``

CS6004NP

Information Systems

****

**Module Code & Module Title**

**CS6004NP Application Development**

**Assessment Type**

**30% Individual Coursework**

**Semester- 5**

**2024 Autumn**

**Student Name: Aman Gurung**

**London Met ID: 22068950**

**College ID: NP04CP4A220019**

**Assignment Due Date: Sunday, January 5, 2025**

**Assignment Submission Date: Sunday, January 5, 2025**

**Submitted To: Mr. Mahesh Dhungana**

**Git:** <https://github.com/amangrx/DigiDhan>

**Declaration**

*I confirm that I understand my coursework needs to be submitted online via My second teacher under the relevant module page before the deadline for my assignment to be accepted and marked. I am fully aware that late submission will be treated as non-submission and a mark of zero will be awarded.*

**Table of Contents:**

[1. Introduction: 1](#_Toc187959959)

[1.1 Aims and Objectives: 2](#_Toc187959960)

[2 Feature and functionalities: 3](#_Toc187959961)

[2.1 Prototype: 3](#_Toc187959962)

[2.1.1 Prototype of Login page: 3](#_Toc187959963)

[2.1.2 Prototype of Home page: 4](#_Toc187959964)

[2.1.3 Prototype of Income page: 5](#_Toc187959965)

[2.1.4 Prototype of Expenses page: 6](#_Toc187959966)

[2.1.5 Prototype of Debt page: 7](#_Toc187959967)

[2.2 Developed Application: 8](#_Toc187959968)

[2.2.1 Login page: 8](#_Toc187959969)

[2.2.2 Home page: 8](#_Toc187959970)

[2.2.3 Income page: 8](#_Toc187959971)

[2.2.4 Expense page: 8](#_Toc187959972)

[2.2.5 Debt page: 9](#_Toc187959973)

[2.2.6 Tags page: 9](#_Toc187959974)

[2.3 Test cases: 10](#_Toc187959975)

[2.3.1 Test 1: Logging in 10](#_Toc187959976)

[3. Development and Logical solution: 11](#_Toc187959977)

[3.1 Entity Relationship Diagram: 11](#_Toc187959978)

[3.2 Use Case Diagram: 12](#_Toc187959979)

[3.3 Flowchart: 13](#_Toc187959980)

[3.4 Version Control (Git): 14](#_Toc187959981)

[3.5 Technology stack: 15](#_Toc187959982)

[3.5.1 Framework: 15](#_Toc187959983)

[3.5.2 External Libraries: 15](#_Toc187959984)

[3.5.3 Persistence Mechanism: 15](#_Toc187959985)

[3.6 Class, properties and methods: 16](#_Toc187959986)

[3.6.1 Classes and their properties: 16](#_Toc187959987)

[4 Conclusion: 17](#_Toc187959988)

[Bibliography 18](#_Toc187959989)

**Table of Figures:**

[Figure 1: Private Git repository initialized. 2](#_Toc186966665)

[Figure 2: UI for login. 3](#_Toc186966666)

[Figure 3: UI for home page. 4](#_Toc186966667)

[Figure 4: UI for income page. 5](#_Toc186966668)

[Figure 5: UI for expenses page 6](#_Toc186966669)

[Figure 6: UI for debt page. 7](#_Toc186966670)

[Figure 7: ERD 8](#_Toc186966671)

[Figure 8: Logging in. 10](#_Toc186966672)

[Figure 9: Adding expenses. 10](#_Toc186966673)

[Figure 10: Adding income. 11](#_Toc186966674)

[Figure 11: Adding Debt. 11](#_Toc186966675)

[Figure 12: Checking balance while recording expense. 12](#_Toc186966676)

[Figure 13: Displaying all transactions. 12](#_Toc186966677)

[Figure 14: Sort transactions by date. 13](#_Toc186966678)

[Figure 15: Displaying the highest and lowest of all transactions. 13](#_Toc186966679)

[Figure 16: Adding transaction without adding note. 14](#_Toc186966680)

[Figure 17: Displaying the total number of transactions performed and transaction amount. 14](#_Toc186966681)

**Table of Table:**

# Introduction:

This report documents our coursework project for Application Development, where we were tasked with creating a desktop application for personal expense tracking. The application was developed using the .NET Core framework and incorporates key features like tracking cash inflows, cash outflows, managing debts, searching transactions, and filtering or sorting them based on criteria like title, type, and more.

The project is titled Bachat Kendra. Bachat Kendra offers an intuitive interface and a userfriendly experience to its end users. Upon launching the application, users are directed to a login page where they can enter their credentials and select their preferred currency type. The application facilitates effortless monitoring of income and expenses while also emphasizing debt management. Users can input transaction amounts, dates, and optional labels or descriptions for better categorization. Furthermore, Bachat Kendra provides comprehensive insights, such as total income, expenses, and debt, alongside advanced features for searching and organizing transactions by title, type, date range, or tags. Users can also customize tags to suit their needs.

This report delves into various aspects of the project, including the userinterface design and the Entity Relationship Diagram (ERD). Additionally, it outlines the technology stack used in the development process, including the framework, libraries, and data persistence mechanisms.

## 1.1 Aims and Objectives:

The primary goal of this project is to facilitate the efficient management and monitoring of an individual's financial transactions. Additional objectives of the project include:

* Ensuring accurate tracking of income, expenses, and outstanding debts.
* Implementing exception handling and verifying sufficient balance before processing transactions.
* Providing robust search functionality for transactions by title, along with sorting and filtering options.
* Effectively listing and emphasizing unpaid debts.
* Displaying transaction details, including current balance, highest and lowest transactions, and more.
* Gaining a comprehensive understanding of the .NET framework and the database platform utilized.
* Incorporating version control mechanisms throughout the application development process.

# Feature and functionalities:

## 2.1 Prototype:

The prototype created for each page of the application is given below:

### 2.1.1 Prototype of Login page:

The UI for login page of Digi Dhan is given below:



Figure : UI for login.

### 2.1.2 Prototype of Home page:

The UI for home page of Digi Dhan is given below:

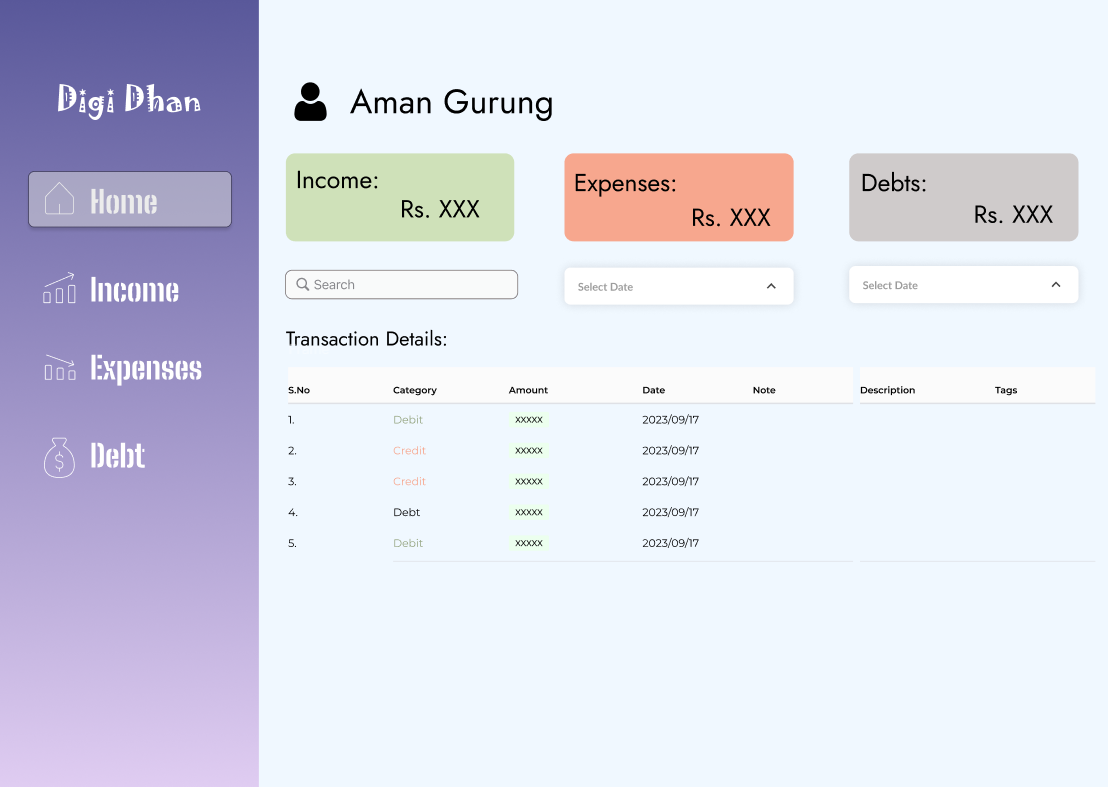


Figure : UI for home page.

### 2.1.3 Prototype of Income page:

The UI for income page of Digi Dhan is given below:

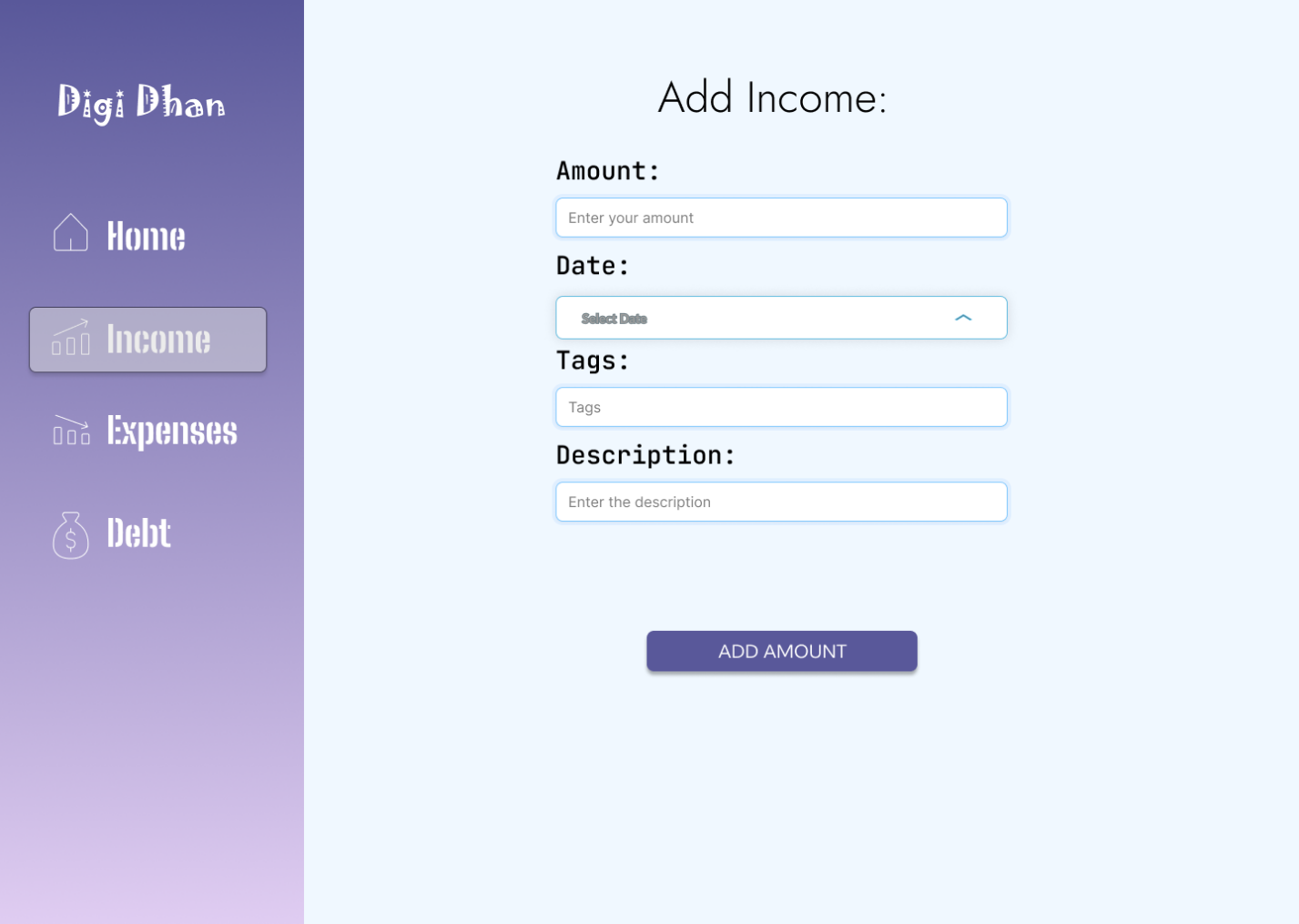


Figure : UI for income page.

### 2.1.4 Prototype of Expenses page:

The UI for expenses page of Digi Dhan is given below:

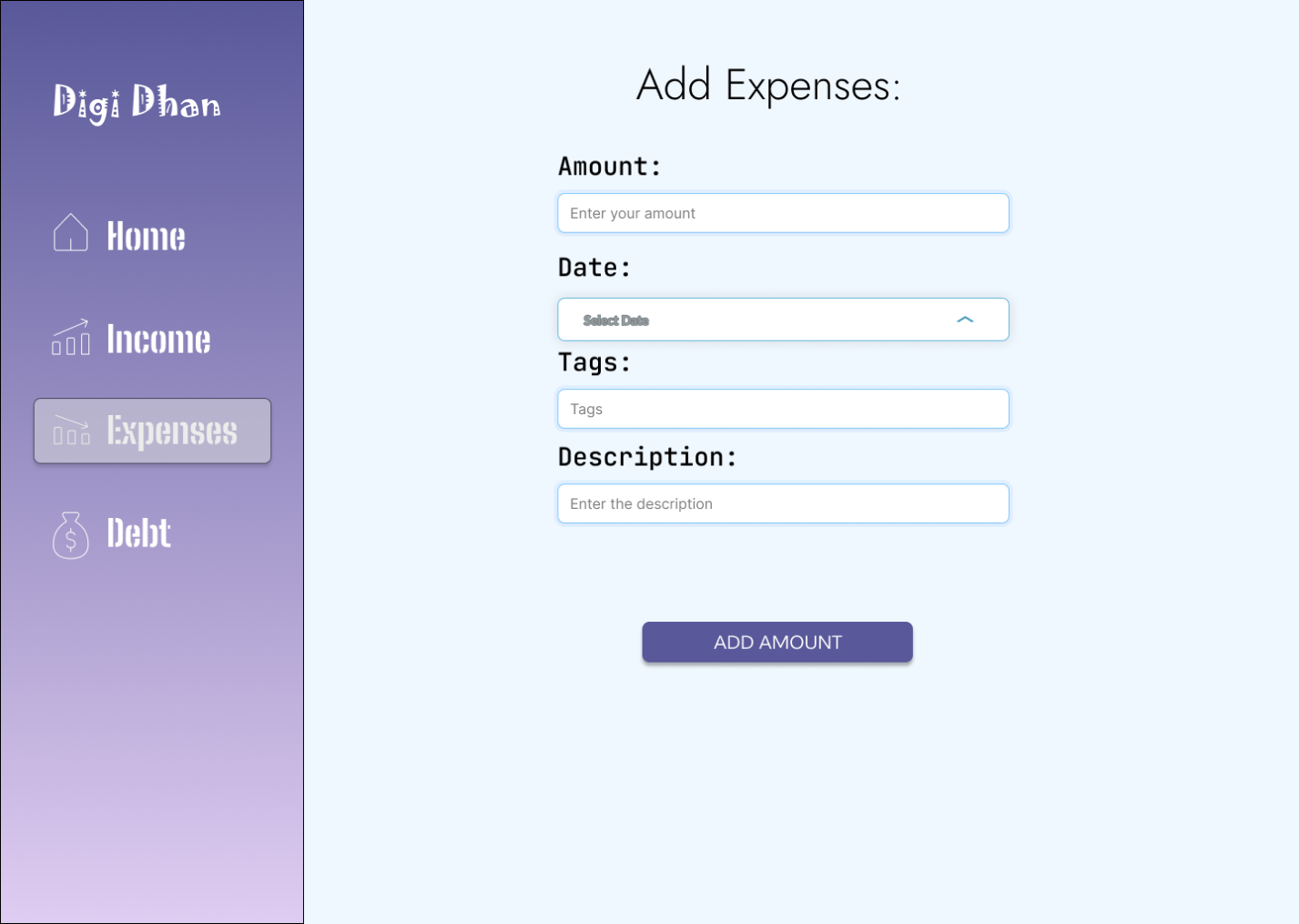


Figure : UI for expenses page

### 2.1.5 Prototype of Debt page:

The UI for debt page of Digi Dhan is given below:

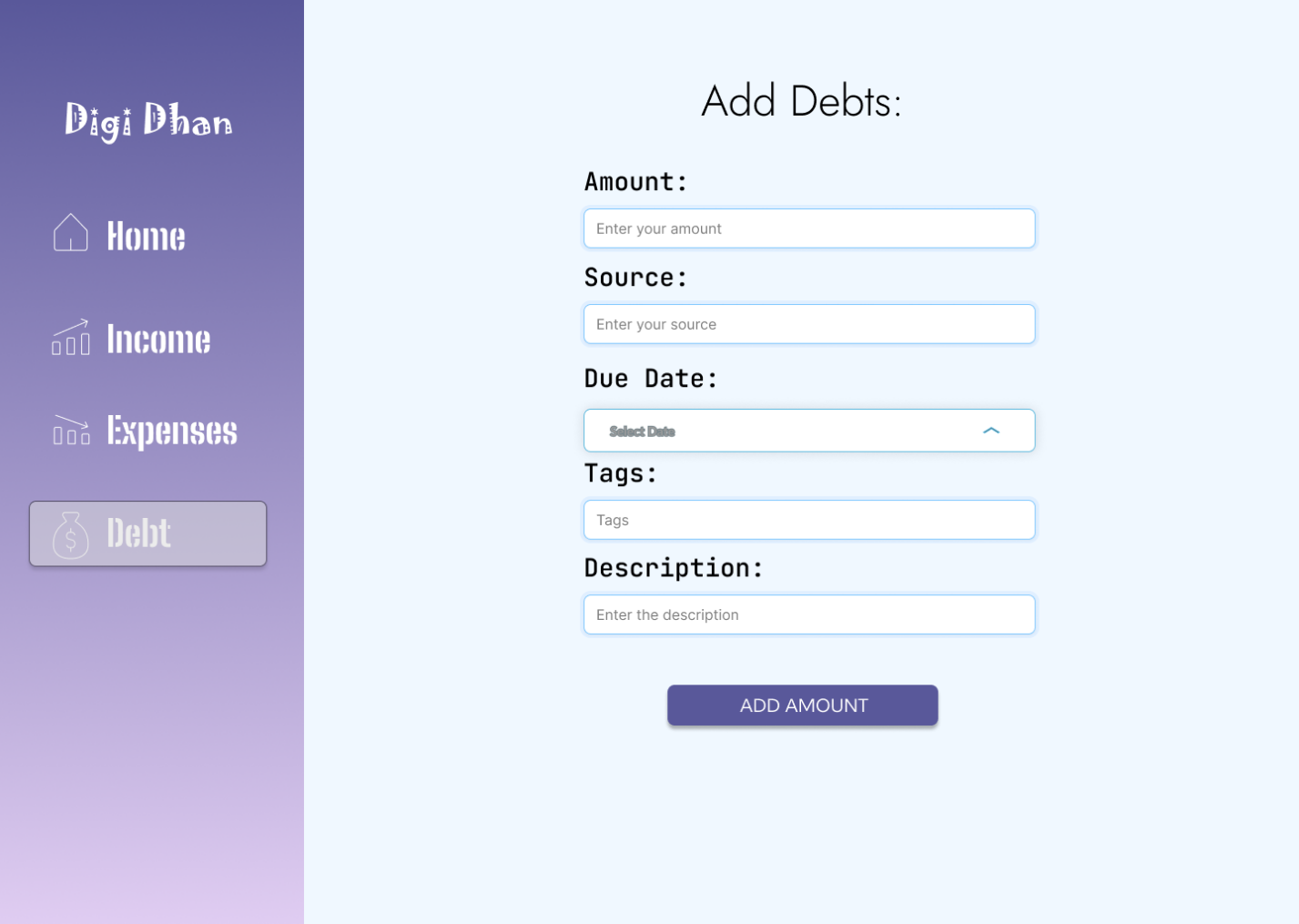
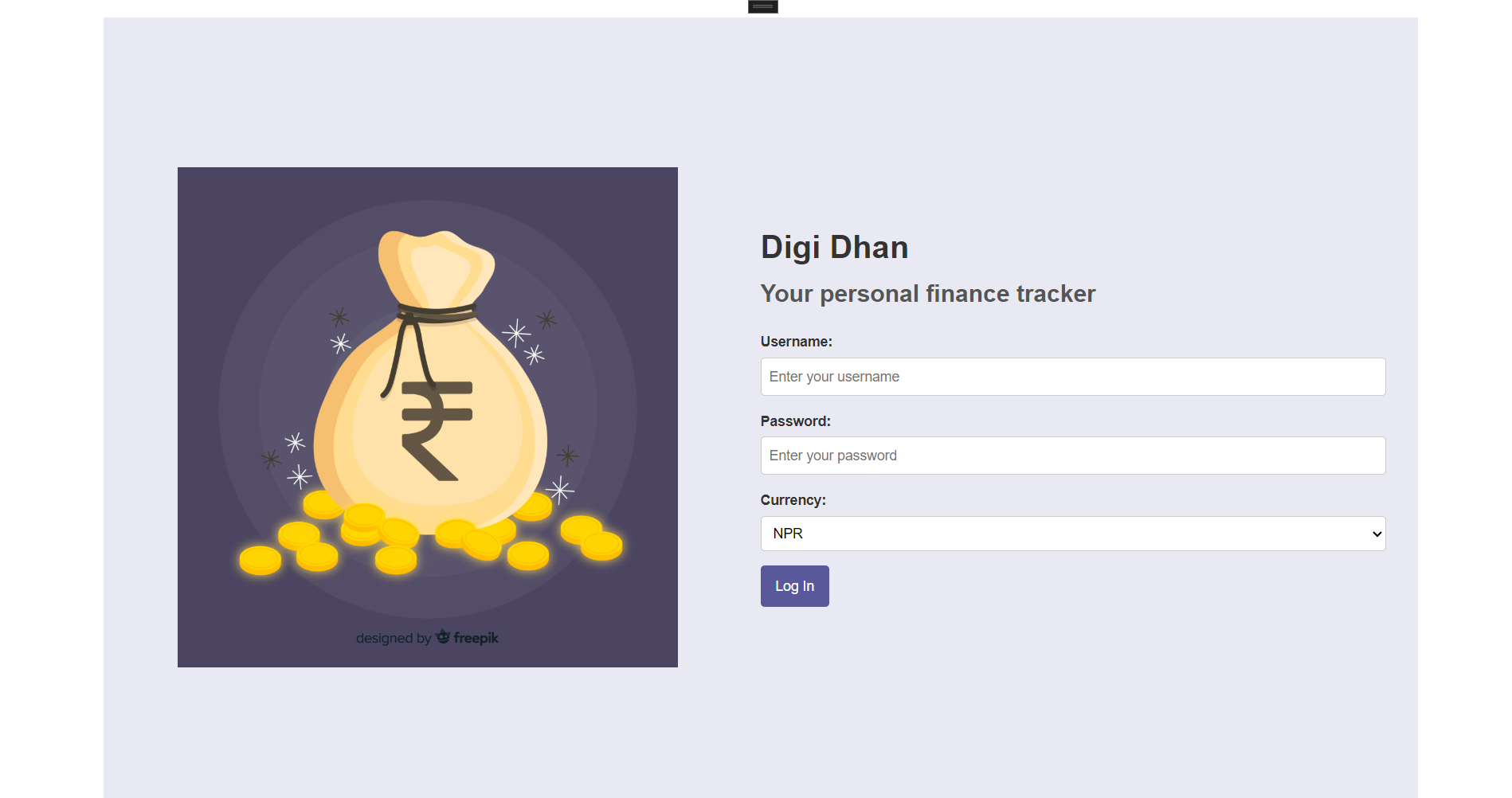


Figure : UI for debt page.

## 2.2 Developed Application:

The developed application ‘DigiDhan’ consists of a total of 6 pages namely: Login page, Home, Add Income, Add Expenses, Add Debt and Create Tag page. The description for each page along with their workings is given below:

### 2.2.1 Login page:



This is the first page that is loaded at the start of the application. From this page, the user can log in to the application using the correct credentials. Upon successful login, the user is redirected to the home page.

### 2.2.2 Home page:

Upon successful login, the user lands on the home page. The home consists of a number of features and components. All the performed transactions are shown in a table in the home page. Sorting of the transactions by date, searching for transactions based on specific date range, searching for transactions by title and filtering can also be done. Similarly, the current user balance along with the highest, lowest and total amount of income, expense and debt is also displayed on home page. The pending and cleared debts are also highlighted in home page.

### 2.2.3 Income page:

Upon clicking on income on the side navbar, the user is directed to the add income page. The users can add the income from this page providing various details such as amount, source, date, tags, note and type. Before adding income, pending debts are checked. Incase of pending debt, the added income is used to clear the pending debt.

### 2.2.4 Expense page:

The expense page is used to add any expenses made by the user. The page contains input field for amount, source, date, tags, note and type. The expenses made is deducted form the balance of user. Incase of insufficient balance, an appropriate message is shown.

### 2.2.5 Debt page:

The debt page is used to handle the debt transactions. The debt page contains input field for amount, source, date, due date, tags, note and type. The pending debts is added to the current balance of the user. The pending debts are cleared from income.

### 2.2.6 Tags page:

The tag page is used to create custom tags.

## 2.3 Test cases:

The test cases for each feature and functionalities of the developed application Digi Dhan are given below:

### 2.3.1 Test 1: Logging in

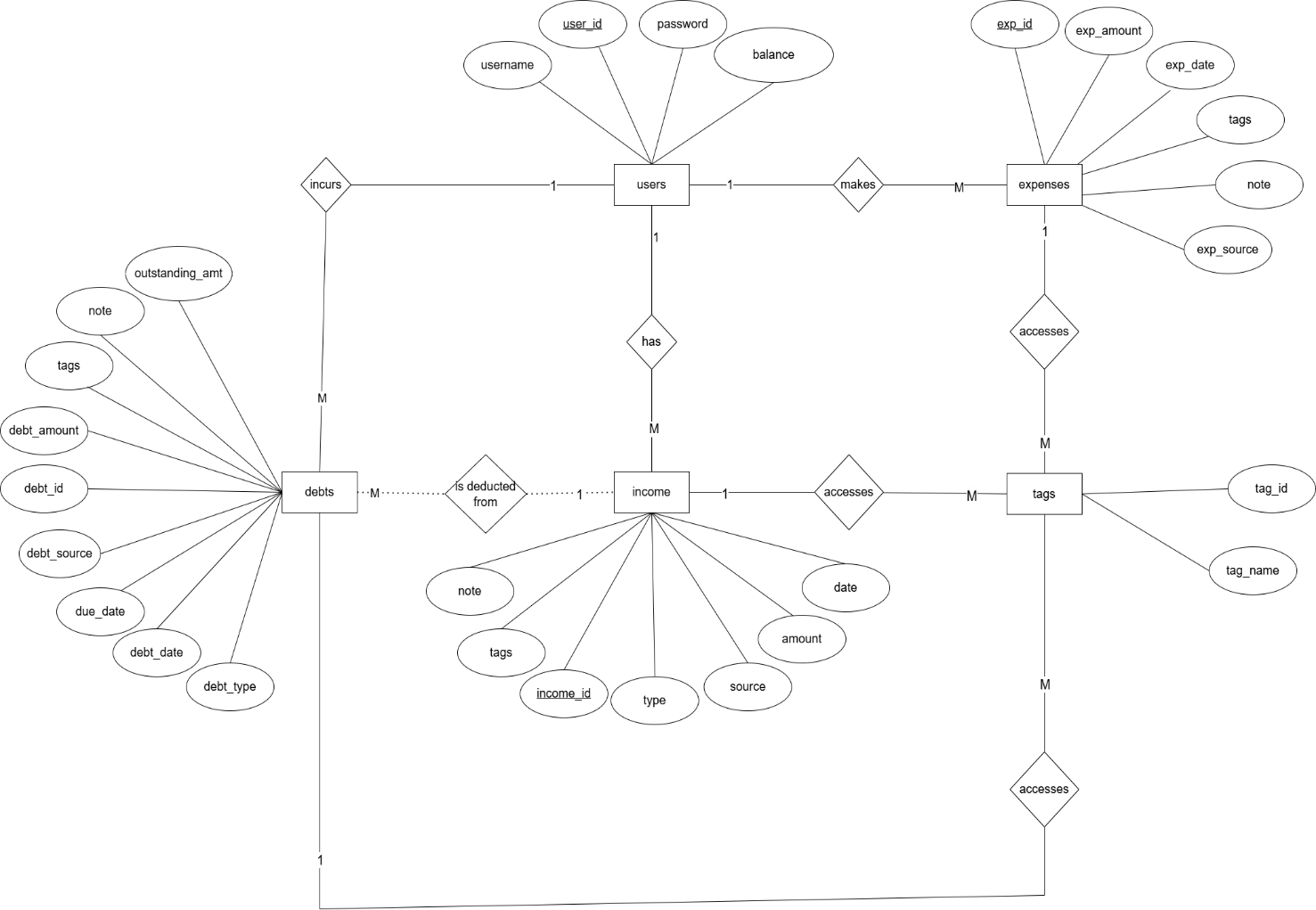
# Development and Logical solution:

In this section, the logical solution for the application such as Entity Relationship Diagram, flowchart, use case as well as class, properties and their method Is given.

## 3.1 Entity Relationship Diagram:

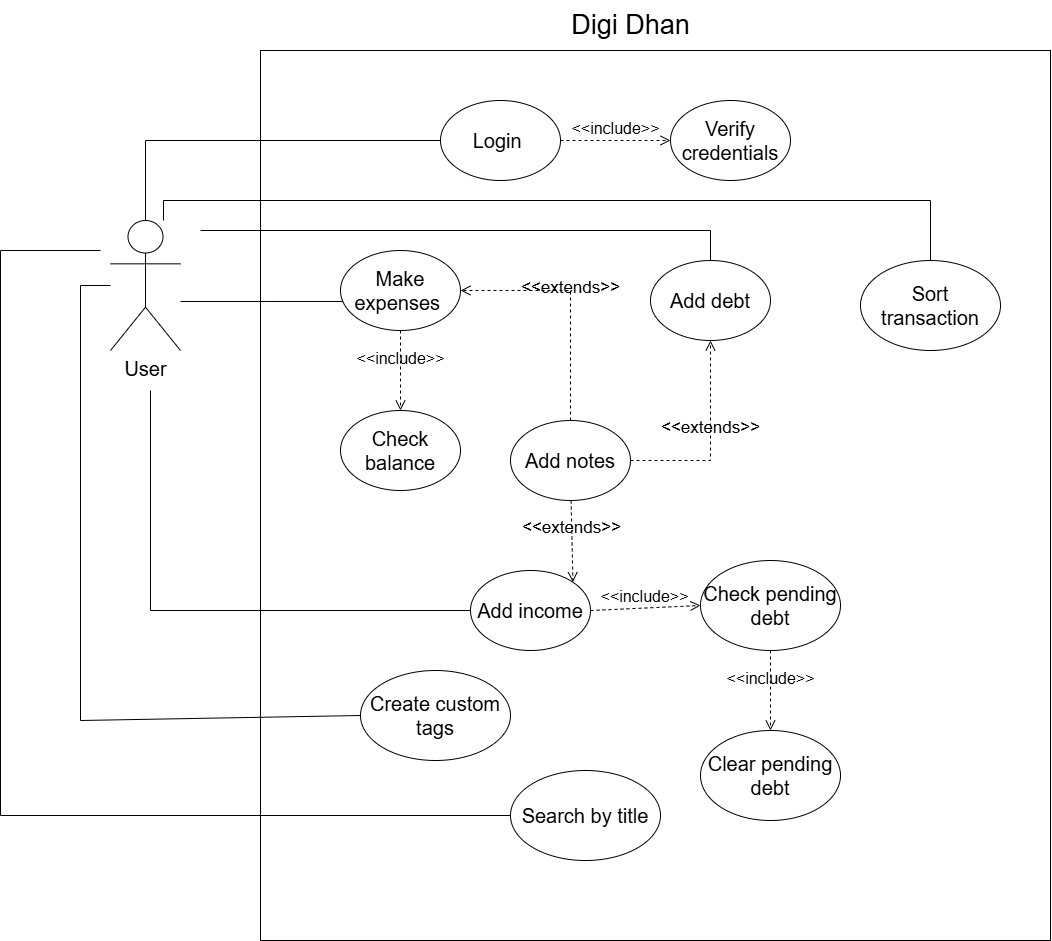
The journal states, “The Entity-Relationship diagram has been widely used in structured analysis and conceptual modeling.”. It is implied to be easy to understand, powerful to model real-world problems and readily translated into a database schema. Similarly, the researcher states, “The ERD views that the real world consists of a collection of business entities, the relationships between them and the attributes used to describe them” (Il-Yeol Song, 1995).

The entity relationship diagram for my project Digi Dhan is given below:



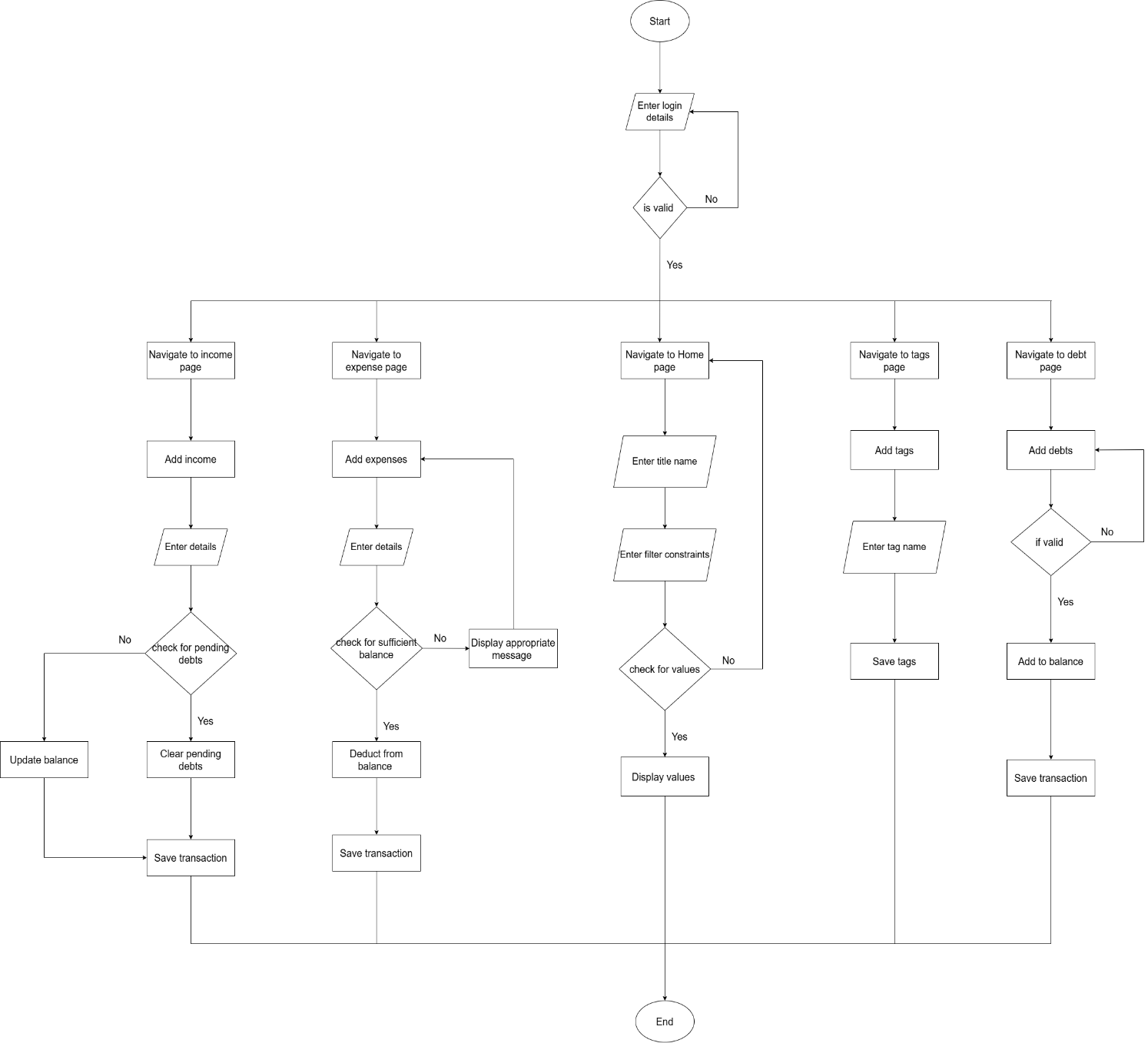
## 3.2 Use Case Diagram:

The use case diagram for my application Digi Dhan is given below:



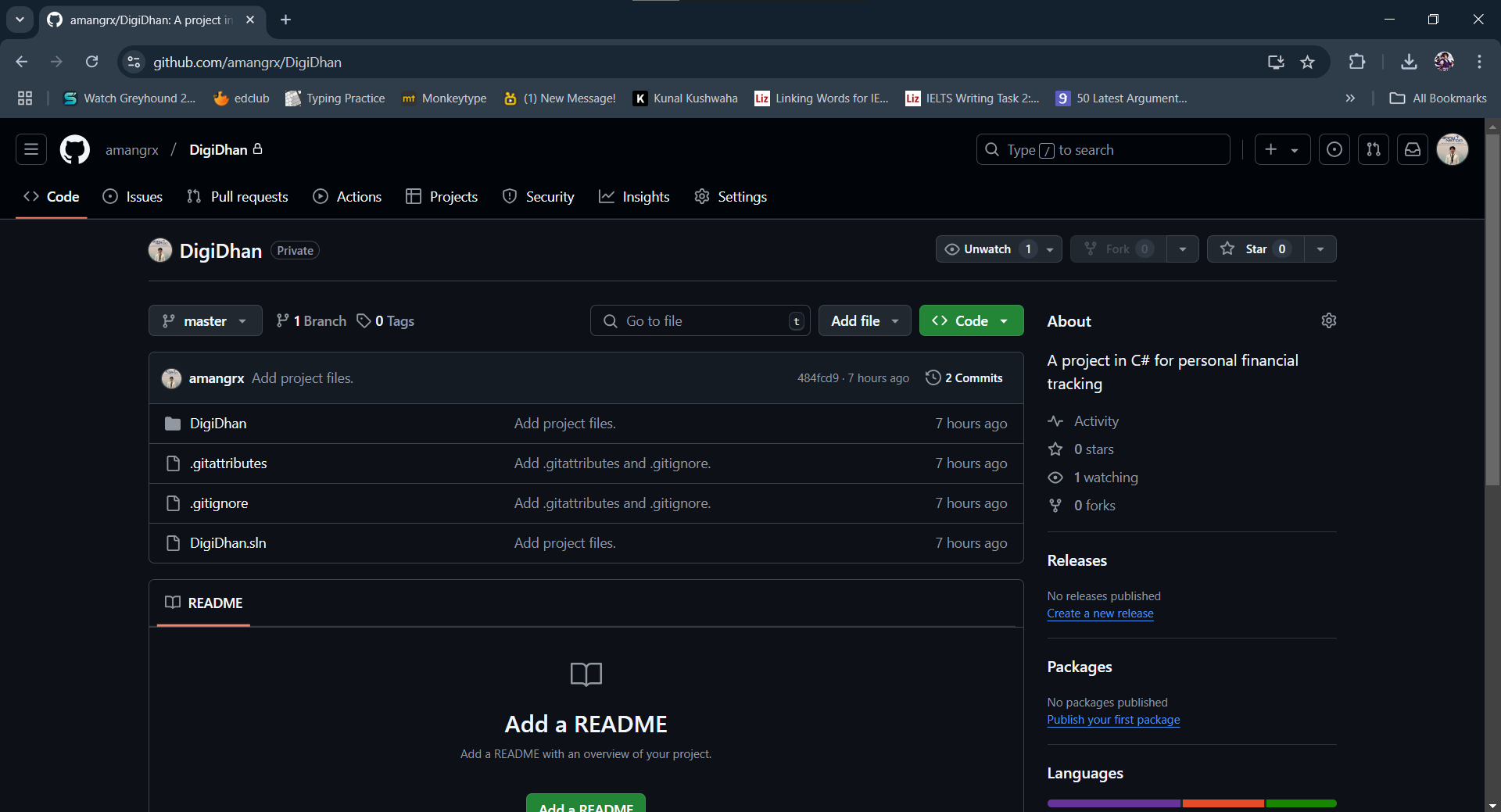
## 3.3 Flowchart:

The flowchart showing the workflow of my project Digi Dhan is given below:



## 3.4 Version Control (Git):

The version control for the project is done with the help of GitHub. A private repository with the name of the application is created in GitHub.



Similarly, the screenshot for the commits of the repository is given below:

## 3.5 Technology stack:

The set of technologies to use in the development of Digi Dhan are as follows:

### 3.5.1 Framework:

Digi Dhan is created using MAUI Blazor Hybrid. .NET MAUI Blazor is a cross-platform framework capable of building hybrid apps.

### 3.5.2 External Libraries:

Some of the external libraries that will be used to develop the application are as follows:

* MudBlazor
* SQLite-net

### 3.5.3 Persistence Mechanism:

For persistence mechanism, local database using SQLite will be applied in the development of applications.

## 3.6 Class, properties and methods:

### 3.6.1 Classes and their properties:

The classes prevalent in my project along with their properties is given below:

# 4 Conclusion:

Digi Dhan has been completed with the specified features and functionalities. The completion of this project brings forth many learning things and experiences. Being completely new to C#, I had relatively no idea on how to develop this application and complete this coursework, however over the period of time, a constant amount of my time and effort was spent on completing this project.

This period of time consisted of numerous trials and errors, research as well as learning from the module leader. While developing the application, I encountered a number of problems such as during developing the sort by specific date range method, the values were not being displayed due to mismatch of format for date. Similarly, at first, I had no idea on how to connect the application with SQLite. After a through research and consultation with my friends, I was able to complete this task as well. I also gained familiarity in working with Visual Studio 2022 as the whole application is developed in Visual Studio. Various concepts of OOP such as overriding, overloading etc. was also used during the development of the application.

Overall, this coursework has been a very fruitful one for me. The completion of this coursework came with its own sense of huge relief. I have learned a number of skills at the completion of this project. From research skills to gaining proficiency in C#, this project has helped me develop skills and knowledge. These gained knowledge and skill will be useful in my future endeavors.

# Bibliography

Il-Yeol Song, M. E. ,. E. P., 1995. A Comparative Analysis of Entity-Relationship Diagrams1. *Journal of Computer and Software Engineering,* Volume 3, pp. 427-459.