Name: Samooh Moosa

Student ID: S2000773

Module Name: Digital Systems Project

Module Code: UFCFXK-30-3\_May 2022

Word Count: 9922/8000 – 10000

Prototype for Budget Management System

# Abstract

This project aims to create a cutting-edge budget management system that addresses the challenges of outdated bookkeeping practices. Through extensive research on the pain points of managing finances, the project will result in a software solution that simplifies budgeting and reporting, with a strong emphasis on automation and ease-of-use.

# Author

Samooh Moosa

[S2000773@students.villacollege.edu.mv](mailto:S2000773@students.villacollege.edu.mv)

# Table of Contents

[Abstract 2](#_Toc125839018)

[Author 2](#_Toc125839019)

[Table of Contents 2](#_Toc125839020)

[Table of Figures 5](#_Toc125839021)

[Chapter 1 – Introduction (350) 5](#_Toc125839022)

[1.1: Explanations of Chapters 6](#_Toc125839023)

[Chapter 2 – Literature Review (3000) 6](#_Toc125839024)

[2.1: An overview of budget management systems and challenges of traditional bookkeeping 6](#_Toc125839025)

[2.1.1 Introduction to budget management systems 6](#_Toc125839026)

[2.1.2 Challenges of traditional bookkeeping methods 6](#_Toc125839027)

[2.1.3 Automation of traditional bookkeeping tasks 7](#_Toc125839028)

[2.2 Analysis of existing budget management software 7](#_Toc125839029)

[2.2.1 Mint 7](#_Toc125839030)

[2.2.2 YNAB 7](#_Toc125839031)

[2.2.3 PocketGuard 7](#_Toc125839032)

[2.2.4 Quicken 8](#_Toc125839033)

[2.2.5 Moneydance 8](#_Toc125839034)

[2.3 Latest trends and advancements in budget management systems 8](#_Toc125839035)

[2.3.1 Mobile Budget Management Apps 8](#_Toc125839036)

[2.3.2 Integration with other financial tools 8](#_Toc125839037)

[2.3.3 Artificial Intelligence & Machine Learning 8](#_Toc125839038)

[2.3.4 Cloud-based Systems 9](#_Toc125839039)

[2.3.5 Blockchain Technology 9](#_Toc125839040)

[2.4.1 Improved Efficiency 9](#_Toc125839041)

[2.4.2 Improved Accuracy 9](#_Toc125839042)

[2.4.3 Financial Insights 9](#_Toc125839043)

[2.5. User Adoption and User-centered Design 10](#_Toc125839044)

[2.5.1 Challenges of User Adoption 10](#_Toc125839045)

[2.5.2 User-centered Design Approach 10](#_Toc125839046)

[2.5.2.1 User Research 10](#_Toc125839047)

[2.5.2.2 Usability Testing 10](#_Toc125839048)

[2.5.2.3 Prototyping 10](#_Toc125839049)

[2.6. Accounting Standards and Regulations 10](#_Toc125839050)

[2.6.1 Importance of compliance with accounting standards and regulations 10](#_Toc125839051)

[2.6.2 Relevant accounting standards and regulations for budget management 11](#_Toc125839052)

[2.6.3 Role of budget management systems in ensuring compliance 11](#_Toc125839053)

[2.6.4 Automation of compliance tasks and identification of non-compliance 11](#_Toc125839054)

[2.6.5 Importance of consulting with an accountant or financial advisor 11](#_Toc125839055)

[2.7 Research Gap and Problem Area 11](#_Toc125839056)

[2.7.1 Research Gap 11](#_Toc125839057)

[2.7.2 Problem Area 11](#_Toc125839058)

[2.7.3 Project Objectives 12](#_Toc125839059)

[2.8 Summary of Main Findings 12](#_Toc125839060)

[2.8.1 Key Issues to be Addressed 12](#_Toc125839061)

[2.8.2 Limitations 13](#_Toc125839062)

[Chapter 3 – Requirement Analysis (800) 13](#_Toc125839063)

[3.1 Problem Description and Project Objectives 13](#_Toc125839064)

[3.2 System Scope and Intended Users 13](#_Toc125839065)

[3.3 Functional Requirements 13](#_Toc125839066)

[3.4 Non-functional Requirements 15](#_Toc125839067)

[3.5 Trade-offs 16](#_Toc125839068)

[3.6 Summary of Findings and Key Issues 16](#_Toc125839069)

[3.7 Assumptions and Constraints 17](#_Toc125839070)

[3.8 Testing and Validation Methods 17](#_Toc125839071)

[3.8.1 Unit Testing 17](#_Toc125839072)

[3.8.2 Integration Testing 17](#_Toc125839073)

[3.8.3 Functional Testing 17](#_Toc125839074)

[Chapter 4 – Planning & Management (800) 18](#_Toc125839075)

[4.1 Project Management Methodology 18](#_Toc125839076)

[4.2 Project Charter 18](#_Toc125839077)

[4.3 Risk Management 18](#_Toc125839078)

[4.4 Personal Role and Management 18](#_Toc125839079)

[Chapter 5 – Methodologies (800) 19](#_Toc125839080)

[4.1 Research Methods 19](#_Toc125839081)

[4.2 Software Development Methodology 19](#_Toc125839082)

[4.3 Design Methodologies 19](#_Toc125839083)

[4.4 Testing Methodologies 19](#_Toc125839084)

[4.5 Project Management Methodology 19](#_Toc125839085)

[4.6 Challenges and Findings 19](#_Toc125839086)

[Chapter 6 – Design (2000) 20](#_Toc125839087)

[6.1 UI & UX Design 20](#_Toc125839088)

[6.1.1 User Interface Design 20](#_Toc125839089)

[6.1.2 User Experience Design 20](#_Toc125839090)

[6.1.3 Design of Some Pages 21](#_Toc125839091)

[6.2 Design Principles and Guidelines 21](#_Toc125839092)

[6.3 Data Model and Database Design 23](#_Toc125839093)

[6.4 Use Case Diagram 24](#_Toc125839094)

[6.5 Sequence Diagram 24](#_Toc125839095)

[Chapter 7 – Implementation (600) 25](#_Toc125839096)

[7.1 Tools and Frameworks Used in Implementation 25](#_Toc125839097)

[7.2 Version Control 25](#_Toc125839098)

[7.3 Sign Up 26](#_Toc125839099)

[7.4 Log In 27](#_Toc125839100)

[7.5 Dashboard 28](#_Toc125839101)

[7.6 Bills & Payments 28](#_Toc125839102)

[7.7 Incomes & Expenses 29](#_Toc125839103)

[7.8 My Profile 30](#_Toc125839104)

[7.9 Future Updates 31](#_Toc125839105)

[Chapter 8 – Testing (600) 32](#_Toc125839106)

[8.1 Functional Requirement Test Cases 32](#_Toc125839107)

[Chapter 9 – Evaluation (400) 35](#_Toc125839108)

[Chapter 10 – Conclusion (400) 36](#_Toc125839109)

[References 36](#_Toc125839110)

[12: Appendix A: Database 39](#_Toc125839111)

[12.1 SQL Table Creation Script 39](#_Toc125839112)

[12.2 Database Tables 45](#_Toc125839113)

[13: Appendix B: Diagrams 49](#_Toc125839114)

[13.1 ER Diagram 49](#_Toc125839115)

[13.2 Use Case Diagram 50](#_Toc125839116)

# Table of Figures

[Figure 1 Sign-In Screen 22](#_Toc125839117)

[Figure 2 Home Page (Dashboard) 22](#_Toc125839118)

[Figure 3 Icons for Pages 23](#_Toc125839119)

[Figure 4 Search Button 23](#_Toc125839120)

[Figure 5 Sign-Up Screen (Prototype) 26](#_Toc125839121)

[Figure 6 Sign in Screen (Prototype) 27](#_Toc125839122)

[Figure 7 Dashboard 28](#_Toc125839123)

[Figure 8 Bills & Payments Screen 29](#_Toc125839124)

[Figure 9 Expenses Table with Headings 29](#_Toc125839125)

[Figure 10 Expenses Table with Add New Button 30](#_Toc125839126)

[Figure 11 My Profile Page 30](#_Toc125839127)

[Figure 12 Logout button 35](#_Toc125839128)

[Figure 13 Logout servlet 35](#_Toc125839129)

[Figure 14 Accounts Table 45](#_Toc125839130)

[Figure 15 Bills Table 45](#_Toc125839131)

[Figure 16 Budgets Table 45](#_Toc125839132)

[Figure 17 Categories Table 45](#_Toc125839133)

[Figure 18 Countries Table 46](#_Toc125839134)

[Figure 19 Currencies Table 46](#_Toc125839135)

[Figure 20 Recurring Frequency Table 46](#_Toc125839136)

[Figure 21 Reports Table 46](#_Toc125839137)

[Figure 22 Report Type Table 47](#_Toc125839138)

[Figure 23 Savings Table 47](#_Toc125839139)

[Figure 24 Settings Table 47](#_Toc125839140)

[Figure 25 Themes Table 47](#_Toc125839141)

[Figure 26 Transactions Table 48](#_Toc125839142)

[Figure 27 Transaction Types Table 48](#_Toc125839143)

[Figure 28 Users Table 48](#_Toc125839144)

[Figure 29 User Logs Table 49](#_Toc125839145)

[Figure 30 ER Diagram (Complete) 49](#_Toc125839146)

[Figure 31 Use Case Diagram 50](#_Toc125839147)

[Figure 32 User Registration & Login Sequence Diagram 51](#_Toc125839148)

# Chapter 1 – Introduction

In our highly productive and fast-paced world, traditional bookkeeping methods have lost efficiency due to time-consuming practices, manual errors, and repetitive tasks (Bote, 2021). The clarity of transactions entered, the audit process, and the budgeting process are all extremely time-sensitive matters that can be improved and redesigned for greater productivity. Thus, the following software prototype has been designed with a high range of versatility which can be customized to be used as a personal expense tracker or as an application that helps settle bills and payments that need to be done regularly.

The application is not tailored to the needs of a particular society or a country as it is designed for the general society. In addition, it is meant to reduce the time-consuming tasks mentioned earlier through automation whilst still following the universal practices of accounting. The software aims at being an easy to use and enter data platform, real-time statistics generating and standardizes the financial data to accounting standards for personals. Hence, in this report it will be detailed on what were the literature gathered to identify the importance of the need for such a software, the methodologies used to construct the application with details of the platforms and other software bases used along with the implementation or a pilot study and an evaluation of the results before concluding.

## 1.1: Explanations of Chapters

Chapter 2: Literature review of budget management systems and software tools

Chapter 3: Functional and non-functional requirements

Chapter 4: Project planning and management strategies

Chapter 5: Methodology and rationale

Chapter 6: Design and database of the budget management system

Chapter 7: Implementation process

Chapter 8: Testing results

Chapter 9: Evaluation of project effectiveness and potential for improvement.

# Chapter 2 – Literature Review

## 2.1: An overview of budget management systems and challenges of traditional bookkeeping

### 2.1.1 Introduction to budget management systems

Budget management systems are software applications that help individuals and businesses manage their financial resources more efficiently (Horton, 2022). They are designed to automate many of the tasks associated with traditional bookkeeping methods, such as expense tracking, bill payment, and budget planning (Parletta, 2022). These systems are increasingly popular in today's fast-paced and highly productive world, as they can help users save time and reduce the risk of errors associated with manual bookkeeping (Chrisos, 2020).

### 2.1.2 Challenges of traditional bookkeeping methods

Despite the many advantages of budget management systems, there are still some challenges that need to be addressed. One of the main challenges faced by traditional bookkeeping methods is their time-consuming nature (Shpak, 2019). Many people find it difficult to maintain accurate records of their expenses and income, especially if they have multiple sources of income or a large number of expenses. This can lead to errors and inconsistencies in the financial data, making it difficult to make informed decisions about budgeting and spending (SunAccounts, 2022). Another major challenge faced by traditional bookkeeping methods is the lack of transparency and standardization (Quinn, 2017). Many people find it difficult to understand the financial data that they are presented with, which can make it hard to analyze and interpret the data. Additionally, there can be a lack of standardization in financial data, making it hard for users to compare and contrast different financial statements (DeBenedetti, 2016).

### 2.1.3 Automation of traditional bookkeeping tasks

In order to overcome these challenges, budget management systems are designed to automate many of the tasks associated with traditional bookkeeping methods (Trigo, Belfo and Estébanez, 2016). They can help users save time by automating the process of expense tracking, bill payment, and budget planning (Castellano, 1998). They can also provide users with real-time statistics and reports that can help them make more informed decisions about budgeting and spending.

Additionally, budget management systems similar to any other application, can be tailored to the needs of different users and can be customized to suit different accounting standards (ScienceSoft, 2021). This allows the users to have a system that is tailored to their specific needs and requirements, making it more efficient and user-friendly.

Now that we have looked into what a budget management system is and its uses let’s look into some of the existing budget management software in the market.

## 2.2 Analysis of existing budget management software

There are a variety of budget management software options available on the market, each with their own unique features and capabilities. Some popular options include Mint, YNAB (You Need A Budget), PocketGuard, Quicken and Moneydance.

### 2.2.1 Mint

Mint is a budget management software that offers a wide range of features, including expense tracking, budgeting, and bill payment (Intuit, 2020). One of its strengths is its ability to automatically import financial data from various accounts, such as bank accounts and credit cards. This makes it easy for users to get an overview of their financial situation in one place. Additionally, Mint offers a variety of financial tools such as credit score monitoring and personalized financial advice, making it a comprehensive financial management tool (Intuit, 2019). However, one of its weaknesses is that it does not have a feature for tracking income, which could be a limitation for some users.

### 2.2.2 YNAB

YNAB, also known as You Need A Budget, is a budget management software designed to aid users in creating and sticking to a budget through real-time feedback on spending habits. It also offers the ability to plan for unexpected expenses and income fluctuations, making it a versatile tool for financial management (YNAB, 2008). However, it lacks the bill payment and account syncing features found in other popular budget management software like Mint.

### 2.2.3 PocketGuard

PocketGuard is a budget management software that primarily focuses on expense tracking and budgeting (PocketGuard, 2022). It offers the ability to give users a comprehensive view of their spending habits through expense categorization and personalized insights. It also allows users to set and monitor savings goals, making it an effective tool for both short-term and long-term financial management. However, similar to YNAB PocketGuard also lacks a bill pay feature which could be inconvenient for some users.

### 2.2.4 Quicken

Quicken is a budget management software that offers a wide range of features, including expense tracking, budgeting, and bill payment. One of its strengths is its ability to automatically import financial data from various accounts, such as bank accounts and credit cards similar to that of Mint. In addition to that, Quicken also offers a variety of financial tools such as investment tracking and retirement planning which are not common in other apps in this category, making it a comprehensive financial management tool (Quicken, 2018). However, one of its weaknesses is that it's not available on mobile devices, which could be a limitation for some users as most people today uses their phones to do all their daily tasks.

### 2.2.5 Moneydance

Moneydance is a budget management software that focuses on budgeting and expense tracking. One of its strengths is its ability to help users create a budget and stick to it by providing real-time feedback on their spending habits. Additionally, similar to YNAB, Moneydance also has a feature that allows users to plan for unexpected expenses and irregular income, making it a flexible tool for managing finances (MoneyDance, 2008). However, one of its weaknesses is that it does not offer bill payment feature like Mint and Quicken do.

Now that existing budget management applications have been covered, it's clear that there is a wide range of this kind of applications available on the market, each with their own unique set of features and capabilities. From Mint to YNAB, PocketGuard to Quicken and Moneydance, all have their own strengths and weaknesses. Some software options offer comprehensive financial management tools, while others focus on budgeting and expense tracking. Additionally, there are other software options that have limitations like no mobile app or no bill payment feature. It's important to note that there is no perfect software, and the best option for a user will depend on their individual needs and requirements.

## 2.3 Latest trends and advancements in budget management systems

### 2.3.1 Mobile Budget Management Apps

The field of budget management systems has seen a number of advancements and trends in recent years, with one of the most notable being the rise in popularity of mobile budget management apps. According to a study by AppRadar, there was a 46% increase in the popularity of money management apps on Android in the first half of 2022 (Freer, 2022). These apps offer users the convenience of tracking expenses and managing finances on-the-go, through their smartphones or tablets.

### 2.3.2 Integration with other financial tools

Another trend is the integration of budget management systems with other financial tools, such as investment tracking and retirement planning. This allows users to have a more comprehensive view of their financial situation and make better-informed decisions (Dziak, 2021).

### 2.3.3 Artificial Intelligence & Machine Learning

Artificial intelligence and machine learning are becoming more prevalent in budget management systems. These technologies allow budget management systems to automatically categorize expenses, provide personalized financial advice, and even predict future expenses (Paro, 2021). This can help users make better-informed financial decisions and reach their savings goals more quickly.

### 2.3.4 Cloud-based Systems

Cloud-based budget management systems are also on the rise, as they allow users to access their financial information from any device with internet access. Additionally, they provide an added layer of security as data is stored on remote servers and can be easily backed up (Centage, 2015).

### 2.3.5 Blockchain Technology

Blockchain technology is also being explored as a way to improve budget management systems. This technology can provide a secure and transparent way to track financial transactions, which could be beneficial for businesses and individuals alike (IBM, 2021).

The above-mentioned advancements are only some in the long list of developments that were seen in the last few years. These developments are aimed at making budget management more convenient, secure, and efficient for users. It's important to note that while this current app doesn't have these features yet, plans for inclusion of features such as cloud synchronization, mobile application, and AI & machine learning are in the pipeline for the full application release.

2.4. The Advantages of Budget Management Systems

### 2.4.1 Improved Efficiency

Budget management systems have a range of benefits for both individuals and businesses. One of the most significant benefits is improved efficiency (SunAccounts, 2021). By automating repetitive tasks such as data entry and tracking expenses, budget management systems can save users a significant amount of time (Data Stems, 2021). This can be especially beneficial for businesses, where time is often a scarce resource.

### 2.4.2 Improved Accuracy

Another key benefit of budget management systems is improved accuracy (Indeed Editorial Team, 2022). By providing real-time feedback and standardizing financial data, budget management systems can help users identify and correct errors in their financial records (Davis, 2022). This can be especially beneficial for businesses, where accurate financial records are essential for compliance with accounting standards and tax laws.

### 2.4.3 Financial Insights

In addition to improved efficiency and accuracy, budget management systems can also provide users with valuable financial insights. By providing detailed reports and analytics, budget management systems can help users identify patterns in their spending and identify areas where they may be able to save money (Volopay, 2022). This can be especially beneficial for individuals, who may not have the same resources and expertise as businesses when it comes to financial management.

Overall, budget management systems can provide a wide range of benefits for both individuals and businesses, including improved efficiency, accuracy, and financial insights. As we have seen from the analysis of existing budget management software, a comprehensive budget management software that addresses the user's needs and incorporates the latest advancements and technology can be a valuable tool for managing personal and business finances. Next, we are going to look into an important feature of any application that helps in determining its success.

## 2.5. User Adoption and User-centered Design

The development of a budget management system is a significant investment, and its success like any other type of application depends on its ability to be adopted and used by its intended audience (Walburg, 2020).

### 2.5.1 Challenges of User Adoption

User adoption can be a significant challenge for budget management systems, as users may be resistant to change, have a lack of understanding of the system, or have difficulty using the system (NTT, 2019). These challenges can be particularly pronounced for certain individuals, such as those who may have limited experience or proficiency with online applications, or those with disabilities or aging populations.

### 2.5.2 User-centered Design Approach

To overcome these challenges, it is essential to adopt a user-centered design approach. User-centered design is a design approach that focuses on the needs, wants, and limitations of the users of a system (Interaction Design Foundation, 2017). By taking a user-centered design approach, developers can ensure that the budget management system is designed with the user in mind, making it more likely to be adopted and used.

#### 2.5.2.1 User Research

User-centered design can include a range of techniques, such as user research, usability testing, and prototyping. By conducting user research, developers can gain a deeper understanding of the needs and wants of their users.

#### 2.5.2.2 Usability Testing

Through usability testing, developers can identify and address any issues with the system that may be preventing users from adopting it.

#### 2.5.2.3 Prototyping

And by prototyping, developers can create a low-fidelity version of the system to test and validate the design (Novoseltseva, 2017).

In summary, user adoption and user-centered design are critical factors in the success of budget management systems. By taking a user-centered design approach, developers can ensure that the system is designed with the user in mind and is more likely to be adopted and used by its intended audience. However, even if these factors are included there is still one more thing that a budget management application needs to have in order to become successful.

## 2.6. Accounting Standards and Regulations

### 2.6.1 Importance of compliance with accounting standards and regulations

Budget management is closely related to accounting, and compliance with accounting standards and regulations is an important aspect of budget management (Multiview, 2022). Accounting standards and regulations provide a framework for financial reporting, and they ensure that financial information is presented in a consistent and comparable manner (Kenton, 2022).

### 2.6.2 Relevant accounting standards and regulations for budget management

There are several accounting standards and regulations that are relevant to budget management, including Generally Accepted Accounting Principles (GAAP) and International Financial Reporting Standards (IFRS) (Kenton, 2022).

### 2.6.3 Role of budget management systems in ensuring compliance

In terms of budget management, compliance with accounting standards and regulations is essential for businesses to ensure that their financial records are accurate and reliable. Budget management systems can play a crucial role in compliance by providing real-time financial information, ensuring consistency in financial reporting, and providing an accurate record of financial transactions (CFI Team, 2022).

### 2.6.4 Automation of compliance tasks and identification of non-compliance

Additionally, budget management systems can provide businesses or individuals with the ability to automate compliance tasks, such as the generation of financial reports, which can save time and reduce the risk of errors. Furthermore, budget management systems can also help businesses to identify and address areas where they may be non-compliant with accounting standards and regulations. For example, a budget management system may reveal that a business is not accurately reporting its expenses, or that it is not keeping accurate records of its financial transactions. By identifying these issues, a budget management system can help businesses to take corrective action and become compliant with accounting standards and regulations.

### 2.6.5 Importance of consulting with an accountant or financial advisor

Compliance with accounting standards and regulations is an important aspect of budget management and budget management systems can play a crucial role in ensuring compliance. The systems can provide real-time financial information, automate compliance tasks, and help identify and address areas where a business may be non-compliant. It is important to consult with an accountant or a financial advisor to understand the specific accounting regulations and standards that apply to the business or individuals and how a budget management system can assist in meeting those requirements.

## 2.7 Research Gap and Problem Area

### 2.7.1 Research Gap

The literature review has shown that budget management is an important aspect of financial management for individuals. However, it also revealed that there are certain challenges and limitations associated with traditional bookkeeping methods and existing budget management systems for individuals. These include time-consuming practices, manual errors, and a lack of flexibility in adapting to different needs. Additionally, there is a lack of research on the user adoption of budget management systems for individuals and the role of user-centered design in addressing these challenges.

### 2.7.2 Problem Area

The specific problem area that this project aims to address is the lack of a budget management system that is versatile, easy to use and provides real-time financial insights for individuals which includes features such as bill payments and account syncing, which are not available in traditional bookkeeping methods. However, please note that some of these features are not in the current prototype of the budget management system but will be included in the final version of the system.

### 2.7.3 Project Objectives

This project aims to address the research gap by developing a budget management system that addresses the limitations of traditional bookkeeping methods and existing budget management systems for individuals. Specifically, the project aims to develop a budget management system that is versatile and can be customized to meet the needs of different individuals. The system will be designed to reduce the time-consuming tasks associated with traditional bookkeeping methods and to provide real-time financial insights. Additionally, this project will focus on user-centered design to improve the user adoption of the system and overcome the challenges of user adoption. However, as mentioned before it should be noted that the current prototype of the budget management system does not include advanced features such as cloud synchronization, mobile application and AI & machine learning capabilities and account syncing, which are planned to be included in the final version of the system.

From everything covered so far, it is noticeable that there is a gap in the field and market of budget management for individuals, specifically in the area of versatile, easy to use systems that provide real-time financial insights and include features such as bill payments and account syncing. This project aims to address this gap by developing a budget management system that addresses the limitations of traditional bookkeeping methods and existing budget management systems for individuals.

## 2.8 Summary of Main Findings

### 2.8.1 Key Issues to be Addressed

Based on the literature review, the key issues that need to be addressed in this project include:

1. The limitations of traditional bookkeeping methods and existing budget management systems for individuals.
2. The lack of research on the user adoption of budget management systems for individuals and the role of user-centered design in addressing these challenges.
3. The need for a budget management system that is adaptable and can be customized to meet the needs of different individuals.
4. The need for a system that reduces the time-consuming tasks associated with traditional bookkeeping methods and provides real-time financial insights.
5. The need for a system that includes features such as bill payments and account syncing, which are not available in traditional bookkeeping methods.

In order to address these issues, the project will focus on developing a budget management system that is versatile, easy to use and provides real-time financial insights for individuals, which includes features such as bill payments and account syncing. Additionally, the project will focus on user-centered design to improve the user adoption of the system and overcome the challenges of user adoption.

### 2.8.2 Limitations

Despite the potential benefits of the budget management system being developed in this project, there are several limitations to consider. Firstly, the system is designed for individuals and may not be suitable for small businesses or organizations with more complex financial needs. The system's capabilities may be limited in its ability to handle multiple accounts, multiple users, or more advanced financial transactions such as investments or taxes. Furthermore, while the system can provide financial insights and aid in budget planning, it may not be able to fully replace the need for professional financial advice and guidance.

Additionally, another limitation to consider is that the system is currently in the prototype phase, and additional testing and development will be needed before it can be fully implemented. The prototype will need to be thoroughly tested to ensure that it is reliable, user-friendly, and can handle a wide range of inputs and scenarios. Further user testing will also be required to ensure that the system meets the needs of its intended user base.

Finally, the system will require regular updates and maintenance to ensure that it remains accurate and up-to-date. This includes updating the system to account for changes in financial regulations, changes in technology, and ensuring that all data is secure and protected from unauthorized access. The system will also need to be continuously improved to ensure that it remains relevant and useful to its intended user base. Overall, the budget management system being developed in this project is a useful tool for individuals to manage their finances, but it is important for users to be aware of its limitations and to seek professional financial advice when necessary.

# Chapter 3 – Requirement Analysis

## 3.1 Problem Description and Project Objectives

The budget management system aims to solve the problem of traditional bookkeeping methods being time-consuming and prone to errors for individuals. The objectives of the project are to develop a budget management system that is versatile, easy to use, and provides real-time financial insights. This includes features such as bill payments and account syncing, which are not available in traditional bookkeeping methods. Additionally, the project aims to improve the user adoption of the system by incorporating user-centered design principles.

## 3.2 System Scope and Intended Users

The scope of the system is to provide a budget management system for individuals. The intended users of the system are individuals who want to manage their finances more efficiently and effectively. They will have the ability to track expenses, manage bills and create a budget plan to achieve their financial goals.

## 3.3 Functional Requirements

The functional requirements of the system include:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identifier | Requirement | Rationale | Priority | Success Criteria |
| FR.1 | Allow users to register for the application | The user will be able to register with a username and password | Must | User will be redirected to the homepage, and their provided username and password will be recorded in the database. |
| FR.2 | To be able to log in to the application. | Enables login for registered users. | Must | Users can access their homepage by entering valid login credentials. |
| FR.3 | The option to log out of the application. | The ability to log out of the application. | Must | Upon signing out, the user will be redirected to the login page of the application. |
| FR.4 | Change User details from my profile | The ability to change the user’s information | Should | My profile page should be created from where users can change their details |
| FR.5 | Add Transactions | User will be able to add incomes and expenses for different categories | Should | Successfully adding a transaction to the user's account. |
| FR.6 | Modify Transactions | Users will be able to edit details of transactions recorded | Could | Successfully able to change details of a transaction |
| FR.7 | Hash Password | The password user enters when signing up will be hashed and stored in the database | Must | Successfully able to hash the password and store in the database |
| FR.8 | Calculate Expense, Income & Balance | User is able to see their expense, income & balance | Must | System is able to calculate the transactions for a particular user to get their expense, income and balance |
| FR.9 | Add Bills | User is able to add details of their bills | Could | Successfully add a bill to the list |
| FR.10 | Load users personalized categories on expenses and income | When user goes to expenses or incomes page, categories need to be loaded with their specific ones | Should | Successfully load users personalized categories on Categories dropdown |
| FR.11 | Filter Table (With Date) | User should be able to filter income and expense tables by date | Should | Successfully filter the table with a given date |
| FR.12 | Filter Table (With Category) | User should be able to filter the table with selected category | Should | Successfully filter the table with selected category |
| FR.13 | Delete Transactions | User is able to delete unwanted transactions | Should | Successfully delete a transaction |

## 3.4 Non-functional Requirements

The non-functional requirements of the system include:

|  |  |  |
| --- | --- | --- |
| Identifier | Requirement | Rationale |
| NFR.1 | Web app must be usable 24/7 from anywhere if the user has internet connection once launched | Any person should be able to login provided they are authorised |
| NFR.2 | App has to be responsive enough so that the user doesn’t feel like it is stuck | App has to be responsive |
| NFR.3 | Any change in the data should be recorded in the database as soon as it happens | All logs must be recorded in the database |
| NFR.4 | Multiple users should be able to login to the webapp without it getting slow | Multiple User Access at the same time |
| NFR.5 | Any device with internet connection should be able to view the map once launched | Mobile phones, iphones etc. should have access |
| NFR.6 | Databases should be kept locked | Passwords should be included for the databases |

## 3.5 Trade-offs

During the requirements analysis phase, trade-offs were made between adding advanced features such as cloud synchronization, mobile application, and AI & machine learning capabilities and maintaining the system's ease of use and accessibility for individuals. It was decided that advanced features would be included in the final version of the system, but not in the current prototype in order to prioritize the system's usability for its intended audience.

## 3.6 Summary of Findings and Key Issues

The main findings of the requirements analysis phase are that there is a need for a budget management system that addresses the limitations of traditional bookkeeping methods for individuals. The key issues that need to be addressed in the project are to ensure that the system is versatile and easy to use, provides real-time financial insights, and incorporates user-centered design principles to improve user adoption.

## 3.7 Assumptions and Constraints

During the requirements analysis phase, a number of assumptions and constraints were identified that will need to be taken into account during the development of the budget management system.

One of the main assumptions made was that the system would be developed for individuals, rather than businesses. This assumption was based on the literature review, which revealed that there is a lack of research on budget management systems for individuals. As such, the system will be designed with the specific needs of individuals in mind, rather than businesses.

Another assumption made was that the system would be developed for users who are familiar with online applications. This assumption was made based on the understanding that most individuals today are familiar with using online applications, and as such, the system will be designed to be user-friendly and easy to use.

A constraint that was identified was the limited resources available for the project. The budget management system will be developed on a limited budget, and as such, it will not be possible to include all the features that were initially identified as requirements. This constraint will need to be taken into account when prioritizing the features and capabilities of the system.

Another constraint that was identified was the limited time available for the project. The budget management system will need to be developed within a specific timeframe, and as such, it will not be possible to include all the features and capabilities that were initially identified as requirements. This constraint will need to be taken into account when prioritizing the features and capabilities of the system.

## 3.8 Testing and Validation Methods

The testing and validation of the budget management system is an essential part of the development process. It is important to ensure that the system meets its requirements and that it is fit for its intended purpose. There are several testing and validation methods that will be used in this project to ensure that the system meets its requirements.

### 3.8.1 Unit Testing

Unit testing is a technique that is used to test individual units or components of the system. The purpose of unit testing is to validate that each unit of the system is working correctly, and that it meets its requirements. Unit tests will be developed to test the individual components of the system, such as the user interface, the database, and the business logic.

### 3.8.2 Integration Testing

Integration testing is a technique that is used to test the interactions between different components of the system. The purpose of integration testing is to validate that the system is working correctly as a whole, and that all the components are integrated correctly. Integration tests will be developed to test the interactions between different components of the system, such as the user interface, the database, and the business logic.

### 3.8.3 Functional Testing

Functional testing is a technique that is used to test the functionality of the system. The purpose of functional testing is to validate that the system is working correctly, and that it meets its requirements. Functional tests will be developed to test the functionality of the system, such as the ability to track expenses, create budgets, and generate reports.

In summary, the requirements analysis phase has identified the key features and capabilities that the budget management system should have in order to meet the objectives of the project and address the problem identified in the literature review. Additionally, the requirements analysis phase has also identified the trade-offs that were made, the key issues that need to be addressed, and the testing and validation methods that will be used to ensure that the system meets its requirements.

# Chapter 4 – Planning & Management

## 4.1 Project Management Methodology

Waterfall model was chosen as the project management methodology for this project as it is a linear sequential approach that is suitable for projects with well-defined requirements and deliverables. The main advantage of using the Waterfall model is that it allows for a clear and structured progression through the project development life cycle, and it is easy to understand and implement.

## 4.2 Project Charter

The project charter is a document that outlines the project scope, objectives, deliverables, and other important information. The scope of this project is to develop a budget management system for individuals that addresses the limitations of traditional bookkeeping methods and existing budget management systems. The objectives of the project are to develop a budget management system that is versatile, easy to use, and provides real-time financial insights. The deliverables of the project include a working prototype of the budget management system.

## 4.3 Risk Management

Risk management is the process of identifying, assessing, and mitigating risks that may impact the project. The steps taken to identify risks included reviewing the project scope and objectives, as well as consulting with experts in the field. The risks that were identified included a lack of understanding of user needs and difficulty in developing a user-friendly interface. To mitigate these risks, user research was conducted and a user-centered design approach was adopted.

## 4.4 Personal Role and Management

In this project, I played the role of the project manager, developer, and tester. As the project manager, I was responsible for creating and managing the project plan, overseeing the development of the budget management system, and ensuring that the project objectives were met on time and within scope. As a developer, I was responsible for designing and developing the budget management system, including the user interface, database, and backend functionality. Additionally, I was responsible for testing the system and ensuring that it met the requirements outlined in the requirements analysis phase. I organized my work by breaking it down into smaller tasks and prioritizing them based on their importance and dependencies. I used a task management tool to keep track of my progress and ensure that I met my deadlines. I also regularly reviewed my work to ensure that it met the quality standards and that it was aligned with the project plan and scope. In terms of time management, I set a schedule for myself and made sure to stick to it as closely as possible. I also made sure to take regular breaks and manage my time effectively to avoid burnout. To track my progress, I regularly reviewed my tasks and updated my task management tool accordingly. Overall, I believe that my role and management approach were effective in ensuring that the project was completed on time and within scope.

# Chapter 5 – Methodologies

## 4.1 Research Methods

The research for this project was conducted primarily through literature review. This involved reading and analyzing existing research papers, articles, and other publications related to budget management systems and their design. Additionally, existing budget management applications were analyzed to gain an understanding of the features and functionality that are commonly found in these systems.

## 4.2 Software Development Methodology

The software development methodology used in this project was the Waterfall methodology. This methodology is a linear, sequential approach to software development, where the process is divided into distinct phases, such as requirements gathering, design, implementation, testing, and maintenance. The Waterfall methodology was chosen for this project because it is well suited to projects with well-defined and fixed requirements, and it provides a clear and structured approach to development.

## 4.3 Design Methodologies

The design of the budget management system was informed by user-centered design principles. This approach emphasizes the needs and wants of the end-users, and it involves conducting user research, usability testing, and prototyping to ensure that the system is designed with the user in mind. Additionally, the design of the system was informed by human-computer interaction principles, which aim to create systems that are easy to use and understand.

## 4.4 Testing Methodologies

A range of testing methodologies were used in this project to ensure the quality of the system. Unit testing, integration testing, and acceptance testing were all used to ensure that the system functions as intended and meets the requirements. Additionally, usability testing was conducted to ensure that the system is easy to use and understand.

## 4.5 Project Management Methodology

The project management methodology used in this project was based on the Waterfall methodology, which is a linear, sequential approach to project management. This methodology was chosen because it provides a clear and structured approach to project management, and it is well suited to projects with well-defined and fixed requirements.

## 4.6 Challenges and Findings

During the implementation of the chosen methodologies, the main challenge encountered was the lack of resources and budget. However, this was overcome by utilizing free and open-source tools and resources, as well as by simplifying the scope of the project. Additionally, the limited time frame was a challenge, which was overcome by prioritizing the most important features and functionalities.

In summary, the research for this project was conducted primarily through literature review. The software development methodology used was the Waterfall methodology, which is a linear, sequential approach to software development. The design of the system was informed by user-centered design principles and human-computer interaction principles. A range of testing methodologies were used to ensure the quality of the system, and project management methodology used was based on the Waterfall methodology. The main challenge encountered was the lack of resources and budget, but it was overcome by utilizing free and open-source tools and resources, as well as by simplifying the scope of the project.

# Chapter 6 – Design

## 6.1 UI & UX Design

### 6.1.1 User Interface Design

The user interface (UI) design of the budget management system is a critical aspect of the overall user experience. The goal of the UI design is to create a simple, intuitive, and visually appealing interface that makes it easy for users to navigate and interact with the system.

To achieve this goal, the UI design of the budget management system includes a number of key elements, such as the layout, color scheme, typography, and iconography. The layout of the system is organized and easy to navigate, with a clear hierarchy of information and intuitive navigation. The color scheme is designed to be simple, clean, and consistent, with a color palette that is easy on the eyes. The typography is designed to be easy to read and consistent throughout the system, while the iconography is designed to be simple and intuitive, making it easy for users to understand the meaning of each icon.

### 6.1.2 User Experience Design

The user experience (UX) design of the budget management system is also a critical aspect of the overall user experience. The goal of the UX design is to create a system that is easy to use, efficient, and enjoyable for users.

To achieve this goal, the UX design of the budget management system includes a number of key elements, such as the user flow, navigation, and feedback. The user flow is designed to be simple and intuitive, with a clear and logical progression from one task to the next. The navigation is designed to be easy to use, with clear and consistent navigation labels and intuitive navigation. Feedback is provided to users through the use of clear and concise messages, which help users to understand the system's responses and what actions they need to take next.

### 6.1.3 Design of Some Pages

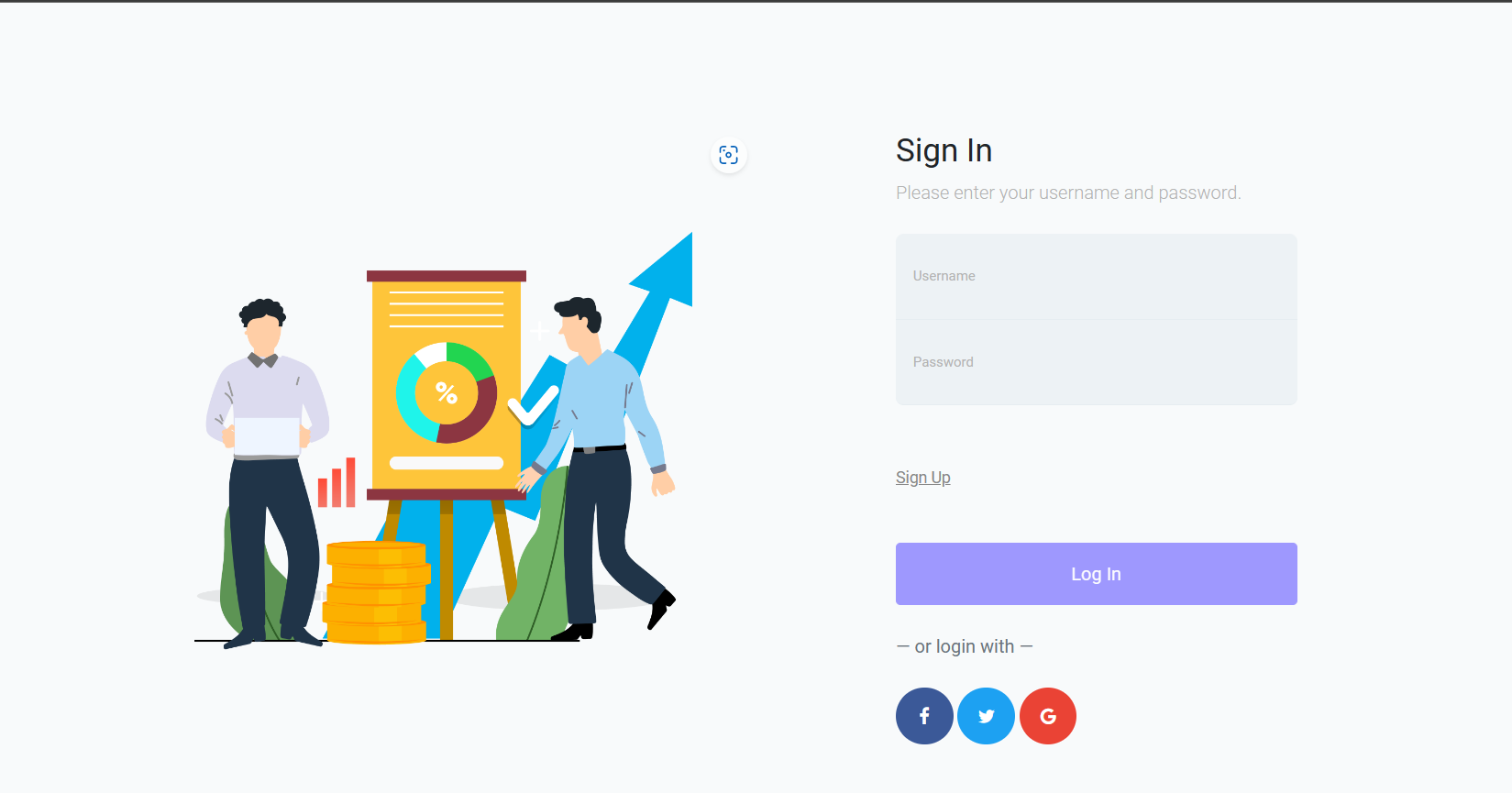


Figure 1 Sign-In Screen

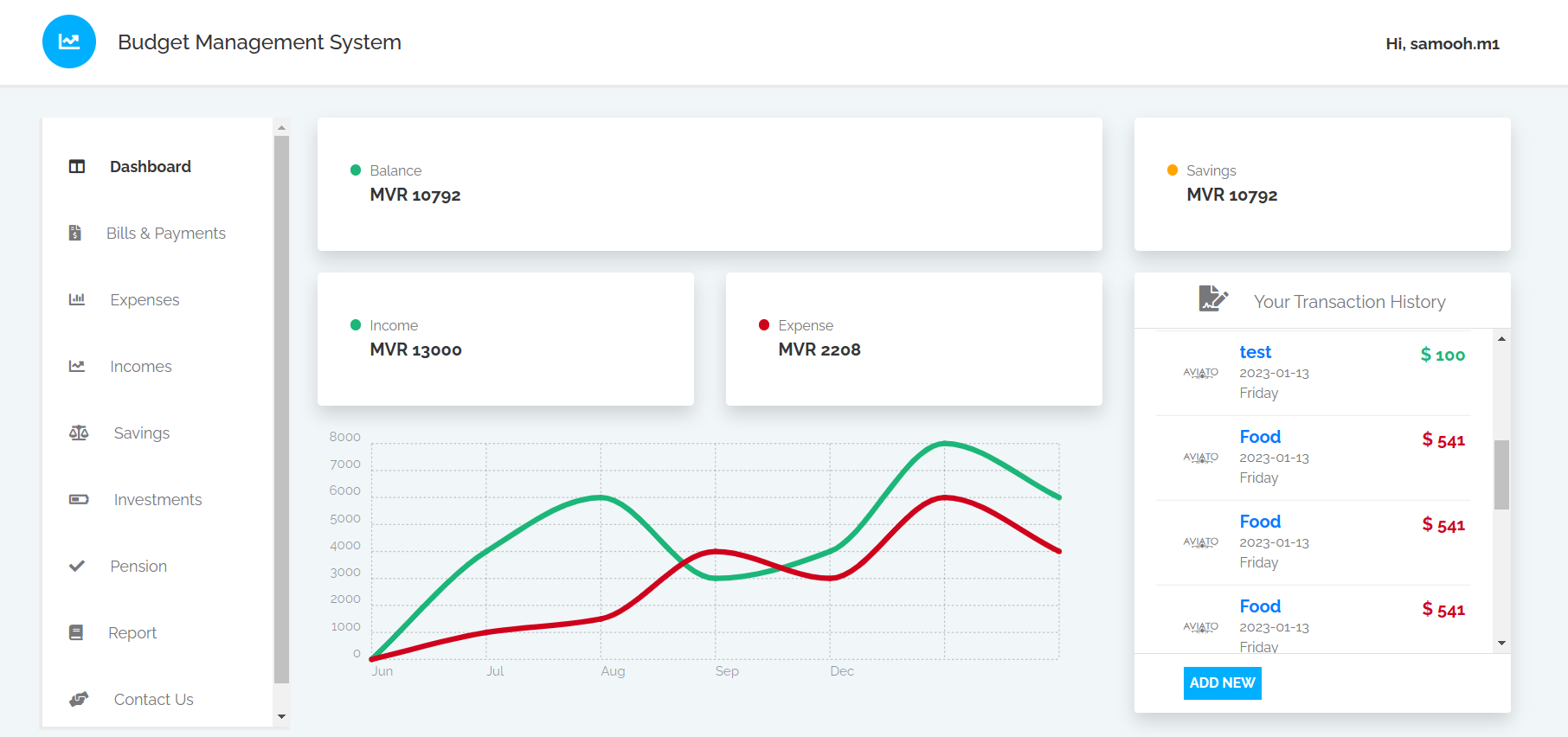


Figure 2 Home Page (Dashboard)

## 6.2 Design Principles and Guidelines

Design principles and guidelines are a set of rules and best practices that are used to guide the design of a system. In the case of budget management systems, these principles and guidelines are used to ensure that the system is accessible, usable, and secure for its users.

Accessibility is the practice of designing systems that can be used by people with disabilities. This includes designing interfaces that are keyboard accessible, providing alternative text for images, and ensuring that the interface is usable with screen readers. In the case of the budget management system, accessibility was taken into consideration in the design process by providing clear and visible labels for buttons, using high-contrast colors, and providing alternative text for images.

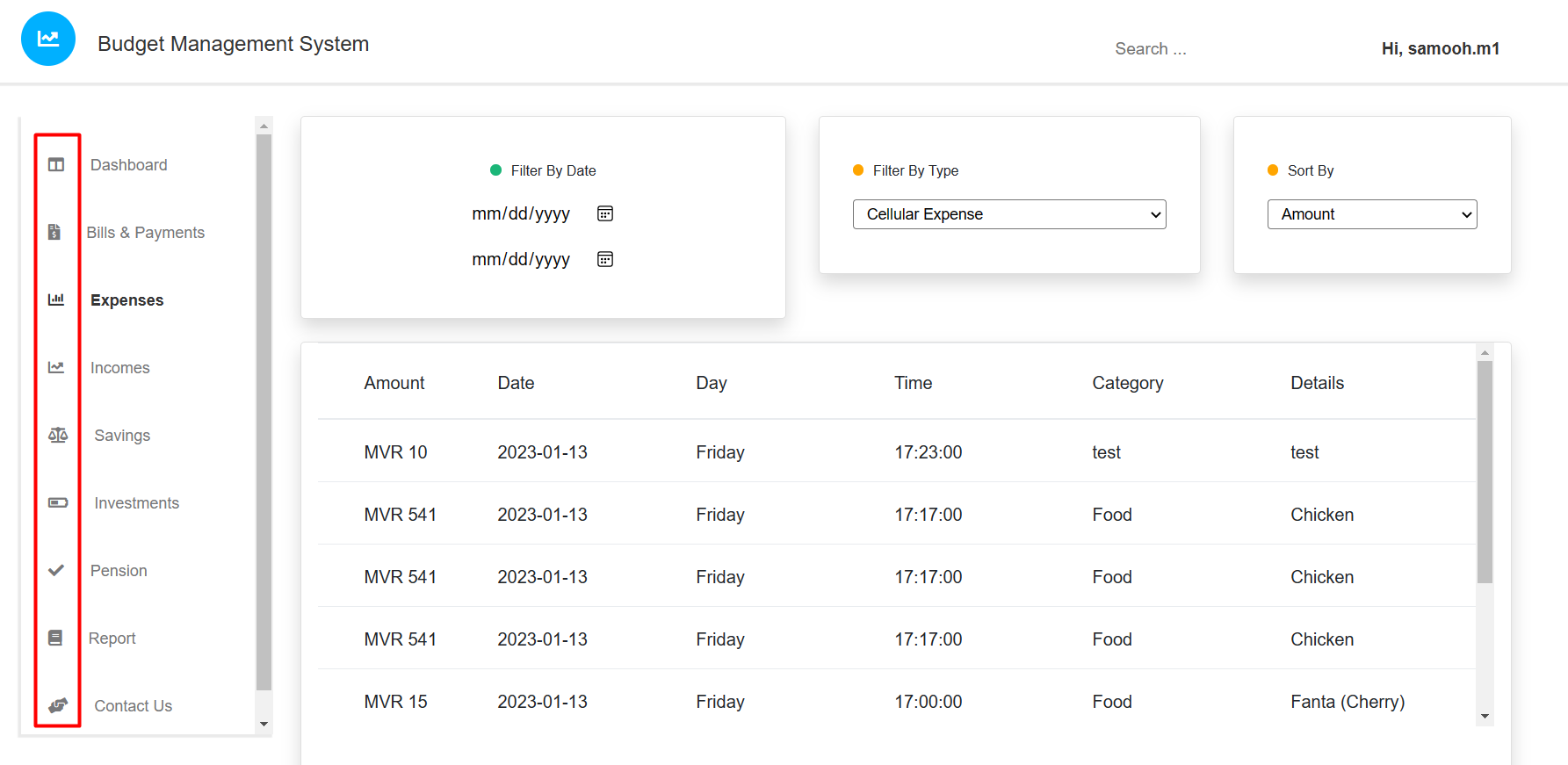


Figure 3 Icons for Pages

Usability refers to the ease of use of a system. This includes making sure that the interface is intuitive and easy to navigate, providing clear instructions, and reducing the number of clicks or steps required to complete a task. In the case of the budget management system, usability was taken into consideration by providing clear instructions and labels, using a simple and intuitive layout, and providing a search function to help users find the information they need quickly.

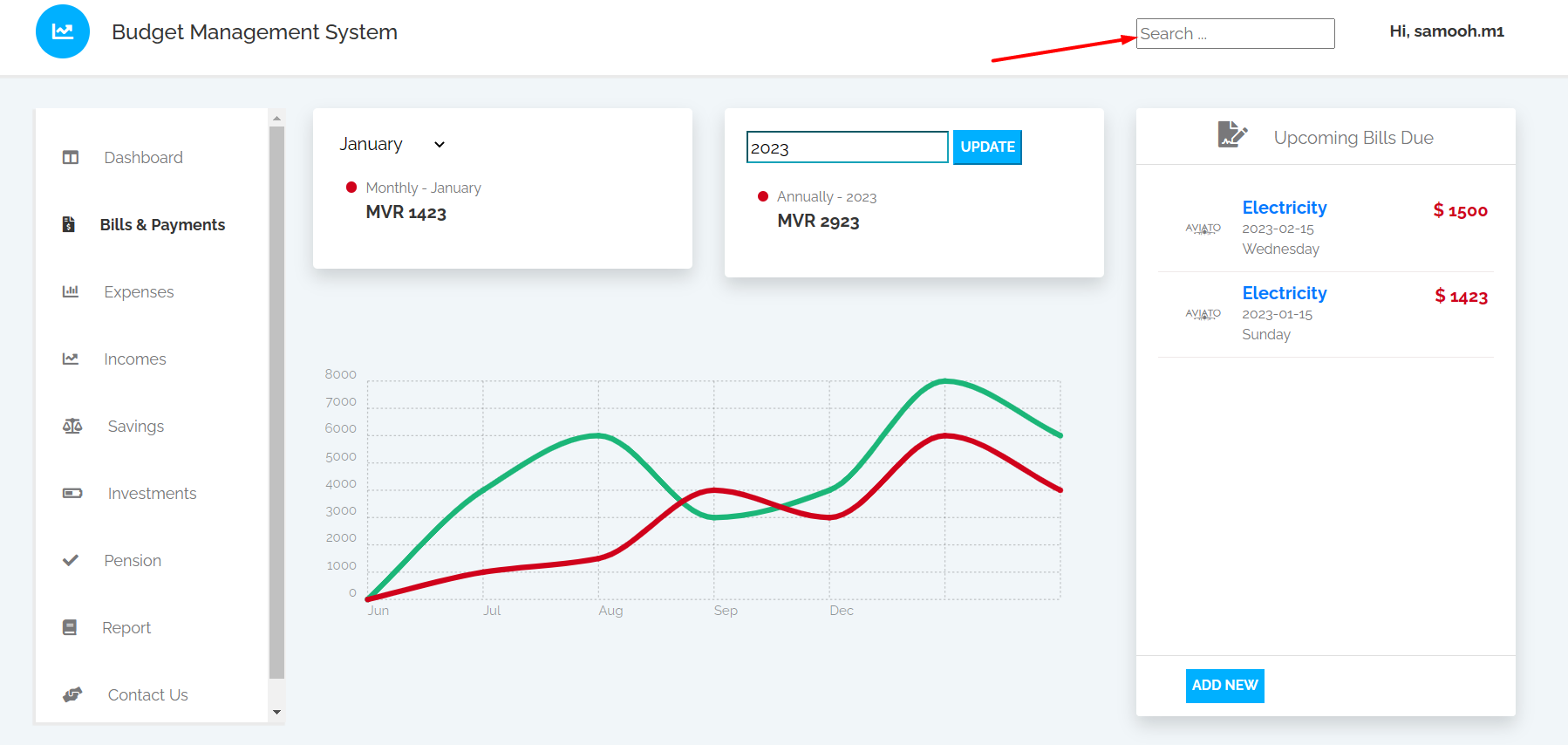


Figure 4 Search Button

Security is the practice of designing systems that are resistant to unauthorized access and data breaches. This includes using encryption, secure authentication, and access controls to protect sensitive information. In the case of the budget management system, security was taken into consideration by encrypting user passwords in SHA256 hashing before storing them in the database. In the future the plan for the application is to use secure authentication methods, such as multi-factor authentication, and implementing access controls to ensure that only authorized users have access to sensitive information.

In summary, accessibility, usability, and security were considered as key design principles in the budget management system, and were integrated into the design process to ensure that the system is usable, easy to navigate, and secure for its intended users.

## 6.3 Data Model and Database Design

The data model and database design of the budget management system play a crucial role in ensuring the efficient storage and retrieval of financial data. In this project, the data model was designed using entity-relationship diagrams (ERDs) to represent the entities and relationships in the system. The ERDs were used to create a detailed representation of the data structure, including the entities, attributes, and relationships between the entities.

The system includes entities for users, accounts, transactions, categories, and reports. The user entity includes information like name, email, and password. The accounts entity includes financial accounts like checking or credit card. The transactions entity includes financial transactions like purchases or deposits. The categories entity includes categories like groceries or rent. The reports entity includes details of the reports generated by the user.

The relationships between the entities were also defined in the ERDs. For example, a user can have multiple accounts, and an account can have multiple transactions. Additionally, a transaction can be assigned to a category, and a category can contain multiple transactions. These relationships were used to ensure data consistency and integrity in the system.

In addition to the ERDs, data constraints were also defined to ensure data integrity and consistency. For example, the system enforces constraints such as unique email addresses for users and unique account names for accounts. Additionally, foreign key constraints were used to ensure that a transaction is associated with an account and a category, and that a category is associated with a user.

Figure 30 shows the ER Diagram that was drawn for this application with all the relationships mentioned in the previous paragraphs. However, please keep in mind that, the tables that are used in the prototype of the application will differ to that from the diagram, as the current prototype of the application will only contain the basic features of the application

## 6.4 Use Case Diagram

In the design phase of the project, a use case diagram was created to visually represent the interactions between the system, users, and administrators. The use case diagram helps to identify the functional requirements of the system and the relationships between the actors and the use cases.

The User has several use cases including Register, Login, Add Account, Add Category, Add Bill, Add Savings, Add Budget, Review Transactions, Review Budget, and Review Savings. The Administrator has only 1 use case, Resolve Queries, which is used to address any issues or questions that the users may have.

The relationships between the use cases were carefully considered in order to ensure that the system would be easy to use and understand. For example, the Register use case has an association with the Login use case, as it is a necessary step for the user to be able to access the system. Similarly, the Add/Edit/Delete Transactions use case has an association with the Add/Edit/Delete Account, Add /Edit/Delete Category, and Add/Edit/Delete Recuring Bills use cases, as the user needs to have added these elements in order to add a transaction.

The use case diagram also includes a System actor, which has several use cases including Generate Reports, Synchronize Data, and Send Reminders. These use cases are included to show the functionality that the system performs in the background to provide an efficient and effective budget management experience for the users.

Overall, the use case diagram provides a clear and comprehensive overview of the system's functionality and the interactions between the actors. It serves as an important tool for understanding the system's capabilities and the user's journey within the system. It also helps to ensure that all the necessary functionality has been considered and that the system is designed to meet the needs of the users. However please note that, not all functionality mentioned in the use case diagram was completed in the prototype of the project.

## 6.5 Sequence Diagram

A sequence diagram is a type of behavioral diagram that illustrates the interactions between objects or components in a system. It is typically used to model the dynamic behavior of a system, and is particularly useful for complex use cases that involve multiple interactions between different objects or components.

In this project, a sequence diagram was drawn for the complex use cases such as login, registration and transaction management. This helped to understand the flow of the system and how different objects or components interact with each other. The sequence diagram shows the flow of messages between the objects and the order in which they are exchanged. It also shows the lifeline of each object and the states it goes through during the interaction.

The importance of using a sequence diagram in this project was to have a clear understanding of the interactions between objects and to identify any potential issues or bottlenecks. It helped to identify the key classes and objects in the system and their responsibilities. By creating the sequence diagram, it was easier to understand the flow of the system and identify any missing interactions or requirements. However, not all of the functionality mentioned in the sequence diagram was coded, such as the OTP verification part as the current version of the application is a prototype.

# Chapter 7 – Implementation

## 7.1 Tools and Frameworks Used in Implementation

The budget management system was implemented using a combination of HTML, CSS, JavaScript, and JSP. The system's front-end was built using HTML and CSS for the layout and design, while JavaScript was used for implementing some of the interactive features. The back-end of the system was built using JSP and Java, which was used for connecting to the database and processing data. The system was developed using the Model-View-Controller (MVC) framework, which helped in separating the presentation layer, business logic layer, and data layer. The development environment used for this project was NetBeans IDE which provided a seamless development experience and made it easier to test and debug the code.

However, despite the effectiveness of the tools and frameworks utilized, a robust version control system is essential in ensuring the success of the project, as it allows for tracking of changes & management of conflicts. Thus, we are going to look into it next.

## 7.2 Version Control

In this project, version control was implemented using GitHub, which allowed for efficient tracking of changes made to the codebase, as well as the ability to manage & deal with conflicts. By using GitHub, I was able to maintain a clear and organized history of the development process, making it easy to identify and revert to previous versions if needed. Additionally, using GitHub's application made it easy to push commits and pull updates from the remote repository, ensuring that I was always working with the latest version of the code. Overall, the use of version control with GitHub greatly improved the project's productivity, efficiency, and organization, as it helped me to keep track of all the changes made to the codebase.

## 7.3 Sign Up

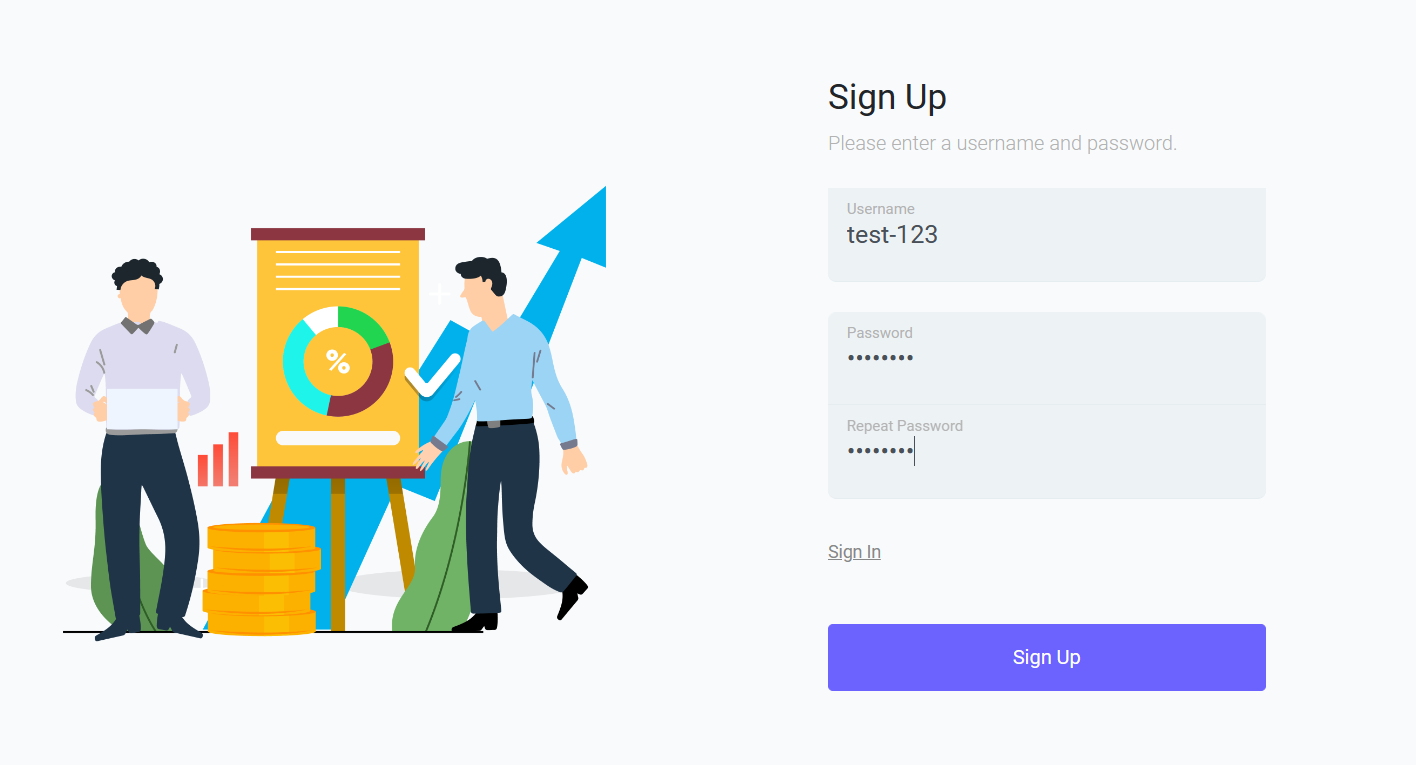


Figure 5 Sign-Up Screen (Prototype)

In the current prototype of the Budget Management System, the user is only required to provide a username and password to sign up. However, for the final version of the application, a more comprehensive registration process is planned. This will include additional information such as first name, last name, email, date of birth, country, phone number, a security question and a security answer. Users will have the option to register with either their email address or phone number, and will be required to enter a One-Time-Password (OTP) sent to their chosen platform to complete registration. These details can be edited from the user's profile page once logged in. Additional optional fields such as address, gender and two-factor authentication will also be included.

Despite the current prototype not having all of these features, some basic security measures have been implemented, such as requiring the user to confirm their password and checking for a minimum username length and unique username availability. After successful registration, the user is directed to the login page.

## 7.4 Log In

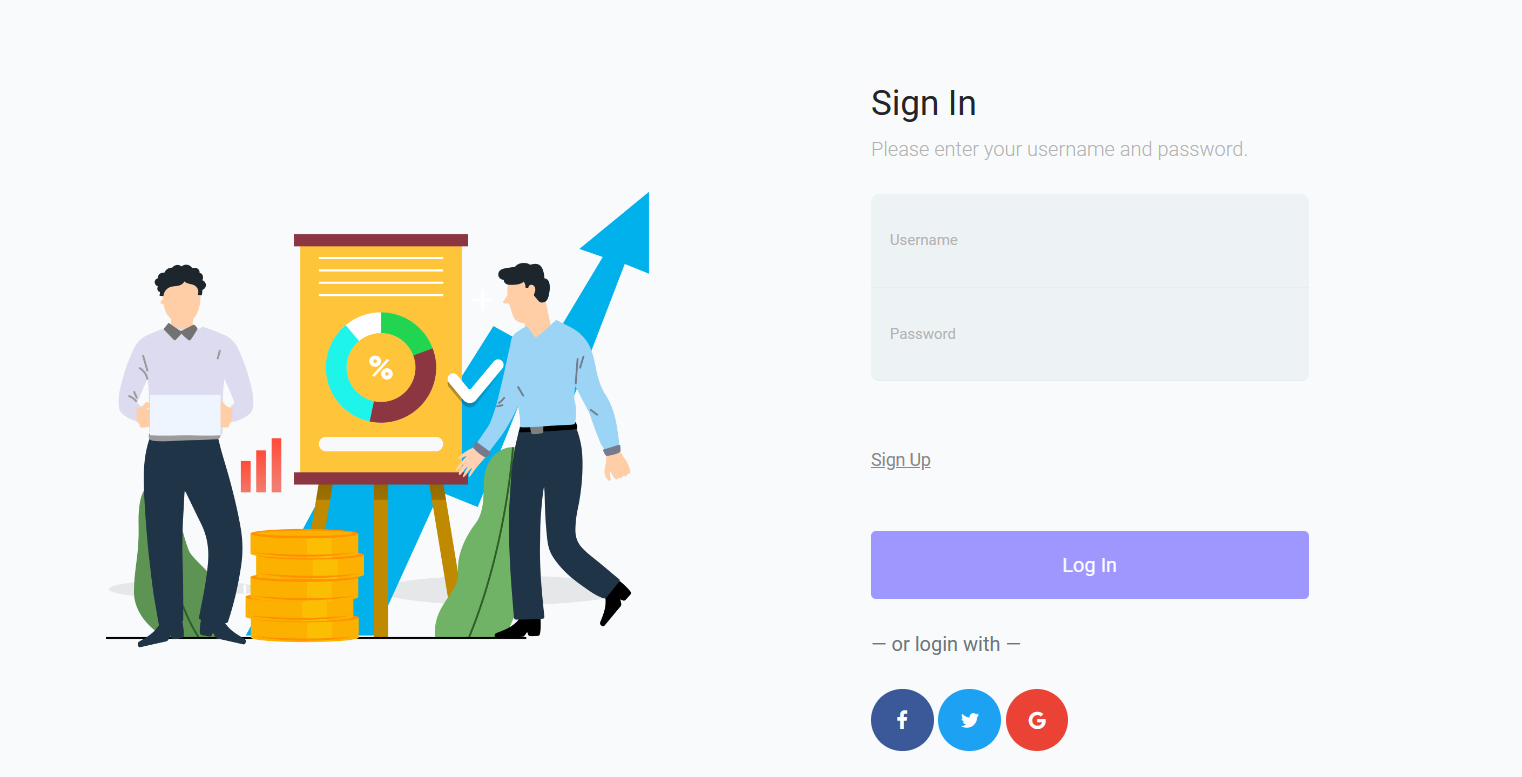


Figure 6 Sign in Screen (Prototype)

The login screen of the Budget Management System prototype requires users to enter their previously registered username and password. In case the entered credentials do not match any existing records in the database, an error message will appear indicating that the username or password is incorrect. To ensure security, similar to the sign-up page, validation checks have been put in place to ensure the username meets the required length.

The logos of Facebook, Twitter, and Google, although currently non-functional, have been included in the prototype as it is planned to have a one-click sign-up feature using these platforms in the final version of the application. Additionally, the option for two-step verification may also be enabled in the user profile menu in the future.

Upon successful login, the user will then be redirected to the dashboard or homepage of the application.

## 7.5 Dashboard

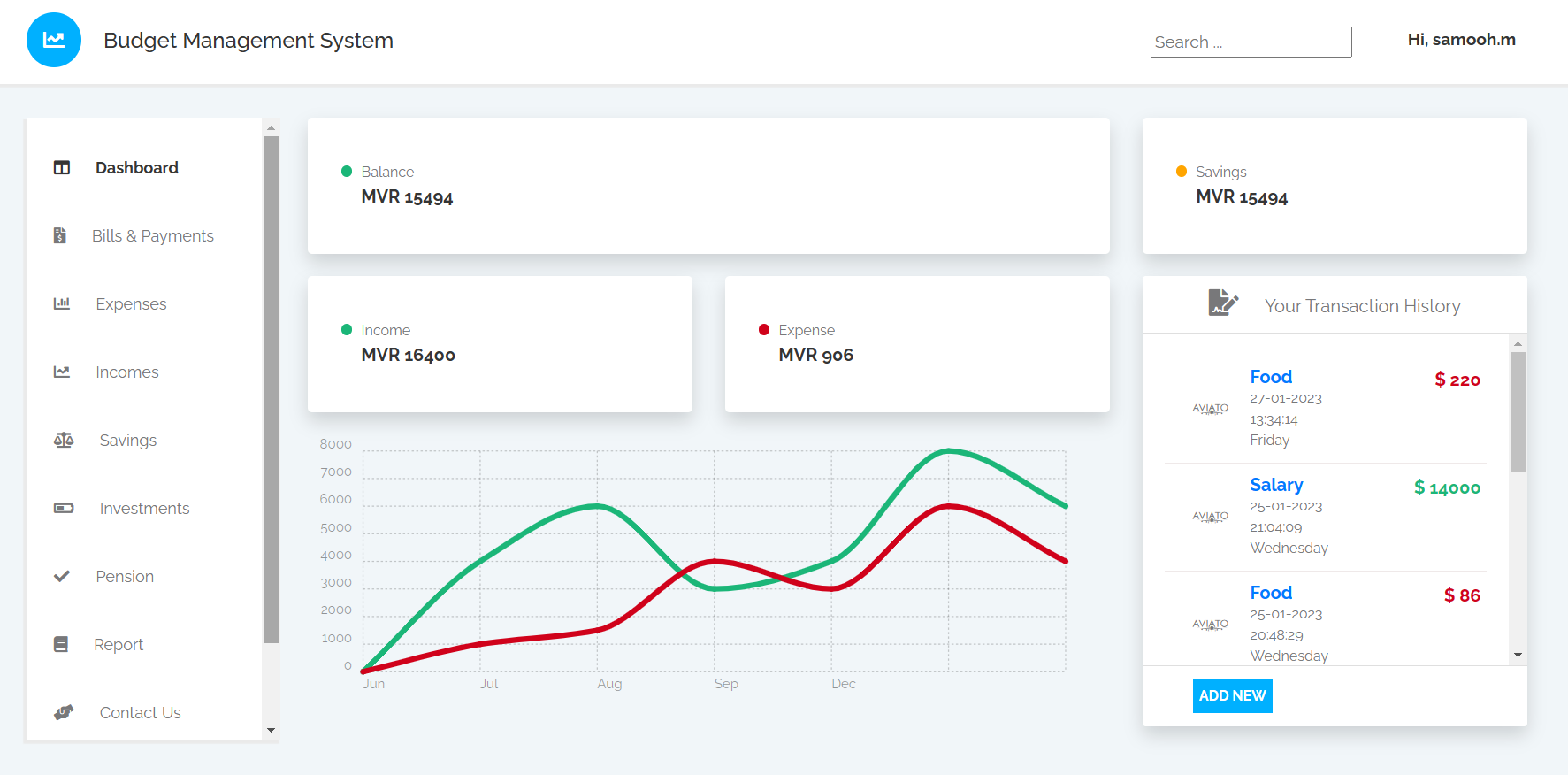


Figure 7 Dashboard

After logging in, the user is directed to the interactive and user-friendly dashboard. It includes a line graph that will show the user's monthly expenses and income once the project is completed, as well as current balance, total income, total expenses, and savings account total. Users can view transaction history and add new transactions using the "add new" button. Users can also access their profile page and navigate to other pages by clicking on corresponding page name on the left side of the screen. The design also includes a search bar, which will be functional once implemented, for easy searching of specific features and pages.

## 7.6 Bills & Payments

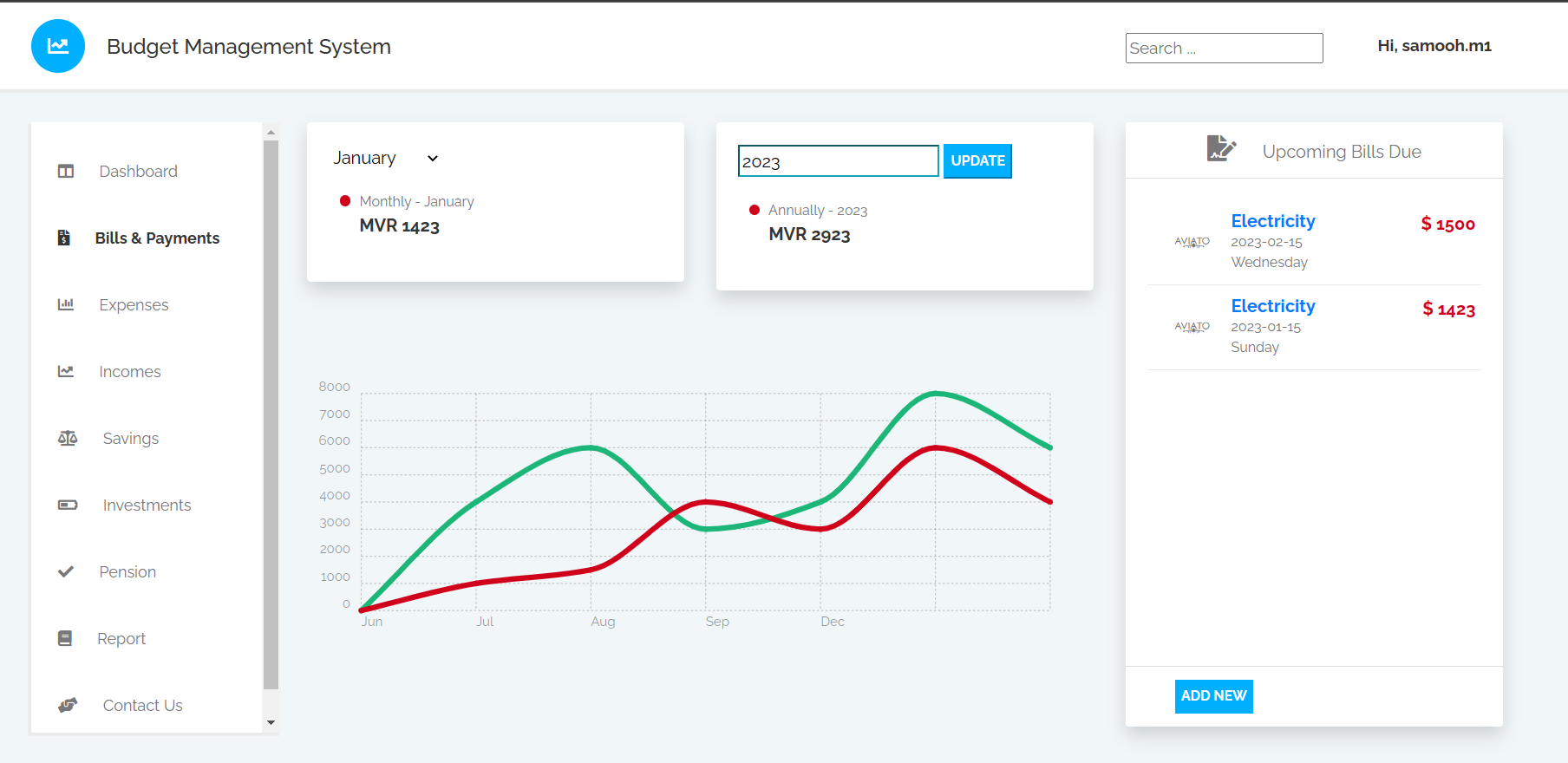


Figure 8 Bills & Payments Screen

The Bills & Payments screen, as shown in Figure 8, has been designed with a line graph similar to that of the dashboard. Although it is not yet functional, once completed, it will allow the user to see the amount spent on various bills each month. Users will also have the option to modify the data displayed on the graph by specifying the month and year range.

Currently, the prototype displays the total amount spent on bills for the selected month or year. Additionally, the right side of the screen showcases upcoming bills that are due soon. While the plan is to include a payment feature, allowing users to pay their bills directly from the application, this function is not yet implemented.

Users will also have the ability to add new bills to the list by clicking the "add new" button. However, this feature is also not yet implemented, but once completed, users will be able to add payments with the option to specify whether the payment is recurring or non-recurring, its category and type, etc.

Furthermore, a report feature is also planned to be included on the Reports page, which will provide users with a comprehensive list of all payments made for bills.

## 7.7 Incomes & Expenses

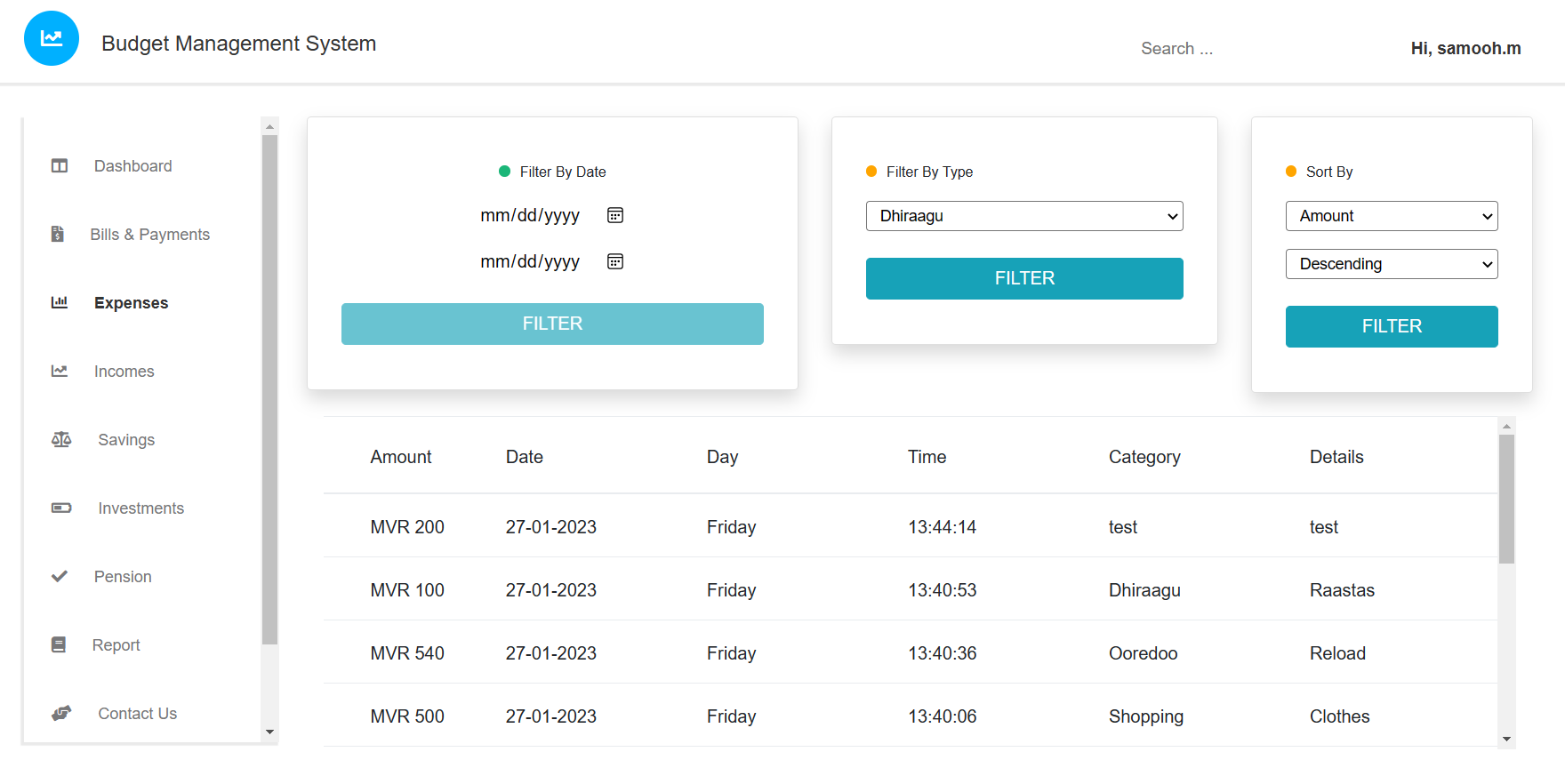


Figure 9 Expenses Table with Headings

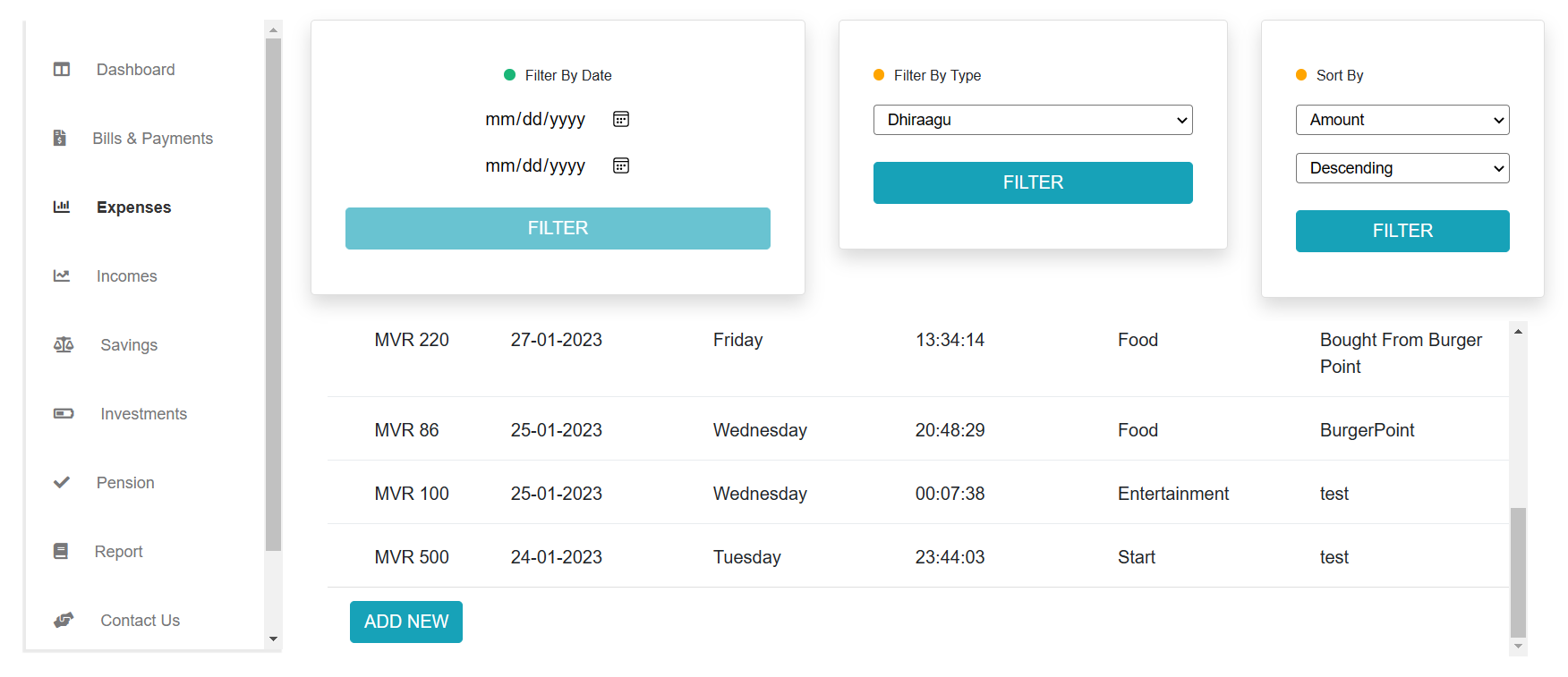


Figure 10 Expenses Table with Add New Button

The expenses table, as shown in Figure 10, allows users to view detailed information about their expenses, including the amount, date, day, time, category and details. The table can be easily filtered by date or category and sorted by amount, date, or time. To add a new expense, users simply need to click on the "Add New" button and fill out the details of the expense in the pop-up window. Similarly, the incomes table also allows users to view and filter/sort information about their income. In future updates, it is planned to include additional functionality such as the ability to change the columns displayed in the table, filter by entering values directly in the table headings, and sort by clicking on the table headings. These features will make the application even more user-friendly and easy to use.

## 7.8 My Profile

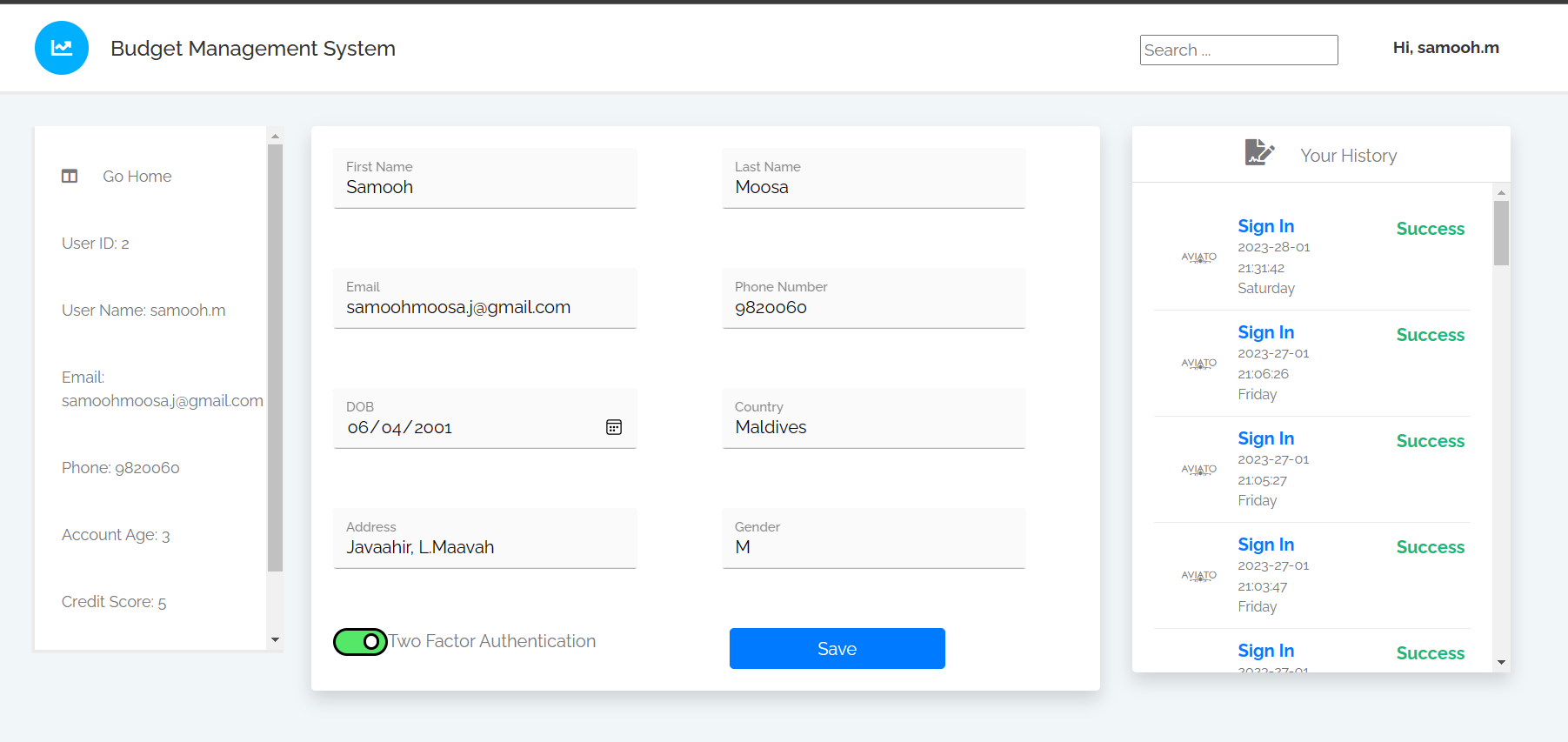


Figure 11 My Profile Page

The "My Profiles" page, as depicted in Figure 11, can be accessed by the user through the navigation menu located in the top right corner of any page. This page allows the user to view and manage their personal account details, including their user ID, username, email, phone number, account age, and credit score. The user is also able to edit certain account details, such as their first and last name, email, phone number, date of birth, country, address, and gender. Additionally, the user has the option to enable two-factor authentication for added security, although this feature has not yet been fully implemented in the prototype.

Furthermore, the "My Profiles" page includes a log table, referred to as "Your History", which allows the user to view details of each attempted and successful sign-in, as well as any changes made to their account profile. This feature serves as an important security measure, as it allows the user to detect any unauthorized attempts to access their account and take appropriate actions, such as enabling two-factor authentication or changing their password. However, it should be noted that while this feature is present in the prototype, it is not currently possible to determine the location from which an attempted sign-in occurred.

## 7.9 Future Updates

Although the current prototype of the budget management system is limited in its functionality, as previously outlined in the report, the final version of the application will include a wide range of new features and enhancements.

One of the new features planned for the budget management application is the ability for users to enter more personal information during registration opposed to the entering the data in my profiles page. Additionally, users will be required to answer a security question and provide a security answer to further secure their account. The database has been configured for this already.

Another new feature planned for the application is the implementation of two-factor authentication. This added security measure will require users to verify their identity through a one-time password sent to their email or phone number before being able to log in to their account.

Another planned feature is the inclusion of a report feature in the application, which will allow users to generate different kind of reports for their expenses and incomes. This report will allow users to see a detailed breakdown of their spending, and will include the ability to filter and sort the data to make it easier to analyze.

In addition, the application will include the ability for users to customize the appearance of their tables, filter by entering the filter value near the headings and sorted by clicking on the heading. This feature will make the application much easier to use.

All of these new features are designed to improve the overall usability and security of the budget management application, giving users more control over their finances and allowing them to make more informed decisions about their spending.

# Chapter 8 – Testing

## 8.1 Functional Requirement Test Cases

|  |  |  |  |
| --- | --- | --- | --- |
| Identifier | Requirement | Expected Result | Actual Result |
| FR.1 | Allow users to register for the application | Check whether the user is able to give a username and password and sign up for the application. The values provided should be entered in the database in the correct columns. | Successful   * User successfully registered and data provided was recorded in the database * User is redirected to the login page |
| FR.2 | To be able to log in to the application. | A registered user should be able to login to the application with the correct credentials. | Successful   * Successfully logged into the application with the correct credentials entered in the sign-up test case. * User redirected to the dashboard page |
| FR.3 | The option to log out of the application. | User should be able to logout of the application after logging in. | 1st fail – 2nd - Success  Logout servlet was implemented but no logout button was added to any page. In order to fix this, logout button was added to the left side menu of each page. After this user was able to logout. |
| FR.4 | Change User details from my profile | User should be able to edit their details from my profile page. Changed details should be visible in the logs table. | Successful  User successfully changed first name, last name, email, date of birth, two factor authentication mobile number from my profiles page. The details were correctly logged in the logs table. |
| FR.5 | Add Transactions | User should be able to add incomes and expenses for different categories | Successful  Transactions of both types (expenses and incomes) were added for different categories from the dashboard page. (New categories were created in the categories table in db) |
| FR.6 | Modify Transactions | Users should be able to edit details of transactions that are already recorded | Failed  Transactions modify function is not yet added in the prototype |
| FR.7 | Hash Password | The password user enters when signing up should be hashed and stored in the database. The original password should not be stored | Successful  Password entered was hashed in “SHA256” and the resulting string was stored in the users table. When logging in the entered password is hashed in the same format and compared to the value in db. |
| FR.8 | Calculate Expense, Income & Balance | User should able to see their expense, income & balance | Successful  System is able to calculate the incomes and expenses from the transactions table for each user. The correct values were shown in the dashboard page. The balance was depicted as income – expense. |
| FR.9 | Add Bills | User is able to add details of their bills from bills and payments screen | Failed  Bill addition feature is not yet included in the prototype |
| FR.10 | Load users personalized categories on expenses and income | When user goes to expenses or incomes page, categories need to be loaded with their specific ones and it should not include other users categories | Successful  Successfully able to load users personalized categories on Categories dropdown |
| FR.11 | Filter Table (With Date) | User should be able to filter income and expense tables by date | Successful  Successfully filtered the table with a given date |
| FR.12 | Filter Table (With Category) | User should be able to filter the table with selected category | Successful  Successfully filtered the table with selected category |
| FR.13 | Delete Transactions | User is able to delete unwanted transactions | Successful  User able to delete transaction from their dashboard after clicking on the name of the transaction. The transaction status was set to 0 from the db. The system calculations do not include transactions with status 0. So, transaction is considered deleted. |

Functional Requirement test case 3 was a must requirement but it failed at first try. So the below changed was brought from the jsp files in order to enable the users to be able to logout from any page.

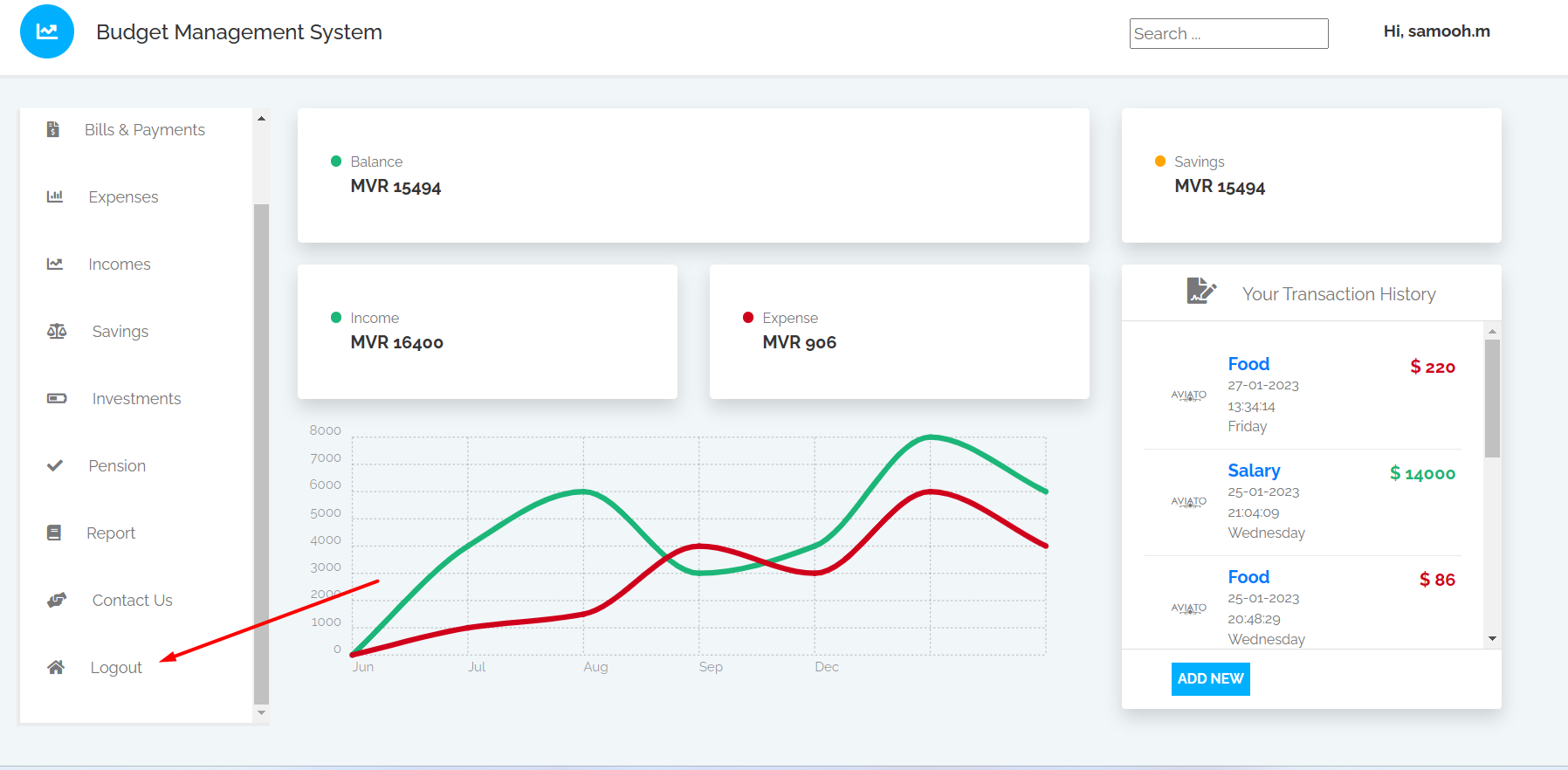


Figure 12 Logout button



Figure 13 Logout servlet

Once the user clicks on the logout button, the session will be invalidated and the user will be redirected back to the home page.

# Chapter 9 – Evaluation

The project being evaluated aimed at creating a budget management system for individuals. It was successful in achieving its goals and the developed system has potential to help individuals manage their finances effectively. Some requirements, such as bills and payments addition and modify transactions, were not implemented due to time constraints. However, the main functionality of the application is almost completed. These tasks were not prioritized as there were more important features that needed to be included. There is still room for improvement before the final release, but overall, I am satisfied with the results despite some challenges faced during implementation, such as my lack of knowledge in web development and difficulty with design. The final design is simple, elegant, and user-friendly, as intended.

# Chapter 10 – Conclusion

In conclusion, the budget management system project was a success in achieving its objectives and the developed system has the potential to assist individuals in managing their finances effectively. It is user-friendly and elegant in design, making it accessible to a wide range of individuals. Additionally, the application's functionality and flexibility allow it to be adapted to meet the needs of different individuals. Overall, I believe this app has the potential to be a valuable tool for many people.

# References

Bote, J. (2021) *Automated Bookkeeping: mimicking human data entry in finance* [Online]. Available from: <https://www.zeni.ai/blog/automated-bookkeeping#:~:text=Simply%20put%2C%20automated%20bookkeeping%20is%20the%20completion%20of,human%20data%20entry%20in%20finance%20software%20tools.%20%E2%80%8D> [Accessed 30 July 2022].

Castellano, M. (1998) Automating expense reports. *Internal Auditor* 55(4), p.21. Available at: <https://go.gale.com/ps/i.do?id=GALE%7CA21119264&sid=googleScholar&v=2.1&it=r&linkaccess=abs&issn=00205745&p=AONE&sw=w&userGroupName=anon%7E67c3542c> [Accessed 18 September 2022].

Centage. (2015) *Advantages of Cloud-Based Business Budgeting Software* [Online]. Available from: <https://www.centage.com/blog/advantages-of-cloud-based-business-budgeting-software/> [Accessed 28 November 2022].

CFI Team. (2022) *Accounting Standard* [Online]. Available from: <https://corporatefinanceinstitute.com/resources/accounting/accounting-standard/> [Accessed 7 January 2023].

Chrisos, M. (2020) *6 Benefits of Financial Management Software for SMBs* [Online]. Available from: <https://www.techfunnel.com/fintech/6-benefits-of-financial-management-software-for-smbs/> [Accessed 10 August 2022].

Data Stems. (2021) *How Automated Business Intelligence will Save Time and Money* [Online]. Available from: <https://www.datastems.com/blog/how-automated-business-intelligence-will-save-time-and-money> [Accessed 10 December 2022].

Davis, K. (2022) *Timing is Everything: The Importance of Real-Time Financial Feedback* [Online]. Available from: <https://www.jmco.com/articles/collegiate-athletics/real-time-financial-feedback/#:~:text=Real-time%20financial%20feedback%20makes,down%20silos%20in%20your%20organization> [Accessed 18 December 2022].

DeBenedetti, J. (2016) *Why Is It Important for a Business to Compare Its Previous Years' Financial Statements?* [Online]. Available from: <https://smallbusiness.chron.com/important-business-compare-its-previous-years-financial-statements-80717.html> [Accessed 8 September 2022].

Dziak, M. (2021) *6 Benefits of an Integrated Budgeting and Planning Process* [Online]. Available from: <https://www.futureviewsystems.com/blog/6-benefits-of-an-integrated-budgeting-and-planning-process> [Accessed 13 November 2022].

Freer, A. (2022) *Money management apps grow users by 46%* [Online]. Available from: <https://www.businessofapps.com/news/money-management-apps-grow-users-by-46/#:~:text=Money%20management%20apps%20on%20Android,the%20cost%20of%20living%20crisis> [Accessed 8 November 2022].

Horton, C. (2022) *What Is a Budget App?* [Online]. Available from: <https://www.thebalancemoney.com/what-is-a-budget-app-5216548#:~:text=Vedfelt%20%2F%20Getty%20Images-,Definition,and%20automate%20your%20savings%20goals> [Accessed 15 August 2022].

IBM. (2021) *Benefits of blockchain* [Online]. Available from: <https://www.ibm.com/topics/benefits-of-blockchain#:~:text=Blockchain%20increases%20trust%2C%20security%2C%20transparency,cost%20savings%20with%20new%20efficiencies.&text=Blockchain%20for%20business%20uses%20a,accessed%20by%20members%20with%20permission> [Accessed 29 November 2022].

Indeed Editorial Team. (2022) *What Is A Finance Management System? (With Benefits)* [Online]. Available from: <https://in.indeed.com/career-advice/career-development/what-is-finance-management-system> [Accessed 15 December 2022].

Interaction Design Foundation. (2017) *User Centered Design* [Online]. Available from: <https://www.interaction-design.org/literature/topics/user-centered-design> [Accessed 3 January 2023].

Intuit. (2019) *Get your free credit score* [Online]. Available from: <https://mint.intuit.com/how-mint-works/credit> [Accessed 2 October 2022].

Intuit. (2020) *Online Monthly Bill Tracking & Reminders* [Online]. Available from: <https://mint.intuit.com/how-mint-works/bills> [Accessed 28 September 2022].

Kenton, W. (2022) *Accounting Standard Definition: How It Works* [Online]. Available from: <https://www.investopedia.com/terms/a/accounting-standard.asp> [Accessed 6 January 2023].

MoneyDance. (2008) *Moneydance - Personal Finance Manager for Mac, Windows, and Linux | Infinite Kind* [Online]. Available from: <https://moneydance.com/> [Accessed 4 November 2022].

Multiview. (2022) *Compliance in Accounting* [Online]. Available from: <https://multiviewcorp.com/blog/accounting-compliance-guide> [Accessed 6 January 2023].

Novoseltseva, E. (2017) *User-Centered Design: An Introduction* [Online]. Available from: <https://usabilitygeek.com/user-centered-design-introduction/> [Accessed 5 January 2023].

NTT. (2019) *7 Most Common User Adoption Challenges Facing IT Professionals* [Online]. Available at: <https://cdn.mediavalet.com/eunl/ntt/gugE3PNddkyVm1CJuvnqsQ/daZ3TO6kIEi4i7nUK9fxfA/Original/NTT-Inbound-P2-eBook-A4-7-Most-Common-User-Adoption-Challenges-Facing-IT-Professionals_EN.pdf> [Accessed 1 January 2023].

Parletta, R. (2022) *What Are the Benefits of Budget Management Software?* [Online]. Available from: <https://www.iinsight.biz/what-are-the-benefits-of-budget-management-software/> [Accessed 20 August 2022].

Paro. (2021) *The Do’s and Don’ts of Incorporating AI into Your Financial Forecasting Model* [Online]. Available from: <https://paro.ai/blog/ai-forecasting-dos-and-donts/#5don%E2%80%99ts-of-ai-forecasting> [Accessed 17 November 2022].

PocketGuard. (2022) *Pocket Guard | Your Personal Money & Bill Organizer* [Online]. Available from: <https://pocketguard.com/> [Accessed 18 October 2022].

Quicken. (2018) *Plan Your Retirement Today - Quicken* [Online]. Available from: <https://www.quicken.com/personal-finance/retirement> [Accessed 28 October 2022].

Quinn, S. (2017) “The Miracles of Bookkeeping”: How Budget Politics Link Fiscal Policies and Financial Markets. *American Journal of Sociology* 123(1), p.1. Available at: <https://www.journals.uchicago.edu/doi/full/10.1086/692461> [Accessed 4 September 2022].

ScienceSoft. (2021) *Custom Accounting Software Development: Key Steps, Success Factors, Financial Outcomes* [Online]. Available from: <https://www.scnsoft.com/financial-management/custom-accounting-software> [Accessed 25 September 2022].

Shpak, S. (2019) *Advantages & Disadvantages of Manual Accounting Systems* [Online]. Available from: <https://smallbusiness.chron.com/advantages-disadvantages-manual-accounting-systems-23862.html> [Accessed 25 August 2022].

SunAccounts. (2021) *What Are The Real Benefits of Financial Management Systems (FMS)?* [Online]. Available from: <https://sunaccounts.com/what-are-the-real-benefits-of-financial-management-systems-fms/> [Accessed 1 December 2022].

SunAccounts. (2022) *The Shortcomings of Manual Accounting (And Why You Need to Avoid Them)* [Online]. Available from: <https://sunaccounts.com/manual-accounting-shortcomings/> [Accessed 30 August 2022].

Trigo, A., Belfo, F. and Estébanez, P.R. (2016) Accounting Information Systems: Evolving towards a Business Process Oriented Accounting. *Procedia Computer Science* 100, pp. 987-94. Available at: <https://www.sciencedirect.com/science/article/pii/S1877050916324334> [Accessed 14 September 2022].

Volopay. (2022) *Advantages of using expense management analytics* [Online]. Available from: <https://www.volopay.com/expense-management/advantages-of-using-expense-management-analytics/> [Accessed 25 December 2022].

Walburg, M. (2020) *What Makes a Good App? Key Factors for Success* [Online]. Available from: <https://binarapps.com/what-makes-a-good-app-key-factors-for-success/> [Accessed 28 December 2022].

YNAB. (2008) *Features - You Need A Budget - YNAB* [Online]. Available from: <https://www.youneedabudget.com/features/> [Accessed 10 October 2022].

# 12: Appendix A: Database

## 12.1 SQL Table Creation Script

/\*

Database Name: budgetManagementDB

User Name: budgetManager

Password: user@123

\*/

--CREATE TRANSACTION TYPES TABLE

CREATE TABLE TRANSACTION\_TYPES (

    type\_id INT NOT NULL PRIMARY KEY GENERATED ALWAYS AS IDENTITY (START WITH 1, INCREMENT BY 1),

    type\_name VARCHAR(100)

);

--CREATE THEMES TABLE

CREATE TABLE THEMES (

    theme\_id INT NOT NULL PRIMARY KEY GENERATED ALWAYS AS IDENTITY (START WITH 1, INCREMENT BY 1),

    theme\_name VARCHAR(100)

);

--CREATE CURRENCIES TABLE

CREATE TABLE CURRENCIES (

    currency\_id INT NOT NULL PRIMARY KEY GENERATED ALWAYS AS IDENTITY (START WITH 1, INCREMENT BY 1),

    currency\_name VARCHAR(100),

    currency\_symbol\_code VARCHAR(100),

    conversion\_rate\_to\_usd INT,

    description LONG VARCHAR,

    is\_active BOOLEAN

);

--CREATE COUNTRIES TABLE

CREATE TABLE COUNTRIES (

    country\_id INT NOT NULL PRIMARY KEY GENERATED ALWAYS AS IDENTITY (START WITH 1, INCREMENT BY 1),

    country\_code VARCHAR(100),

    country\_name VARCHAR(100),

    country\_phone\_code VARCHAR(100),

    time\_zone VARCHAR(100),

    currency\_id INT,

    FOREIGN KEY (currency\_id) REFERENCES CURRENCIES (currency\_id)

);

--CREATE RECURRING FREQUENCY TABLE

CREATE TABLE RECURRING\_FREQUENCIES (

    recurring\_frequency\_id INT NOT NULL PRIMARY KEY GENERATED ALWAYS AS IDENTITY (START WITH 1, INCREMENT BY 1),

    frequency\_name VARCHAR(100),

    frequency\_code VARCHAR(100),

    description LONG VARCHAR,

    created\_at TIMESTAMP,

    updated\_at TIMESTAMP,

    status BOOLEAN,

    start\_date TIMESTAMP,

    end\_date TIMESTAMP,

    interval INT

);

--CREATE USERS TABLE

CREATE TABLE USERS (

    user\_id INT NOT NULL PRIMARY KEY GENERATED ALWAYS AS IDENTITY (START WITH 1, INCREMENT BY 1),

    username VARCHAR(100),

    first\_name VARCHAR(100),

    last\_name VARCHAR(100),

    email VARCHAR(100),

    password LONG VARCHAR,

    created\_date TIMESTAMP,

    status BOOLEAN,

    last\_access TIMESTAMP,

    date\_of\_birth TIMESTAMP,

    phone\_number INT,

    country\_id INT,

    city VARCHAR(100),

    address VARCHAR(100),

    gender VARCHAR(100),

    security\_question VARCHAR(200),

    security\_answer VARCHAR(200),

    two\_factor\_authentication BOOLEAN,

    credit\_score INT,

    FOREIGN KEY (country\_id) REFERENCES COUNTRIES(country\_id)

);

--CREATE CATEGORIES TABLE

CREATE TABLE CATEGORIES (

    category\_id INT NOT NULL PRIMARY KEY GENERATED ALWAYS AS IDENTITY (START WITH 1, INCREMENT BY 1),

    user\_id INT NOT NULL,

    category\_name VARCHAR(100),

    created\_at TIMESTAMP,

    updated\_at TIMESTAMP,

    status BOOLEAN,

    deleted\_date TIMESTAMP,

    FOREIGN KEY (user\_id) REFERENCES USERS(user\_id)

);

--CREATE BUDGET TABLE

CREATE TABLE BUDGETS (

    budget\_id INT NOT NULL PRIMARY KEY GENERATED ALWAYS AS IDENTITY (START WITH 1, INCREMENT BY 1),

    category\_id INT,

    start\_date TIMESTAMP,

    end\_date TIMESTAMP,

    recurring\_id INT,

    notes LONG VARCHAR,

    user\_id INT,

    last\_modified\_date TIMESTAMP,

    is\_active BOOLEAN,

    FOREIGN KEY (category\_id) REFERENCES CATEGORIES (category\_id),

    FOREIGN KEY (user\_id) REFERENCES USERS (user\_id),

    FOREIGN KEY (recurring\_id) REFERENCES RECURRING\_FREQUENCIES (recurring\_frequency\_id)

);

--CREATE ACCOUNTS TABLE

CREATE TABLE ACCOUNTS (

    account\_id INT NOT NULL PRIMARY KEY GENERATED ALWAYS AS IDENTITY (START WITH 1, INCREMENT BY 1),

    user\_id INT,

    account\_name VARCHAR(100),

    account\_type VARCHAR(100),

    account\_number VARCHAR(100),

    bank\_name VARCHAR(100),

    current\_balance INT,

    opening\_date TIMESTAMP,

    closing\_date TIMESTAMP,

    status BOOLEAN,

    notes LONG VARCHAR,

    currency\_id INT,

    last\_transaction\_date TIMESTAMP,

    FOREIGN KEY (user\_id) REFERENCES USERS (user\_id),

    FOREIGN KEY (currency\_id) REFERENCES CURRENCIES (currency\_id)

);

--CREATE BILLS TABLE

CREATE TABLE BILLS (

    bill\_id INT NOT NULL PRIMARY KEY GENERATED ALWAYS AS IDENTITY (START WITH 1, INCREMENT BY 1),

    user\_id INT NOT NULL,

    account\_id INT NOT NULL,

    category\_id INT NOT NULL,

    bill\_name VARCHAR(100),

    description LONG VARCHAR,

    amount INT,

    payment\_date TIMESTAMP,

    due\_date TIMESTAMP,

    payment\_status BOOLEAN,

    recurring\_frequency\_id INT NOT NULL,

    date\_created TIMESTAMP,

    date\_modified TIMESTAMP,

    notes LONG VARCHAR,

    FOREIGN KEY (user\_id) REFERENCES USERS(user\_id),

    FOREIGN KEY (account\_id) REFERENCES ACCOUNTS(account\_id),

    FOREIGN KEY (recurring\_frequency\_id) REFERENCES RECURRING\_FREQUENCIES(recurring\_frequency\_id)

);

--CREATE SETTINGS TABLE

CREATE TABLE SETTINGS (

    settings\_id INT NOT NULL PRIMARY KEY GENERATED ALWAYS AS IDENTITY (START WITH 1, INCREMENT BY 1),

    user\_id INT,

    default\_currency\_id INT,

    date\_format VARCHAR(100),

    email\_notifications BOOLEAN,

    SMS\_notifications BOOLEAN,

    theme\_id INT,

    updated\_date TIMESTAMP,

    FOREIGN KEY (user\_id) REFERENCES USERS(user\_id),

    FOREIGN KEY (default\_currency\_id) REFERENCES CURRENCIES(currency\_id),

    FOREIGN KEY (theme\_id) REFERENCES THEMES (theme\_id)

);

--CREATE TRANSACTIONS TABLE

CREATE TABLE TRANSACTIONS (

    transaction\_id INT NOT NULL PRIMARY KEY GENERATED ALWAYS AS IDENTITY (START WITH 1, INCREMENT BY 1),

    user\_id INT,

    transaction\_date TIMESTAMP,

    account\_id INT,

    category\_id INT,

    bill\_id INT,

    amount INT,

    type\_id INT,

    description LONG VARCHAR,

    recurring\_id INT,

    created\_date TIMESTAMP,

    updated\_date TIMESTAMP,

    status BOOLEAN,

    FOREIGN KEY (user\_id) REFERENCES USERS (user\_id),

    FOREIGN KEY (account\_id) REFERENCES ACCOUNTS (account\_id),

    FOREIGN KEY (category\_id) REFERENCES CATEGORIES (category\_id),

    FOREIGN KEY (bill\_id) REFERENCES BILLS (bill\_id),

    FOREIGN KEY (type\_id) REFERENCES TRANSACTION\_TYPES (type\_id),

    FOREIGN KEY (recurring\_id) REFERENCES RECURRING\_FREQUENCIES (recurring\_frequency\_id)

);

--CREATE SAVINGS TABLE

CREATE TABLE SAVINGS (

    savings\_id INT NOT NULL PRIMARY KEY GENERATED ALWAYS AS IDENTITY (START WITH 1, INCREMENT BY 1),

    user\_id INT,

    name VARCHAR(100),

    amount INT,

    target\_amount INT,

    start\_date TIMESTAMP,

    end\_date TIMESTAMP,

    notes LONG VARCHAR,

    created\_at TIMESTAMP,

    updated\_at TIMESTAMP,

    savings\_status VARCHAR(100),

    category\_id INT,

    recurring\_id INT,

    account\_id INT,

    budget\_id INT,

    is\_active BOOLEAN,

    FOREIGN KEY (user\_id) REFERENCES USERS(user\_id),

    FOREIGN KEY (category\_id) REFERENCES CATEGORIES(category\_id),

    FOREIGN KEY (recurring\_id) REFERENCES RECURRING\_FREQUENCIES(recurring\_frequency\_id),

    FOREIGN KEY (account\_id) REFERENCES ACCOUNTS(account\_id),

    FOREIGN KEY (budget\_id) REFERENCES BUDGETS(budget\_id)

);

--CREATE REPORT TYPES TABLE

CREATE TABLE REPORT\_TYPES (

    report\_type\_id INT NOT NULL PRIMARY KEY GENERATED ALWAYS AS IDENTITY (START WITH 1, INCREMENT BY 1),

    user\_id INT,

    report\_name VARCHAR(100),

    description LONG VARCHAR,

    created\_date TIMESTAMP,

    modified\_date TIMESTAMP,

    is\_active BOOLEAN,

    FOREIGN KEY (user\_id) REFERENCES USERS (user\_id)

);

--CREATE REPORTS TABLE

CREATE TABLE REPORTS (

    report\_id INT NOT NULL PRIMARY KEY GENERATED ALWAYS AS IDENTITY (START WITH 1, INCREMENT BY 1),

    user\_id INT,

    report\_type\_id INT,

    generated\_date TIMESTAMP,

    file\_type VARCHAR(100),

    report\_start\_date TIMESTAMP,

    report\_end\_date TIMESTAMP,

    description LONG VARCHAR,

    status BOOLEAN,

    error\_message LONG VARCHAR,

    FOREIGN KEY (user\_id) REFERENCES USERS(user\_id),

    FOREIGN KEY (report\_type\_id) REFERENCES REPORT\_TYPES(report\_type\_id)

);

--CREATE USER LOGS TABLE

CREATE TABLE USER\_LOGS (

    log\_id INT NOT NULL PRIMARY KEY GENERATED ALWAYS AS IDENTITY (START WITH 1, INCREMENT BY 1),

    user\_id INT,

    description LONG VARCHAR,

    updated\_at TIMESTAMP,

    type VARCHAR(100),

    message VARCHAR(100),

    FOREIGN KEY (user\_id) REFERENCES USERS(user\_id)

)

## 12.2 Database Tables

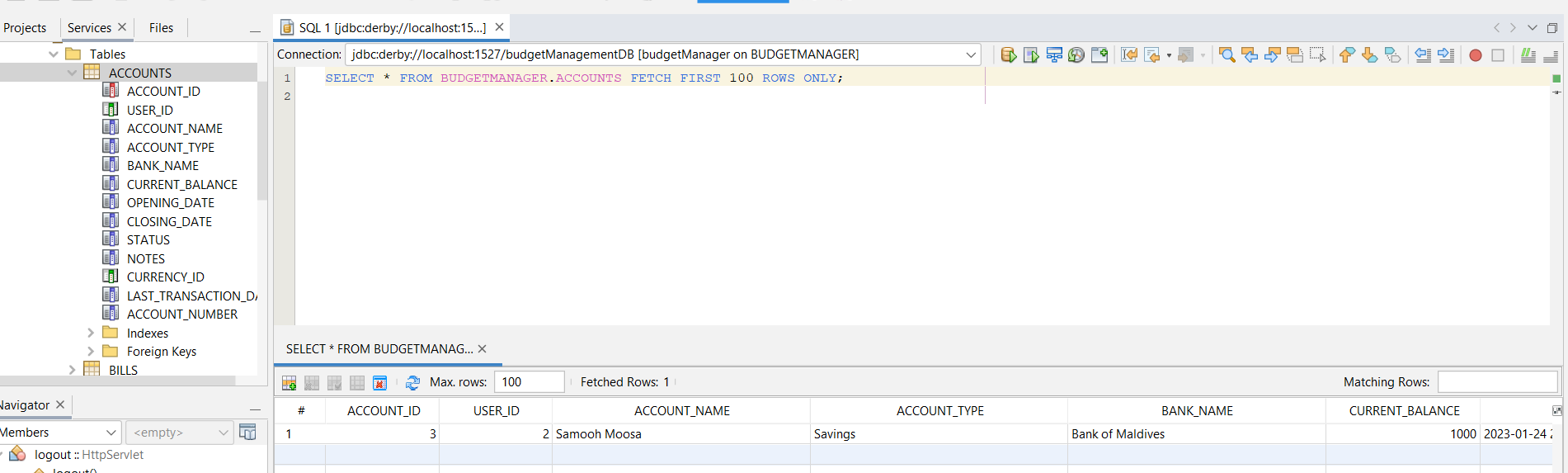


Figure 14 Accounts Table

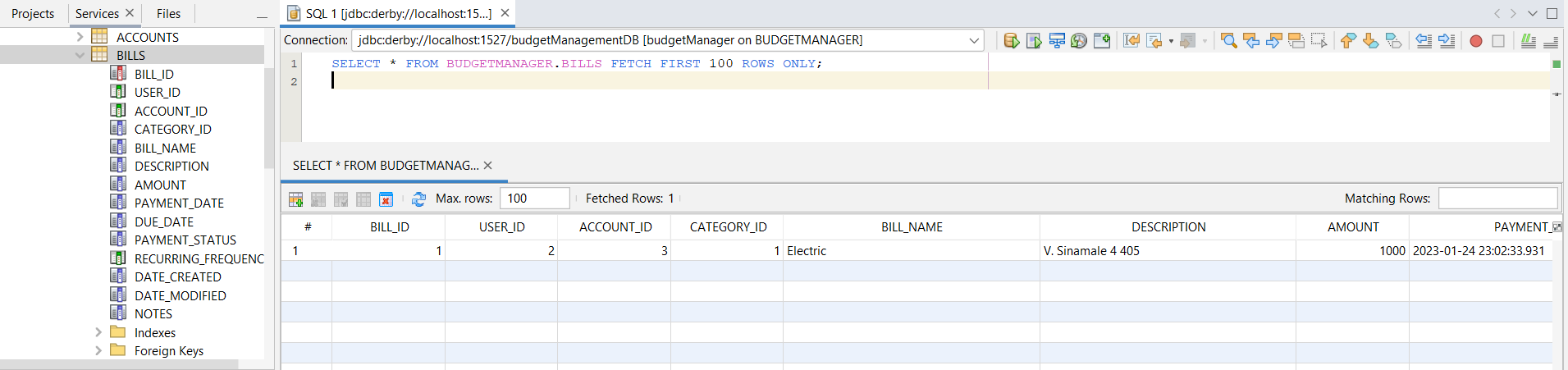


Figure 15 Bills Table

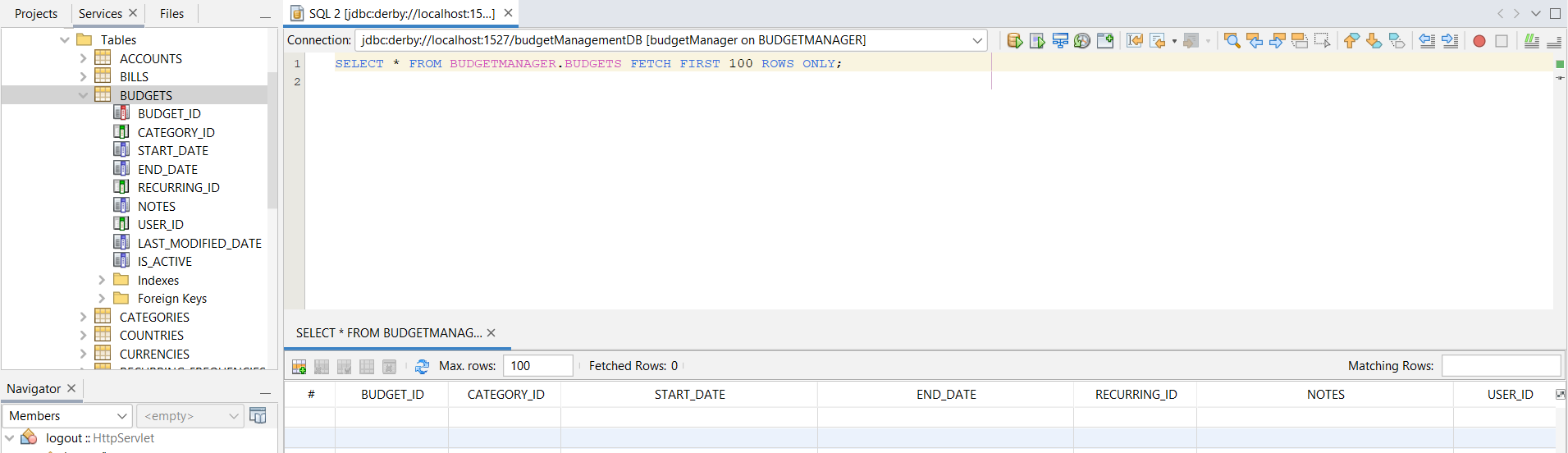


Figure 16 Budgets Table

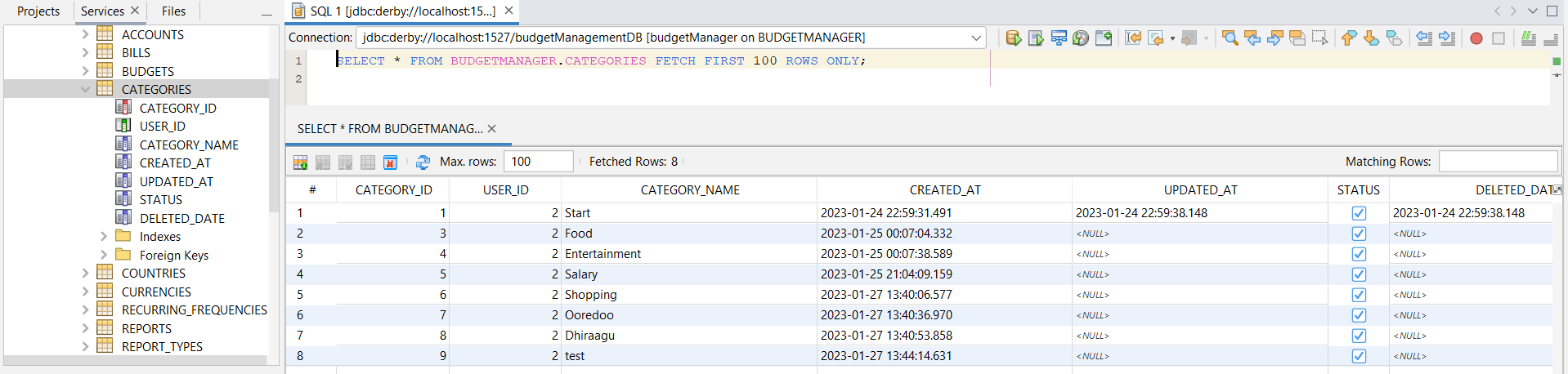


Figure 17 Categories Table

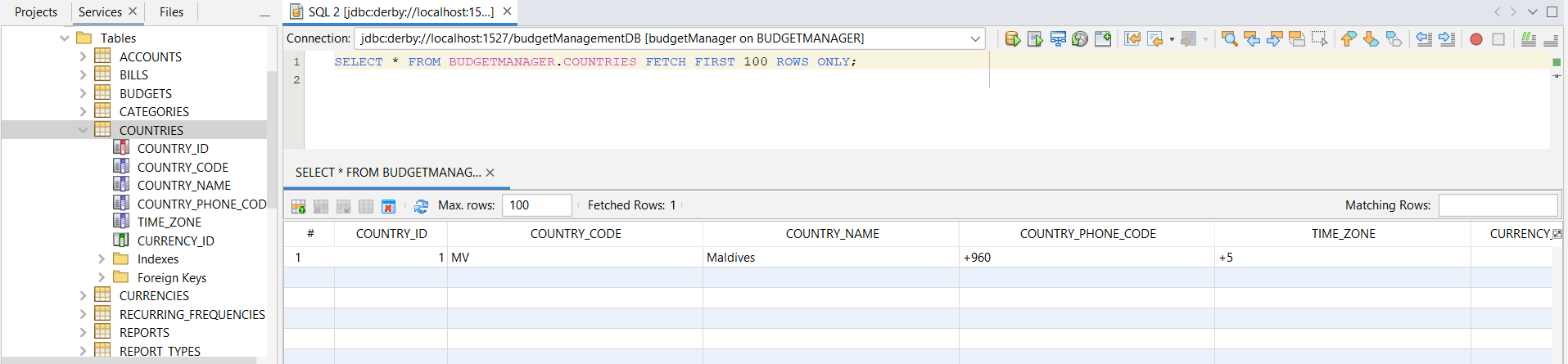


Figure 18 Countries Table

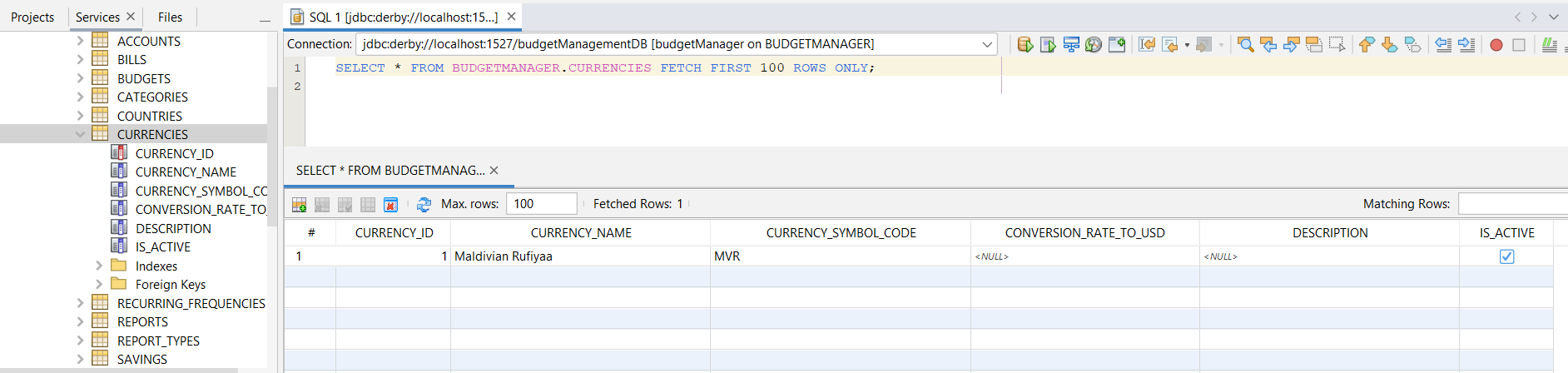


Figure 19 Currencies Table

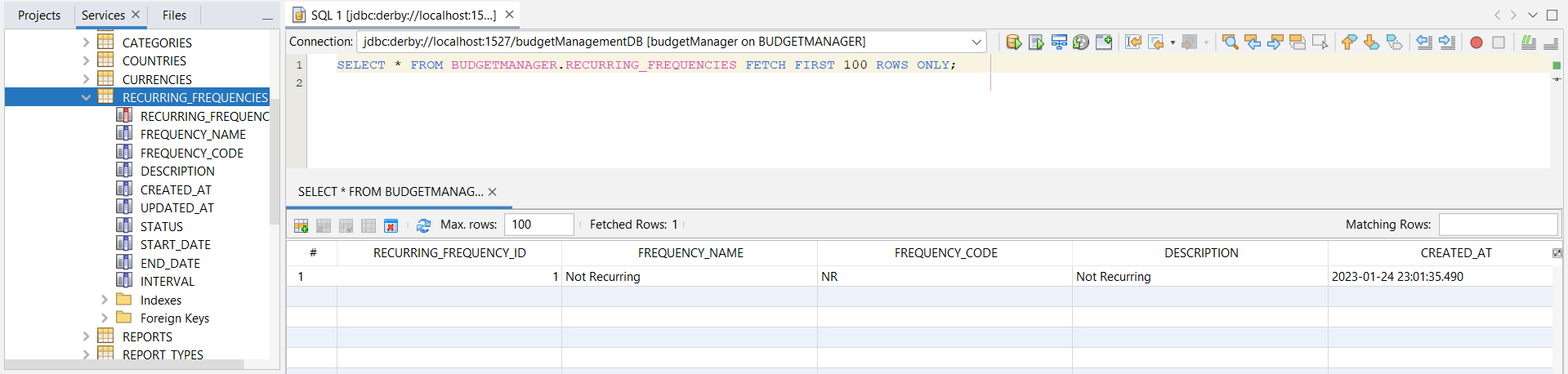


Figure 20 Recurring Frequency Table

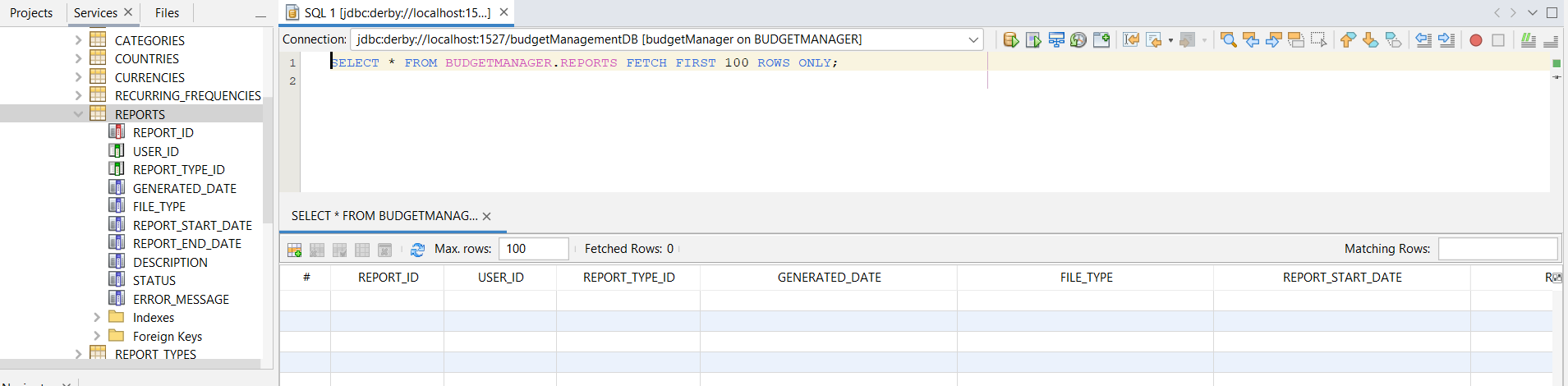


Figure 21 Reports Table

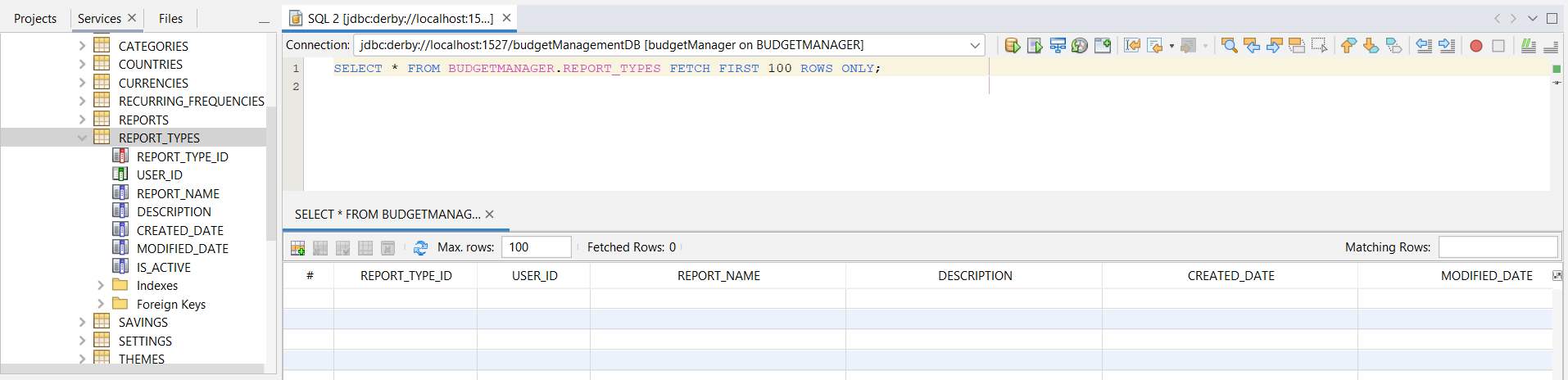


Figure 22 Report Type Table

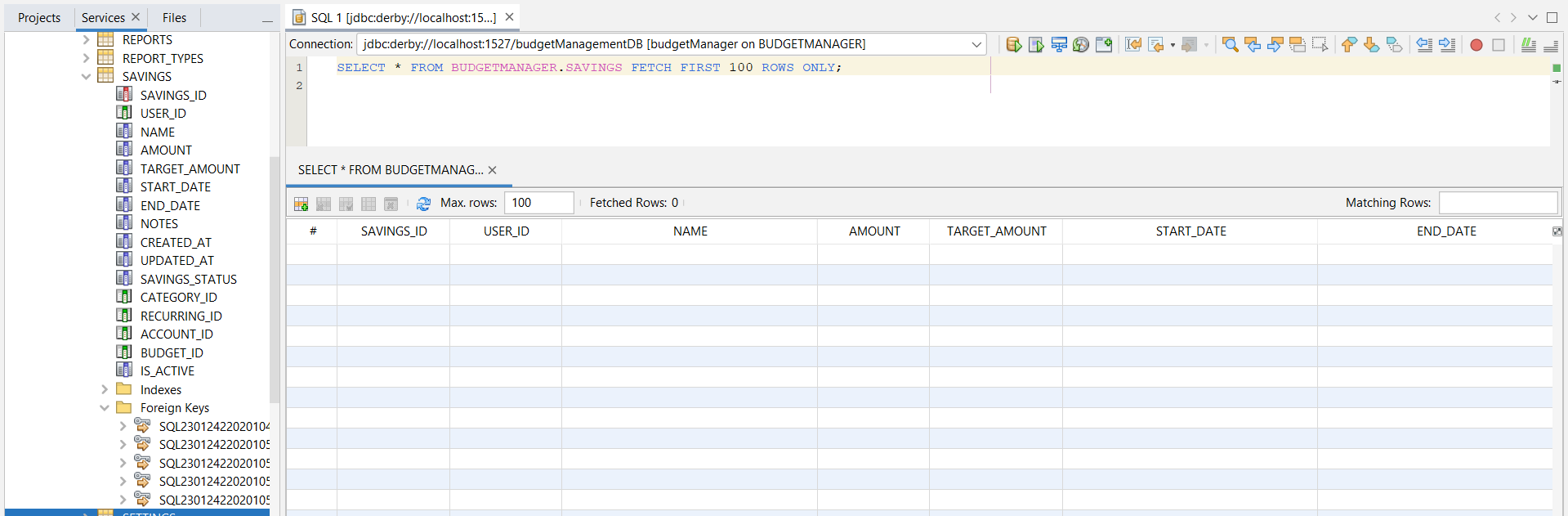


Figure 23 Savings Table

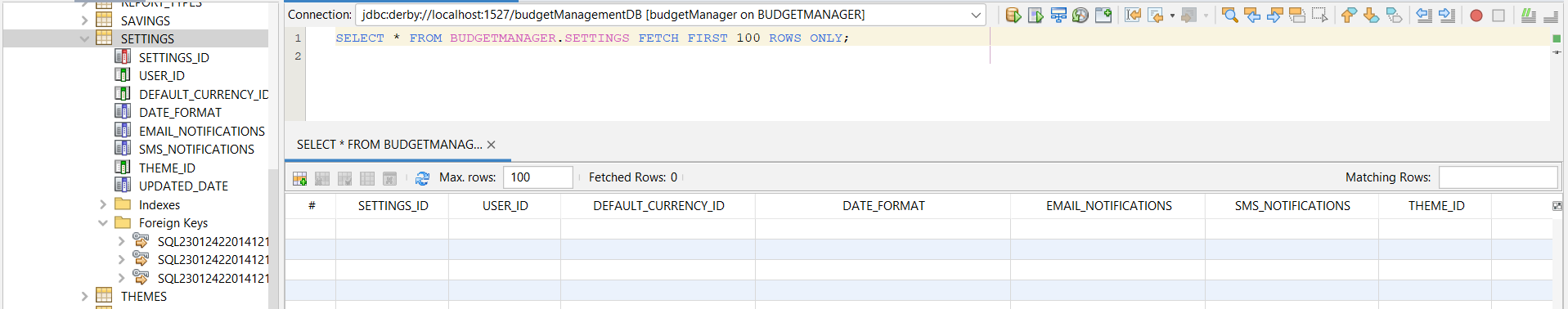


Figure 24 Settings Table

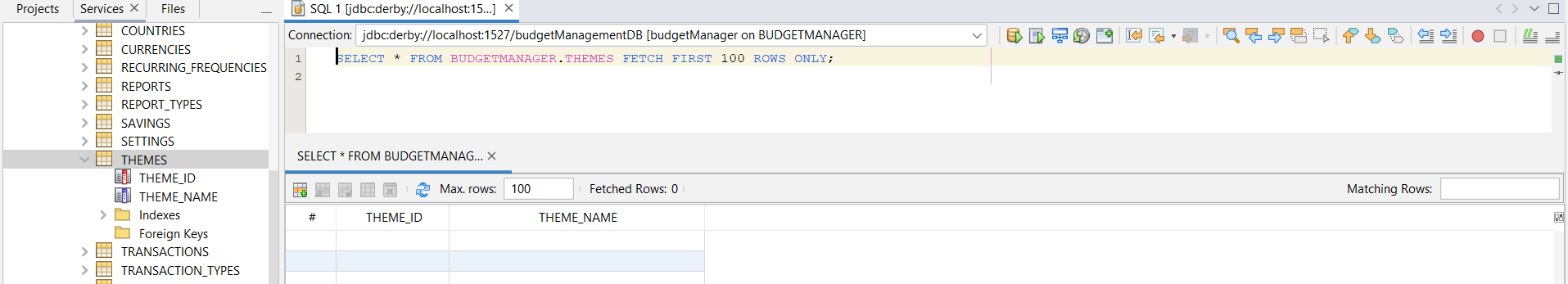


Figure 25 Themes Table

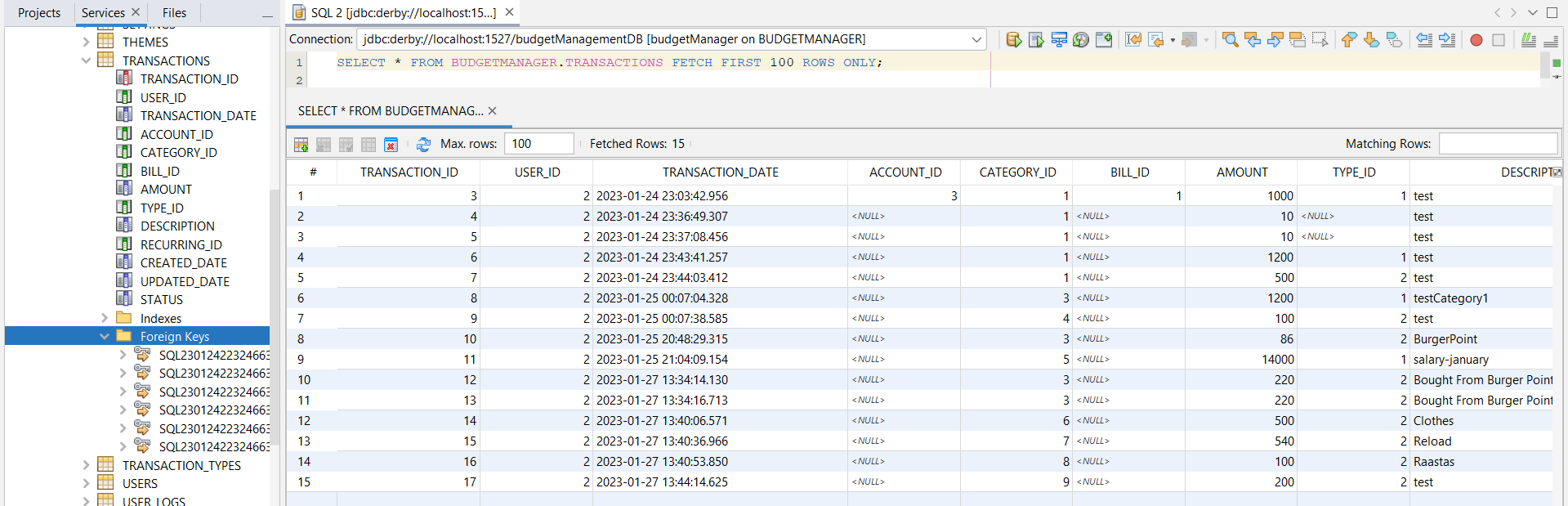


Figure 26 Transactions Table

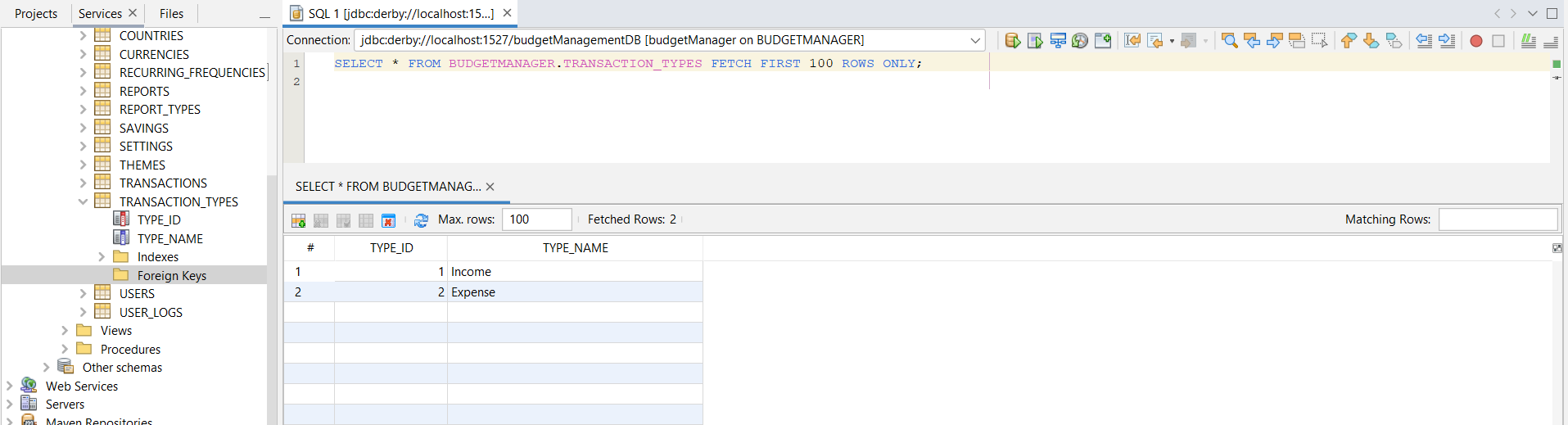


Figure 27 Transaction Types Table

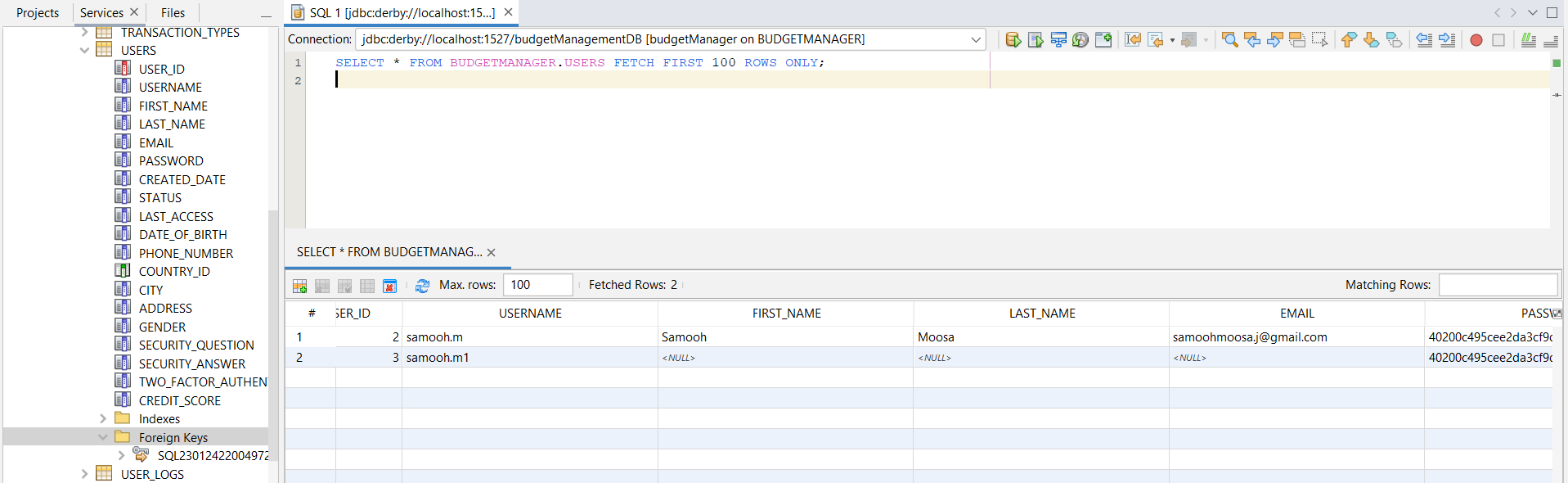


Figure 28 Users Table

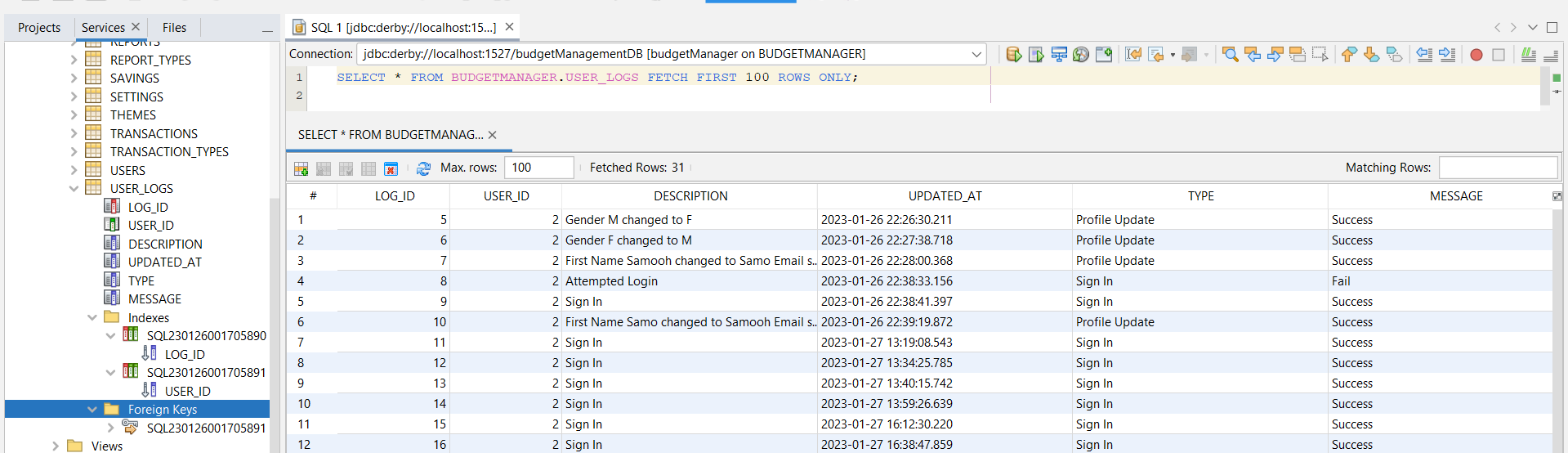


Figure 29 User Logs Table

# 13: Appendix B: Diagrams

## 13.1 ER Diagram

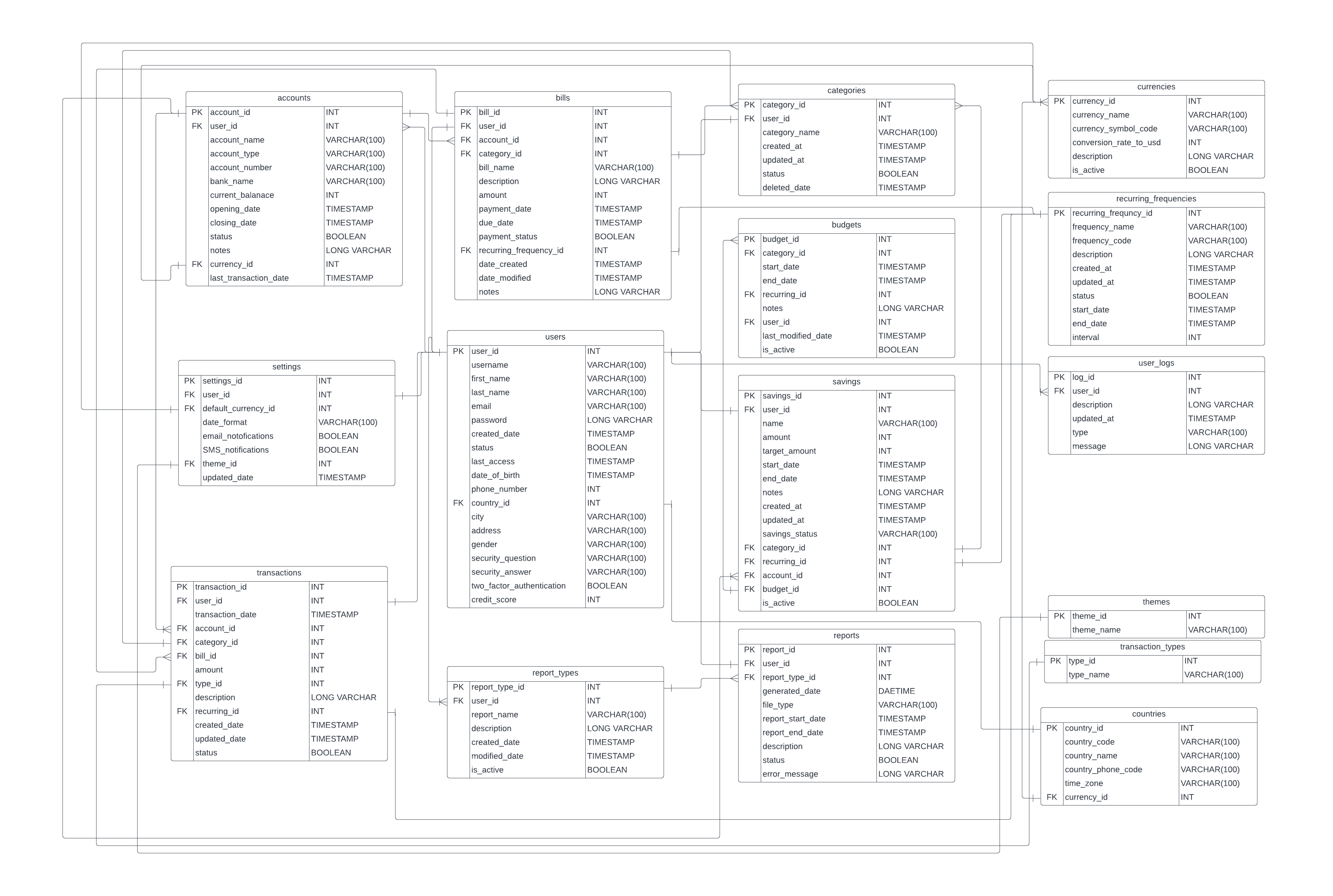


Figure 30 ER Diagram (Complete)

## 13.2 Use Case Diagram

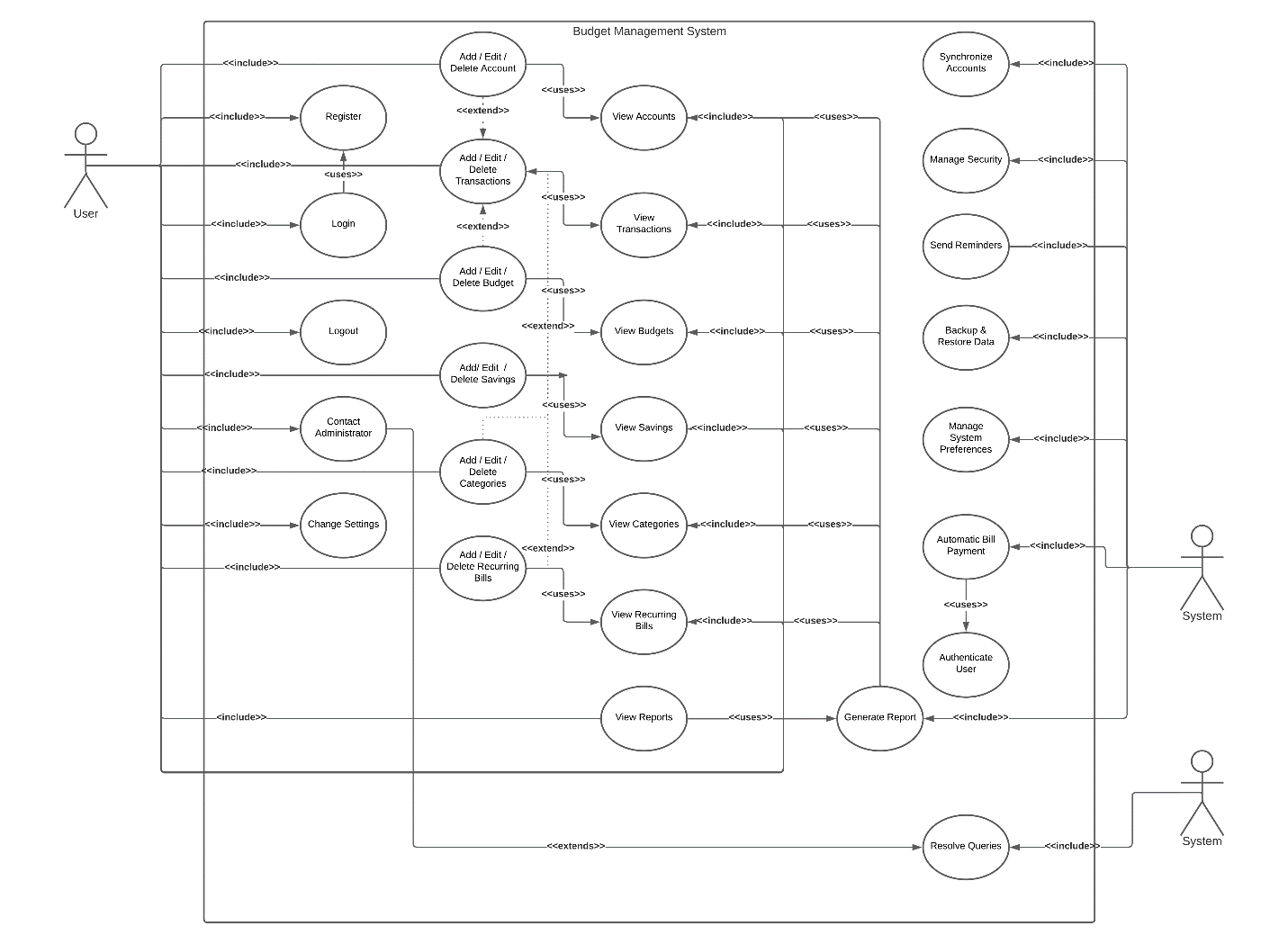


Figure 31 Use Case Diagram

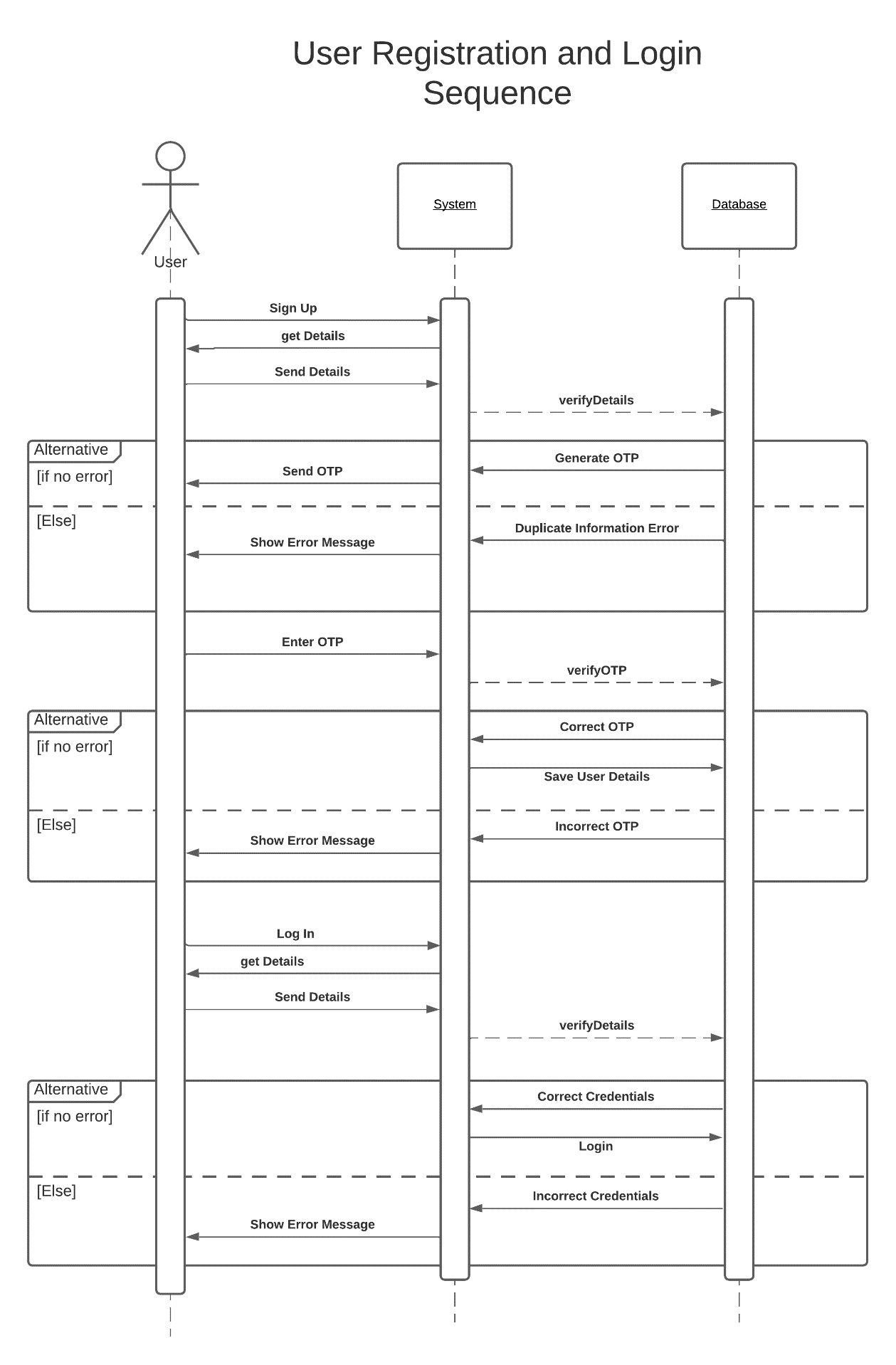


Figure 32 User Registration & Login Sequence Diagram