User should be able to add a new category to his expenses if it doesn't exist.

1.2.4. User should be able to know the amount of money he spent on a specific day, month year.

1.2.5. The user should be able to add expenses, incomes, and debts in a good convenient way.

1.2.6. The expenses and incomes will be recorded in a daily manner with basic information such as price, date, category, receipt, also notes.

1.3. Plans

1.3.1. The user also has to be able to track what he lent and what he borrowed in a scheduled manner - supported with dates – and also keep a history of them.

1.3.2. Also, the user can set a goal for himself, the goal can be anything he wants to achieve, buy or save, he can add what actually he did for the goal and what's the target of the goal such as how much money it needs. He can keep track of it manually, not automatically calculated.

1.3.3. The user can schedule some of the payments or incomes, such as salaries and installments, he does not need to do it every time it happens, just added or subtracted automatically.

1.4. Notifications

1.4.1. The user must be notified of any planned payment or loan time before it happens and must give the confirmation first, also a daily notification to tell him to save his records if he forgot.

1.4.2. The application must have a separate part for functions control (settings part).

2. Statistical and Graphical Requirements

- 2.1. The user can view and see some numerical statistics about his money such as the average spending per day, month, year, comparisons between different months, and years, and which categories take how much money.
- 2.2. The statistics should be supported with graphical representations to be more formal and convenient.
- 2.3. The user also can see a report about his transactions and actions in a comprehensive way.

3. I/O Requirement

- 3.1. The user can export his records in different file formats such as Excel, CSV, PDF, or just a PNG.
- 3.2. The user also can export some graphical statistics as PNG.

4. Machine Learning Requirement

- 4.1. Machine learning to predict the total monthly expenses which provides user with valuable insights about his expected expense, user budgeting and planning for his financial situation, user also can treat this prediction as a threshold which he can't exceed, also helping in goal tracking and savings for achieving goal.

Non-functional Requirements

1. Usability
 - 1.1. The system will be easy to use for users by providing flexible UI.
2. Robustness
 - 2.1. The system will deal with bad user inputs and wrong data.
3. Reliability
 - 3.1. The system will show only the data related to each user.
 - 3.2. No other users can access another user's data.
4. Performance
 - 4.1. Response Time
 - 4.1.1. The system will respond to user requests in a minimum amount of time.
 - 4.2. Availability
 - 4.2.1. The application can work offline.
 - 4.2.2. The online version will be available so that users can reach their data from another place, data will be saved if a user is online.
5. Security
 - 5.1. Authentication
 - 5.1.1. The application will know the identity of a user when the user login with their email and password.

Use Case Diagrams

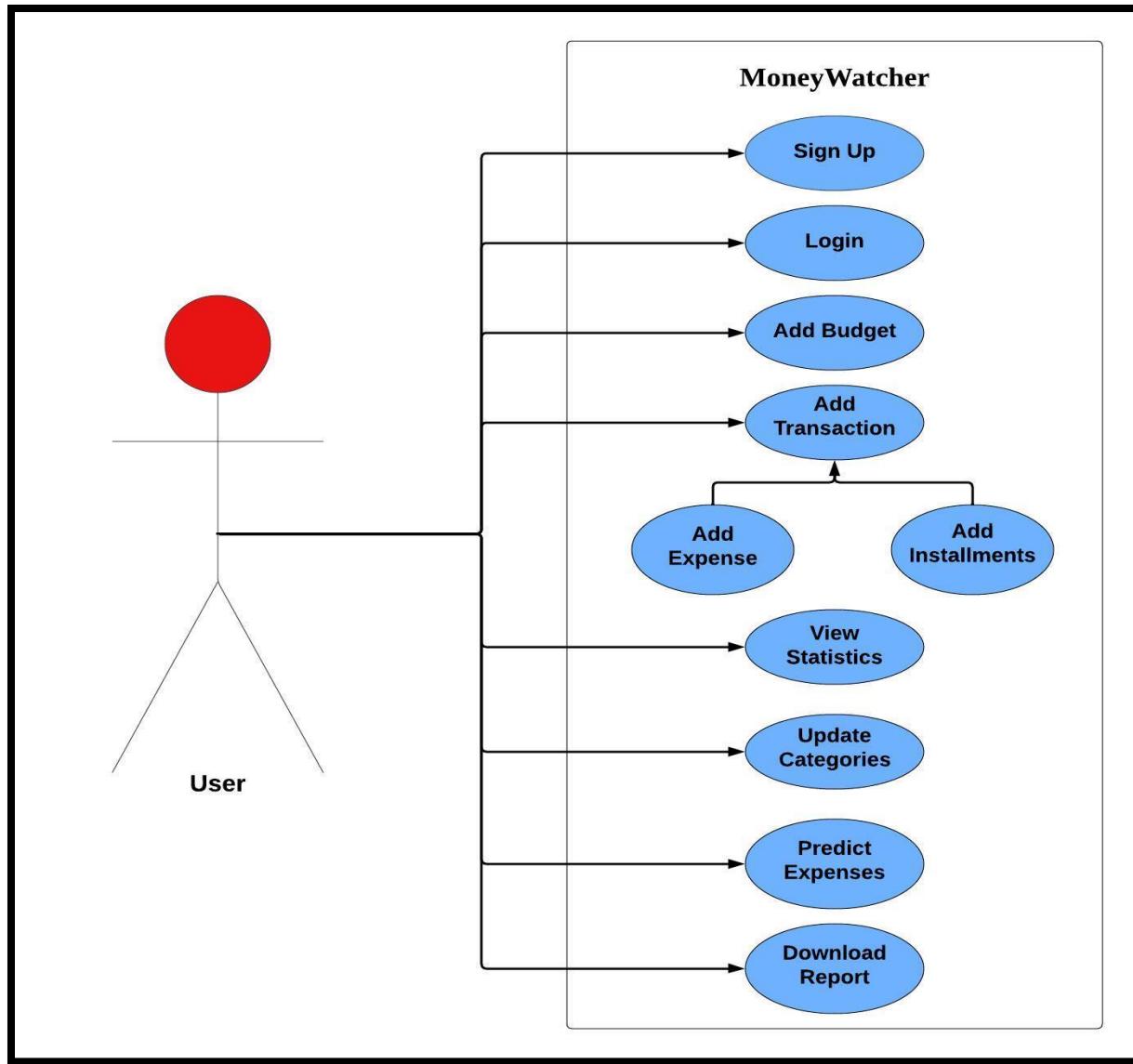


Figure 9: Use Case Diagram

Use Case Tables (User Stories)

Use Case ID	US #1
Use Case Name:	Sign UP
Actors:	User
Description:	<p>As a user.</p> <p>I like to be able to sign up in the system.</p> <p>So that I create my profile.</p>
Pre-conditions:	User is not registered on the mobile app.
Post-conditions:	User profile is added to the system.
Flow of Events:	<ol style="list-style-type: none"> 1- User enters his first name, last name, email address, password. 2- System checks the validation of inputs. 3- User is redirected to the login page.

Table 1: Use Case 1 --> Sign Up

Use Case ID	US #2
Use Case Name:	Login
Actors:	User
Description:	<p>As a user.</p> <p>I like to be able to login into the system.</p> <p>So that I entered the home page.</p>
Pre-conditions:	User is Signed Up.
Post-conditions:	User can add transaction, edit profile, view statistics, etc.
Flow of Events:	<ol style="list-style-type: none"> 1- User Types his first and last. 2- User Types his Password. 3- User clicks the “Login” button. 4- If the Credentials are wrong, the user is told so. 5- If the Credentials are Corrected the user will be redirected to his Home Page.

Table 2: Use Case 2 --> Login

Use Case ID:	US #3
Use Case Name:	Add Expense
Actors:	User
Description:	<p>As a user.</p> <p>I like to be able to add an expense to my profile.</p> <p>So that the expense will be saved in my profile.</p>
Pre-conditions:	User is logged into the system.
Post-conditions:	The Expense is saved in the User Profile.
Flow of Events:	<ol style="list-style-type: none"> 1- User enters an expense name. 2- User can enter a description for the expense "Optional". 3- User chooses a specific budget and category. 4- User enters an amount. 5- User enters a date of expense. 6- FINAL User clicked “Add Expense“.

Table 3: Use Case 5 --> Add Expense

Use Case ID:	US #4
Use Case Name:	Add Installment
Actors:	User
Description:	<p>As a user.</p> <p>I like to be able to add some notes of the installment date.</p> <p>So that the system will notify me of the date for payment installment.</p>
Pre-conditions:	User is logged into the account profile.
Post-conditions:	The app will notify the user of the date of payment installment.
Flow of Events:	<ol style="list-style-type: none"> 1- User enters the installment name to be paid. 2- User can enter a description for the installment "Optional". 3- User defines the monthly or yearly installment recurrence process. 4- User enters an amount of installment. 5- User enters the payment date of the installment. 6- FINAL User clicked “Add Installment”.

Table 4: Use Case 6 --> Add Installment

Use Case ID:	US #5
Use Case Name:	View Statistics
Actors:	User
Description:	<p>As a user.</p> <p>I like to be able to view my statistics in my profile.</p> <p>So that I can see a pie chart of my categories.</p>
Pre-conditions:	User is logged into the system.
Post-conditions:	The pie chart of categories is shown to the user.
Flow of Events:	<ol style="list-style-type: none"> 1- User must be logged in to the mobile app. 2- User chooses a “stats page” from the bottom bar in the app. 3- The pie chart will be shown divided into all categories and a percentage of the expenses of each category.

Table 5: Use Case 7 --> View Statistics

Use Case ID:	US #6
Use Case Name:	Predict Expenses
Actors:	User
Description:	<p>As a user.</p> <p>I like to be able to predict the monthly expenses.</p> <p>So that the app will show the predicted monthly salary based on some inputs that the user will enter it.</p>
Pre-conditions:	User is logged into the system.
Post-conditions:	The app will show the predicted monthly salary.
Flow of Events:	<ol style="list-style-type: none"> 1- User must be logged in to the mobile app. 2- User chooses a “extra” page from the bottom bar in the app. 3- Then the user will go to the “Predict Expenses” page. 4- User will enter 14 mandatory input fields for the machine learning model to predict the monthly salary. 5- FINAL User clicked “ Add Installment “.

Table 6: Use Case 9 --> Predict Expenses

Chapter 4: System Design

System Components Diagram

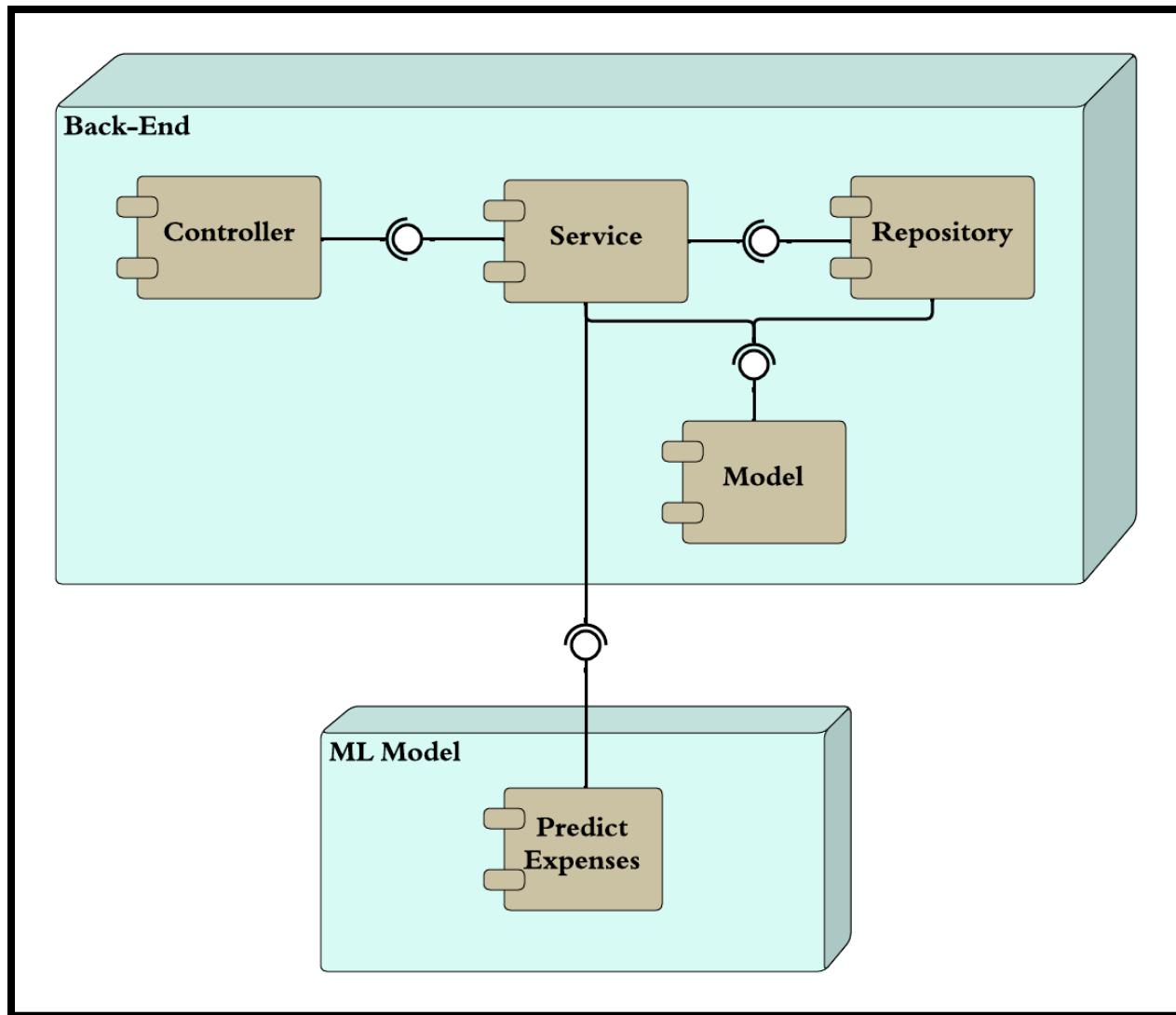


Figure 10: Component Diagram

System Class Diagrams



Figure 11: Class Diagram 1

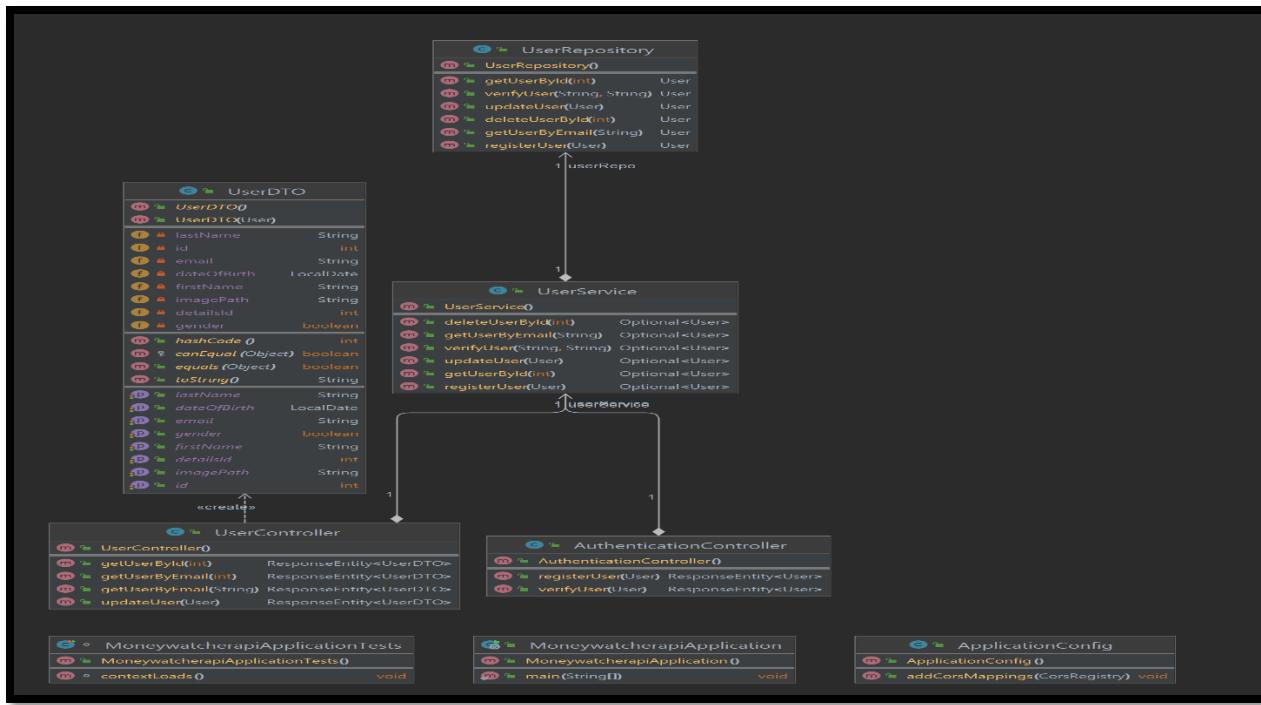


Figure 13: Class Diagram 2



Figure 12 : Class Diagram 3

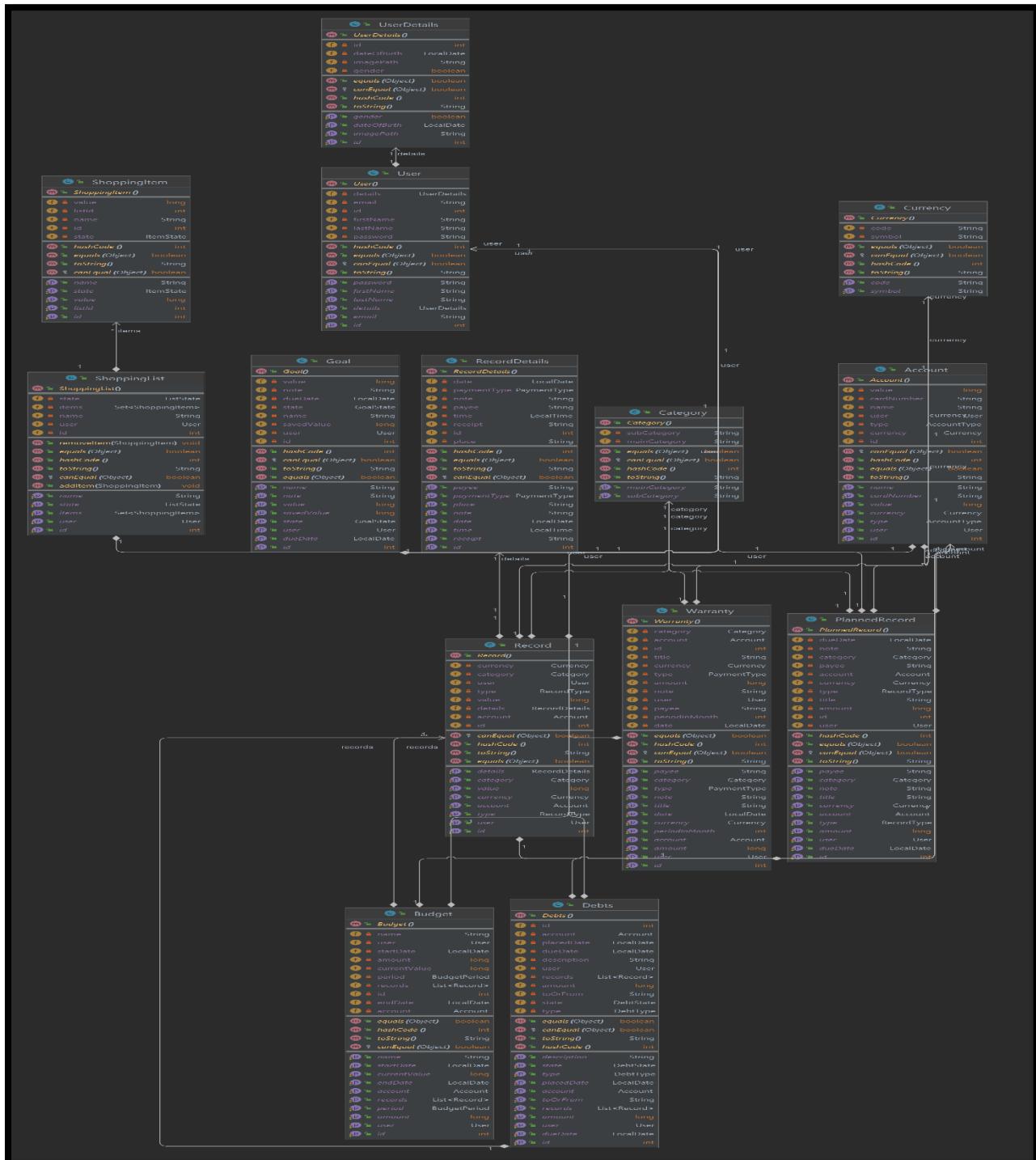


Figure 14: Class Diagram 4

Sequence Diagrams

1. Sign UP

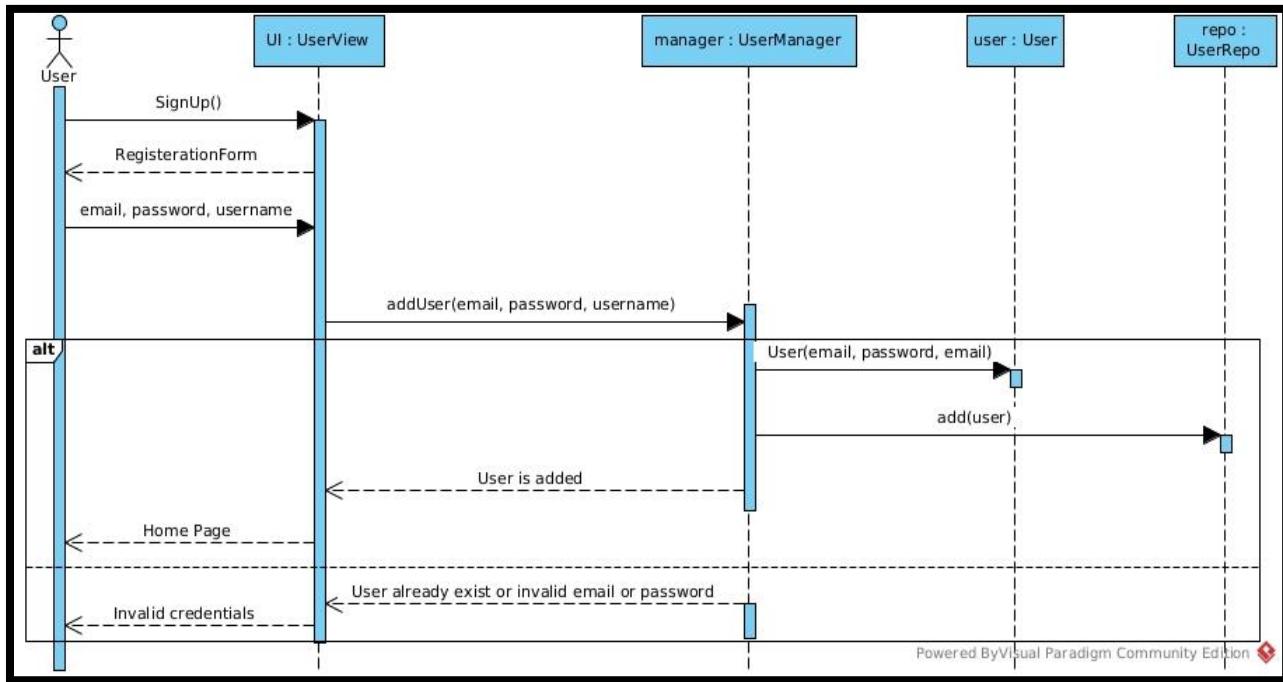


Figure 15: Sequence 1 (Sign Up)

2. Login

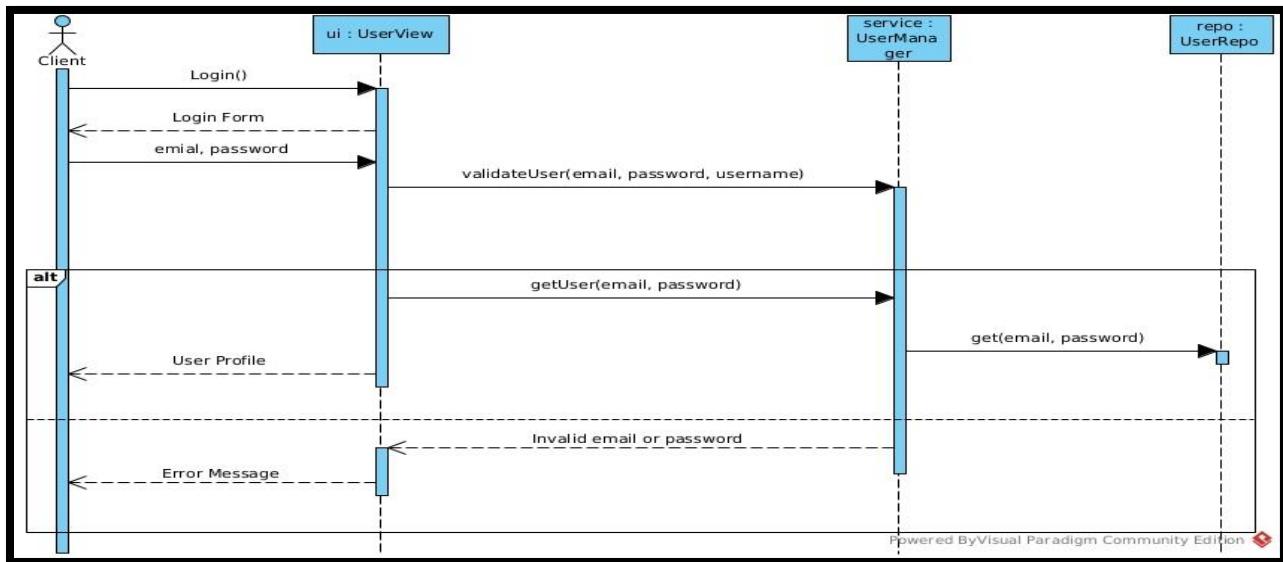


Figure 16: Sequence 2 (Login)

3. Add Budget

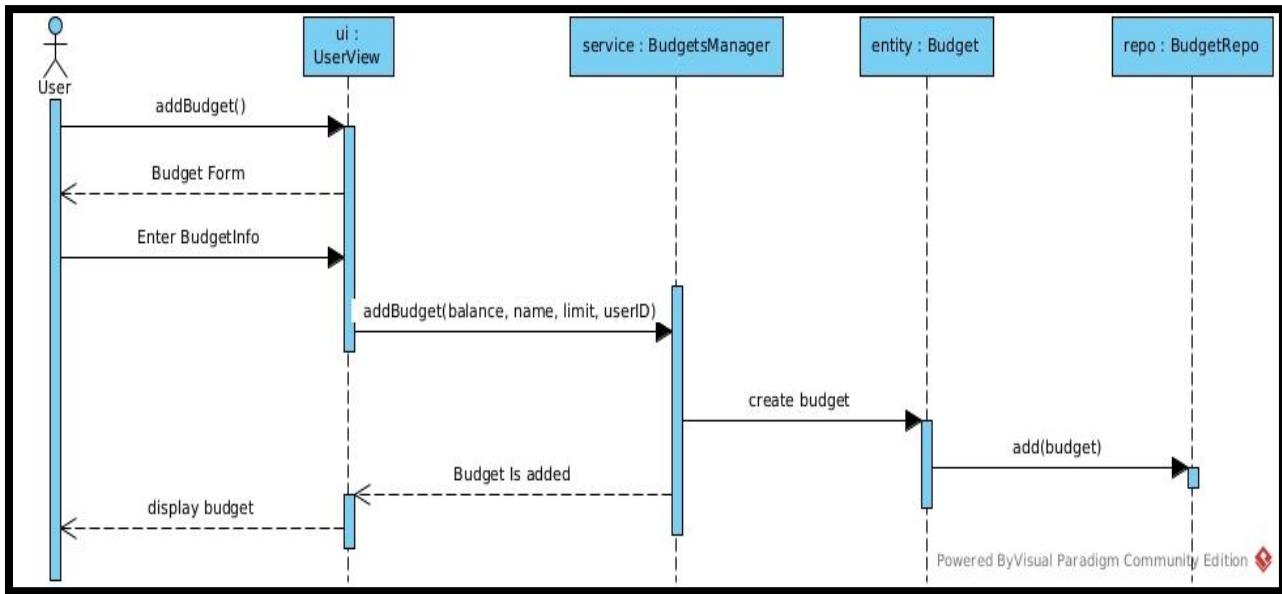


Figure 17: Sequence 3 (Add Budget)

4. Add Expense

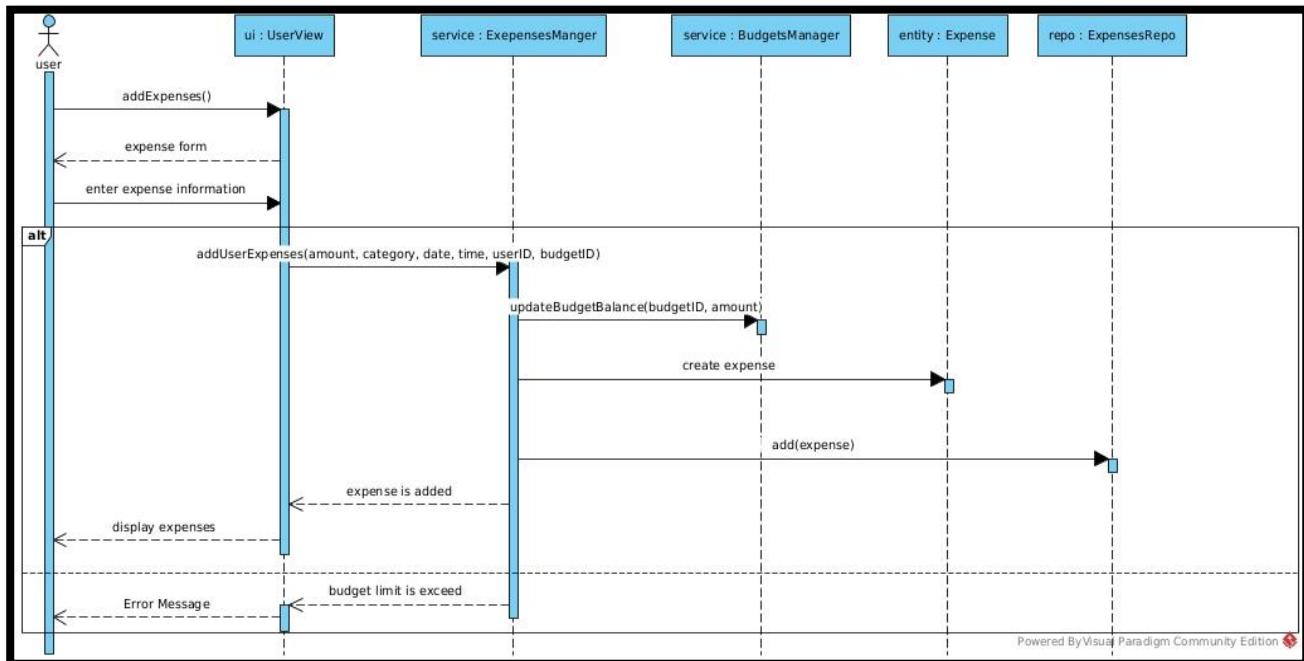


Figure 18: Sequence 4 (Add Expense)

5. Export Report

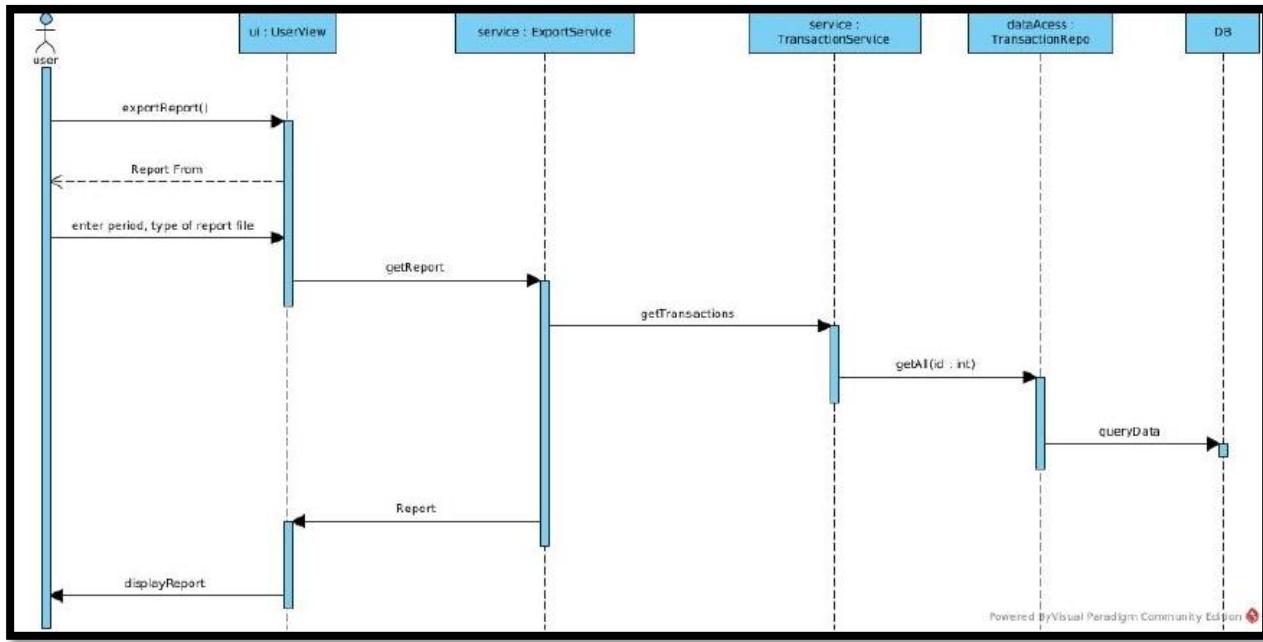


Figure 19: Sequence 5 (Export Report)

6. Show Statistics

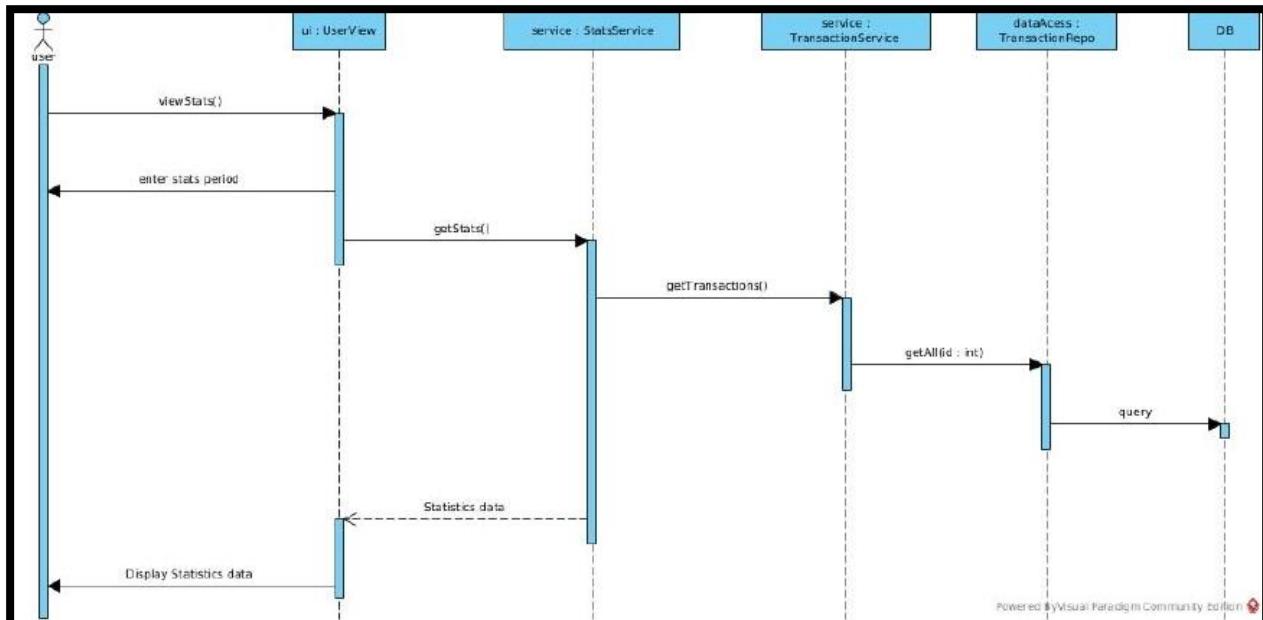


Figure 20: Sequence 6 (View Statistics)

Project ERD

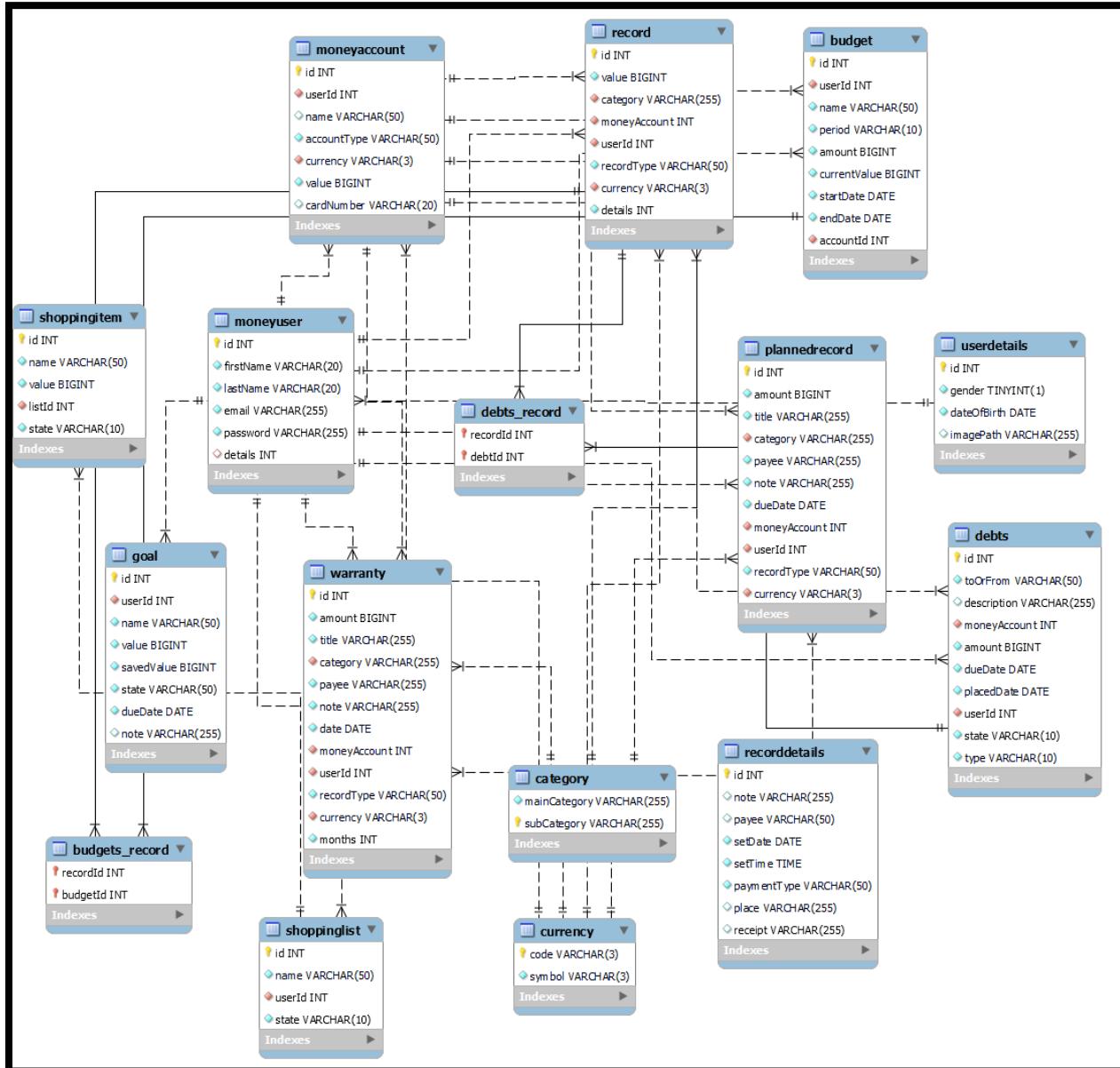
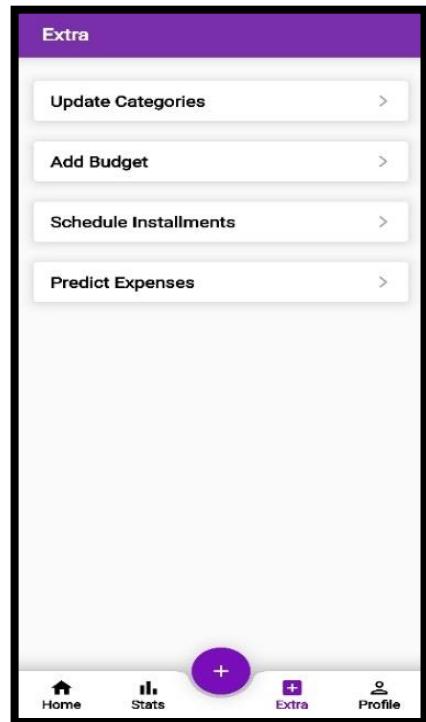
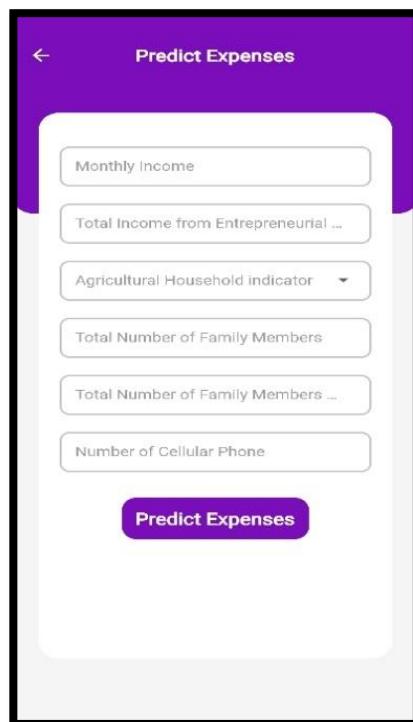
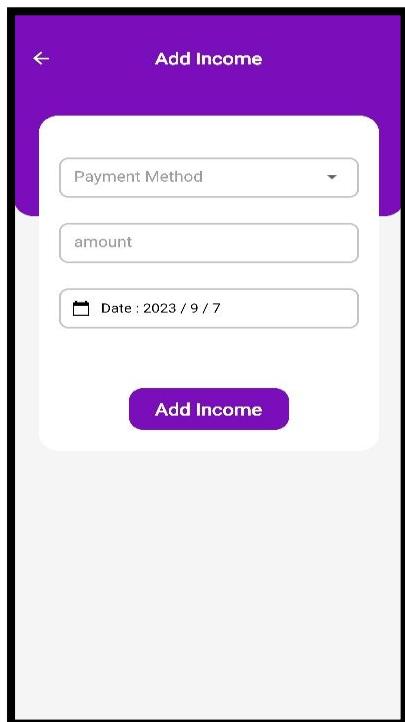
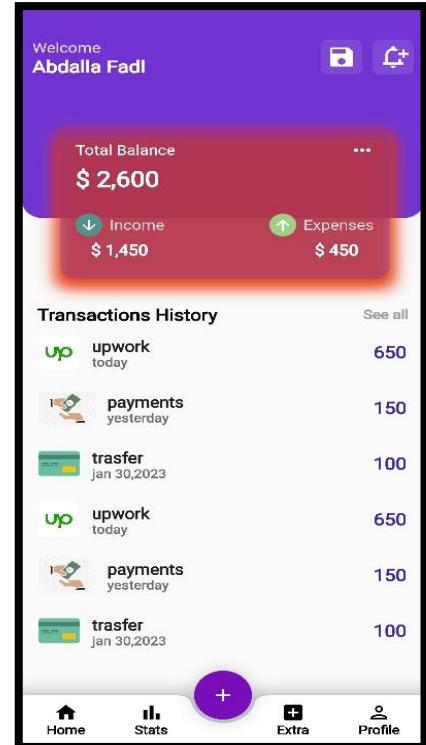
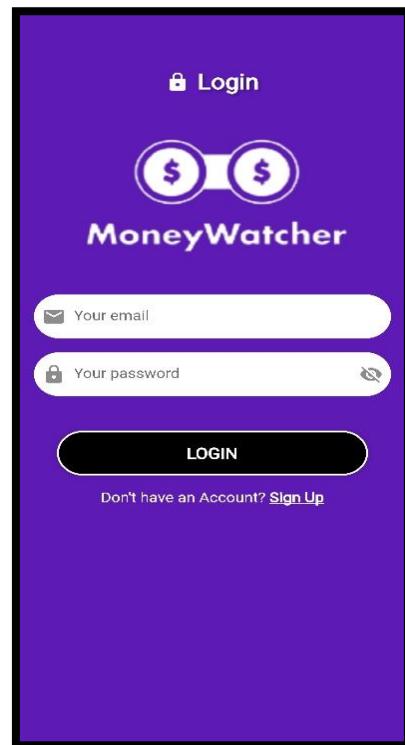
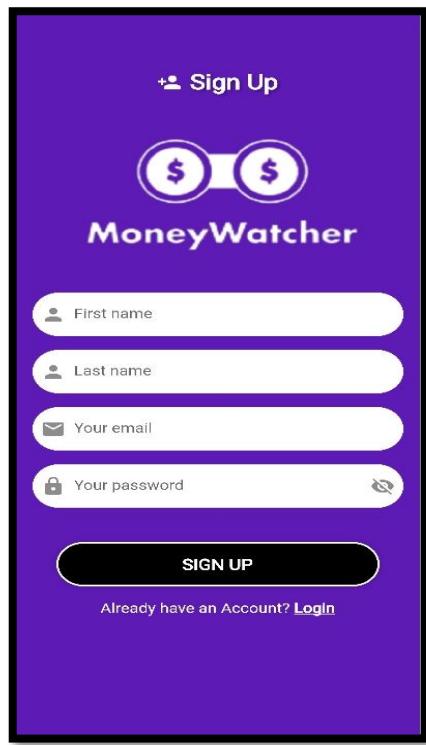
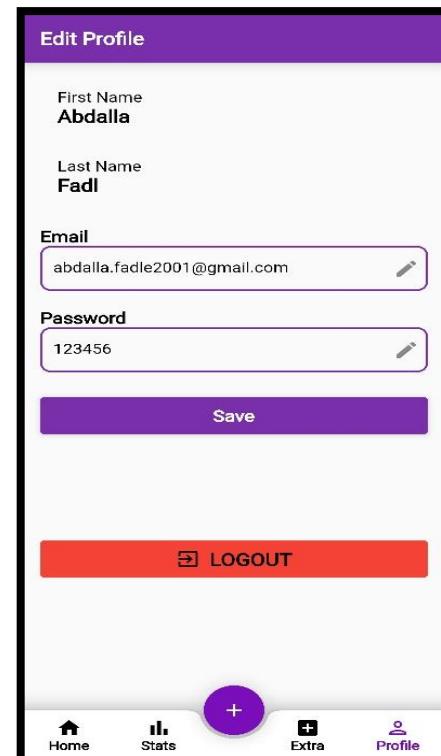
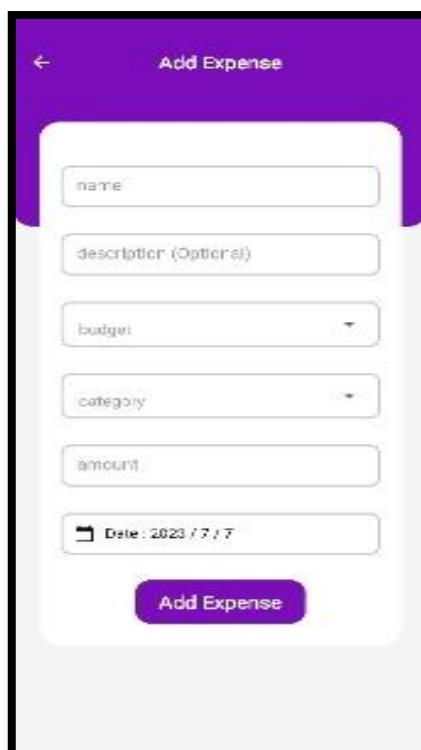
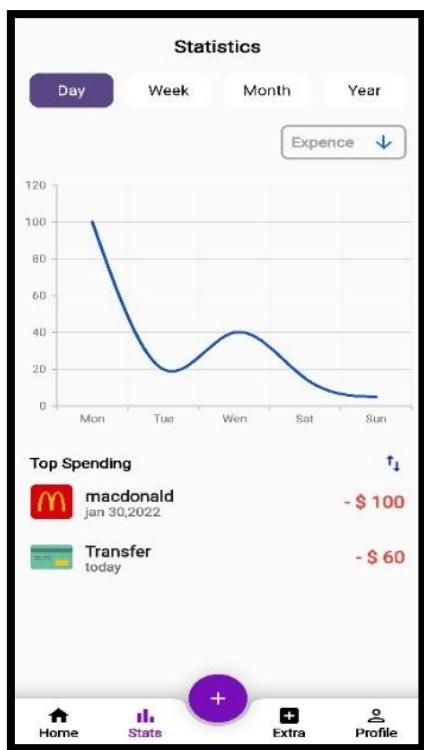
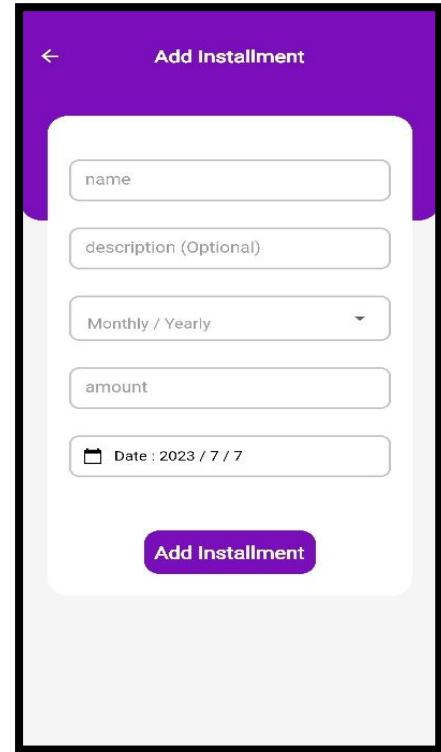
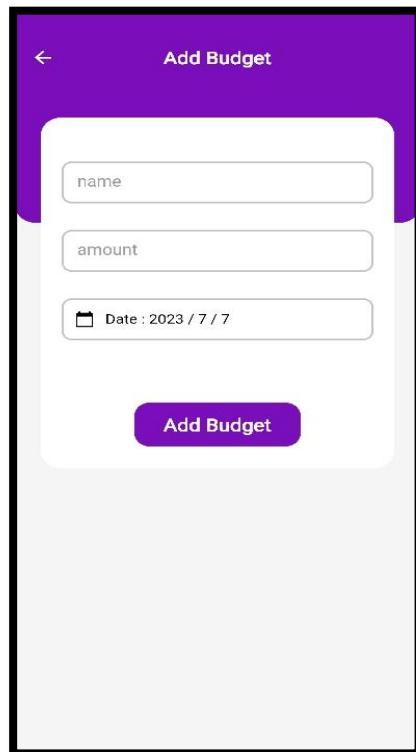
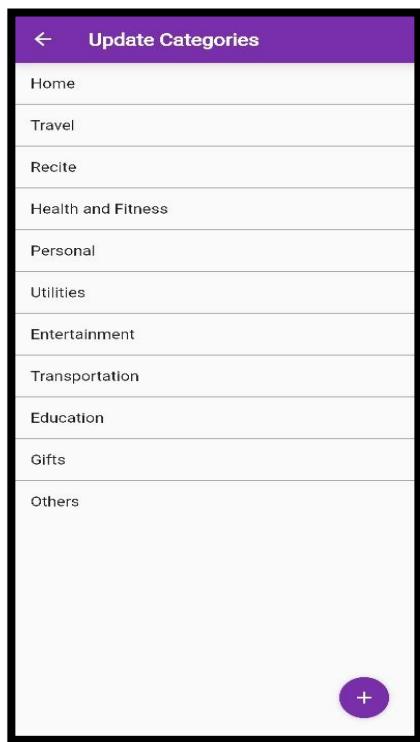


Figure 21: System ERD

System GUI Design





Chapter 5: Implementation and Testing

Implementation

Frontend

- *Flutter*

Our main technology for the front-end of our mobile app was Flutter, we choose flutter because it offers the following benefits:

- Flutter has a fast development cycle, which means that you can build and release your app quickly.
- Flutter allows you to build apps that can run on both Android and iOS platforms using a single codebase.

- *JavaFX*

Provides a set of powerful UI controls, layout managers, and styling options that allow you to create visually appealing and interactive user interfaces for desktop application.

Backend

- *Java + Spring Boot*

Our main technology for the back end of our application was Spring Boot, we chose Spring because it offers the following benefits:

- Helps greatly in building loosely coupled applications.
- Easy separation of different layers and packages.
- Minimum boilerplate code for configuration.
- Can be easily integrated with other applications and servers.

Machine Learning

- *Python*

Python was the language of choice for the Machine Learning model because almost all ML frameworks and libraries are supported in Python.

- *SKLearn Pandas*

Pandas was used to read data in the form of Data Frames in order to easily apply SKLearn techniques on our data.

- *Flask*

Flask was used to build a REST API that is used by the back-end application to find suggested peers based on personality.

Testing

Functionality	Input	Expected Output
Sign Up	Empty	“Sign Up” button is disabled until the form is filled completely
	Full Credentials but wrong Email Format First Name: Abdalla Last Name: Fadl Email: abdalla.fadle Password: Abdalla_2001	Invalid email format error message
	Full Credentials but wrong Password Format First Name: Abdalla Last Name: Fadl Email: abdalla.fadle1995@gmail.com Password: 12345678	Invalid password format Password should contain both letters and numbers
	Full Correct Credentials First Name: Abdalla Last Name: Fadl Email: abdalla.fadle1995@gmail.com Password: Abdalla_1995	Create The Account and Forward to Login Page
Login	Empty	Error (Must Enter Credentials)
	Wrong Email	Error (Wrong Username or Password)
	Wrong Password	Error (Wrong Username or Password)
	Correct Email and Password Email: abdalla.fadle1995@gmail.com Password: Abdalla_1995	Login in Successfully and redirect to the Home Page
Add Expense	Missing Name Name: Description: “Bought groceries for the week” Budget: Expense Category: Food Amount: 100.00 Date: 2023-07-07	Error (Name filed is required)

	<p>Missing Description (Optional)</p> <p>Name: Grocery Shopping</p> <p>Description: ""</p> <p>Budget: Expense</p> <p>Category: Food</p> <p>Amount: 100.00</p> <p>Date: 2023-07-07</p>	No validation error for the Description field
	<p>Amount Negative Value</p> <p>Name: Grocery Shopping</p> <p>Description: Bought groceries for the week</p> <p>Budget: Expense</p> <p>Category: Food</p> <p>Amount: -100.00</p> <p>Date: 2023-07-07</p>	Invalid amount value. Amount cannot be negative
	<p>Valid Input</p> <p>Name: Grocery Shopping</p> <p>Description: "Bought groceries for the week"</p> <p>Budget: Expense</p> <p>Category: Food</p> <p>Amount: 100.00</p> <p>Date: 2023-07-07</p>	Valid Input (Expense Added Successfully)
Add Budget	<p>Empty Name</p> <p>Name: ""</p> <p>Amount: 200.00</p> <p>Date: 2023-07-07</p>	Error message displayed (Name field is required)
	<p>Negative Amount</p> <p>Name: Shopping</p> <p>Amount: -50.00</p> <p>Date: 2023-07-07</p>	Error message (Amount must be a positive value)
	<p>No Inputs</p> <p>Name:</p> <p>Amount:</p> <p>Date: Not selected</p>	Budget not be added, and error messages displayed (Name and Amount fields are required)
	<p>Valid Inputs:</p> <p>Name: "Groceries"</p> <p>Amount: "200.00"</p> <p>Date: 2023-07-07</p>	Budget added successfully with the provided name, amount, and date
Add Income	<p>Empty Payment Method:</p> <p>Payment Method:</p> <p>Amount: 500.00</p> <p>Date: 2023-07-07</p>	Error message (payment method is required)

	Invalid Date: Payment Method: ELECTRONIC WALLET Amount: 750.00 Date: 2023-13-08	Error message (selected date is invalid)
	No Inputs: Payment Method: Amount: Date:	Income not be added, and an error message (payment method, amount, and date fields are required)
	Valid Inputs: Payment Method: CASH Amount: 1000.00 Date: 2023-07-07	Income added successfully
Add Installment	No Inputs: Name: Description: Category: Amount: Date:	Installment not be added, and error messages (name, category, and amount fields are required)
	Invalid Date: Name: Insurance Premium Description: Monthly insurance premium Category: Yearly Amount: 400.00 Date: 2024-13-07	Installment not be added, and an error message (selected date is invalid)
	Negative Amount: Name: Gym Membership Description: Yearly gym membership fee Category: Yearly Amount: -500.00 Date: 2023-07-07	Invalid amount value. Amount cannot be negative
	Valid Inputs: Name: Car Loan Description: "Monthly installment for car loan" Category: Monthly Amount: 500.00 Date: 2023-07-07	Installment added successfully
Predict Expenses	No Inputs: Monthly Income: Total Income from Entrepreneurial Activities: Agricultural Household Indicator:	Expenses not be predicted, and error messages (Monthly Income, Total Number of Family Members,

	Total Number of Family Members: Total Number of Family Members Employed: Number of Cellular Phone:	Total Number of Family Members Employed, and Number of Cellular Phone fields are required)
	Empty Monthly Income: Monthly Income: Total Income from Entrepreneurial Activities: 1500.00 Agricultural Household Indicator: 1 Total Number of Family Members: 3 Total Number of Family Members Employed: 1 Number of Cellular Phone: 2	Expenses not be predicted, and an error message (Monthly Income field is required)
	Empty Total Number of Family Members Employed: Monthly Income: 1500.00 Total Income from Entrepreneurial Activities: Agricultural Household Indicator: 2 Total Number of Family Members: Total Number of Family Members Employed: 2 Number of Cellular Phone: 2	Expenses not be predicted, and an error message (Total Number of Family Members Employed field is required)
	Valid Inputs: Monthly Income: 5000.00 Total Income from Entrepreneurial Activities: 0 Agricultural Household Indicator: YES Total Number of Family Members: 5 Total Number of Family Members Employed: 2 Number of Cellular Phone: 3	The expenses predicted successfully based on the provided inputs.

Conclusion

In conclusion, the **MoneyWatcher** application has been developed with the aim of assisting users in effectively managing and monitoring their financial matters. The application is designed to facilitate tracking income and expenses, creating monthly budgets, and providing users with detailed reports and statistics.

Throughout the project development process, modern programming techniques and best practices have been employed to ensure high performance and a user-friendly interface. The main objectives of the project have been achieved, including providing comprehensive application for financial management and aiding users in making informed financial decisions.

In the end, we believe that the MoneyWatcher application will be a powerful and valuable tool for individuals seeking to improve their financial management. The application enables users to gain a better understanding of their financial lifestyle and achieve financial stability.

We look forward to sharing the application with users and receiving their feedback and comments to continue improving and enhancing MoneyWatcher in the future.

Finally, we would like to express our gratitude to all the individuals who helped us in this project, especially **Dr. Manar**, who provided us with all possible support and guidance, and we hope that MoneyWatcher will prove to be useful and effective in achieving financial stability for users.

References

- **ML Model Dataset**

Filipino Family Income and Expenditure
<https://shorturl.at/uyJPS>

- **ML Model Tutorial**

Medium Article
<https://rb.gy/b49by>

Kaggle notebook
<https://rb.gy/c89hd>

- **Flutter Tutorial**

<https://shorturl.at/cACI2>

<https://shorturl.at/enDIP>

- **Spring Boot Tutorial**

<https://shorturl.at/djoyQ>

- **JavaFX Tutorial**

<https://shorturl.at/fgpwR>