



BSCS-F22-018

03-134192-007 AHSAN FAROOQ

03-134182-077 M MUZAMMIL MAHMOOD SINDHU

Expense Management System

In partial fulfilment of the requirements for the degree of
Bachelor of Science in Computer Science

Supervisor: Mouzna Tahir

Department of Computer Sciences
Bahria University, Lahore Campus

June 2023

Certificate



We accept the work contained in the report titled

“Expense Management System”

written by

AHSAN FAROOQ

M MUZAMMIL MAHMOOD SINDHU

as a confirmation to the required standard for the partial fulfilment of the degree of
Bachelor of Science in Computer Science.

Approved by:

Supervisor: MOUZNA TAHIR

(Signature)

June 20th, 2023

DECLARATION

We hereby declare that this project report is based on our original work except for citations and quotations which have been duly acknowledged. We also declare that it has not been previously and concurrently submitted for any other degree or award at Bahria University or other institutions.

Enrolment	Name	Signature
03-134192-007	AHSAN FAROOQ	
03-134182-077	M MUZAMMIL MAHMOOD SINDHU	

Date : June 20th, 2023

Specially dedicated to
my beloved grandmother, grandfather, mother, father, and my friends

(AHSAN FAROOQ)

my beloved grandmother, grandfather, mother, and father

(M MUZAMMIL MAHMOOD SINDHU)

ACKNOWLEDGEMENTS

We would like to thank everyone who had contributed to the successful completion of this project. We would like to express our gratitude to our research supervisor, Miss MOUZNA TAHIR for her invaluable advice, guidance and her enormous patience throughout the development of the research.

In addition, we would also like to express our gratitude to our loving parent and friends who had helped and given me encouragement.

AHSAN FAROOQ
M MUZAMMIL MAHMOOD SINDHU

Expense Management System

ABSTRACT

The "Expense Management System" project. Online financial management tools have made it easier to handle everyday personal finances. This project's main objective is to develop an online spending management solution. The Expense Management System's goal is to manage our everyday spending more effectively and affordably, as the name of the initiative indicates. Users occasionally forget where they spend their money. Users struggle in managing their financial flow. Users must find a remedy for this issue that allows everyone to control their spending. Considering this, we decided to look for a simpler solution to the issue.

As a result, the Expense Management System tries to minimize the load of manual computation on the user while keeping track of expenses. Instead of keeping a journal or cost log, this web application allows the user to create and preserve reports in addition to maintaining control over their spending. The user of this program may manage their monthly, weekly, and daily spending with the use of this tool. Users may manage their income and expenses, create, and store their reports, as well as insert and remove transactions. The major component of the system is the graphical representation of the program it appeals to users more and is simpler to grasp.

TABLE OF CONTENTS

DECLARATION	ii
ACKNOWLEDGEMENTS	iv
ABSTRACT	v
TABLE OF CONTENTS	vi
LIST OF TABLES	x
LIST OF FIGURES	xi

CHAPTERS

1	INTRODUCTION	1
	1.1 Background	1
	1.2 Problem Statements	2
	1.3 Objectives	3
	1.4 Scope of Project	3
	1.4.1 Optional Scope	3
2	SOFTWARE REQUIREMENT SPECIFICATION	5
	2.1 User Classes and Characteristics	5
	2.1.1 Regular User	5
	2.2 Operating Environment	5
	2.3 Functional Requirements	6
	2.4 Non-Functional Requirements	6
	2.5 Design and Implementation Constraints	6
	2.6 Assumption and Dependencies of the Expense Management System	7

2.7	Use-Case Diagrams	8
2.7.1	System Use Case Diagram	8
2.7.2	Sign-up Use Case	8
2.7.3	Login Use Case	9
2.8	Use-Case Description Table	9
2.8.1	Sign-In/Login	9
2.8.2	Add Expense	10
2.8.3	View Expense	11
2.8.4	Edit Expense	11
2.8.5	Delete Expense	12
2.8.6	Create Expense Report	13
2.8.7	View Expense Report	14
2.8.8	Update Expense Report	14
2.8.9	Delete Expense Report	15
2.8.10	Approve Expense Report	16
2.9	System Requirement Chart	17
2.10	Other Non-functional Requirements	18
2.10.1	Performance Requirements	18
2.10.2	Safety Requirements	18
2.10.3	Security Requirements	19
2.10.4	Software Quality Attributes	19
3	DESIGN AND METHODOLOGY	21
3.1	UML Diagrams	21
3.1.1	Use Case Diagram	21
3.1.2	Domain Model	22
3.1.3	Sequence Diagram	22
3.1.4	Collaboration Diagram	26
3.1.5	Class Diagram	28
3.1.6	ERD Diagram	29
3.2	Methodology	29
3.2.1	Build Feature List	30
3.2.2	Sprint Planning	30

3.2.3	Testing and Sprint demonstration	30
3.2.4	Retrospective and Next Spring Planning	30
4	DATA AND IMPLMENTATION	31
4.1	Implementation	31
4.1.1	Planning and designing	32
4.1.2	Implementation	32
4.1.3	Testing	32
4.1.4	Documentation	32
4.1.5	Deployment and Maintenance	33
4.1.6	Technical Aspects of Expense Management System	33
4.2	Hardware	33
4.3	Front-end	33
4.4	Back-end	33
5	USER MANUAL	34
5.1	General Information	34
5.2	Overview	34
5.3	Organization of the Manual	34
5.4	System Configuration	35
5.5	User Access Level	35
5.6	Contingencies	35
5.7	Understanding the Expense Management System	35
5.7.1	Sign Up	35
5.7.2	Login	36
5.7.3	Add Bank Account	37
5.7.4	Add Income	37
5.7.5	Add Expense	38
5.7.6	Transfer	38
5.7.7	Transaction	39
5.7.8	Calendar	39

6	CONCLUSION AND RECOMMENDATIONS	40
6.1	Conclusion	40
6.2	Recommendation	40
6.2.1	Strong Internet Connection	40
	REFERENCES	41
	APPENDICES	42
	TURNITIN REPORT	43

LIST OF TABLES

TABLE	TITLE	PAGE
Table 2.1:	Use-Case 1	9
Table 2.2:	Use-Case 2	10
Table 2.3:	Use-Case 3	11
Table 2.4:	Use-Case 4	11
Table 2.5:	Use-Case 5	12
Table 2.6:	Use-Case 6	13
Table 2.7:	Use-Case 7	14
Table 2.8:	Use-Case 8	14
Table 2.9:	Use-Case 9	15
Table 2.10:	Use-Case 10	16
Table 2.11:	System Requirement Chart	17

LIST OF FIGURES

FIGURE	TITLE	PAGE
Figure 3.1:	Use–Case Diagram	21
Figure 3.2:	Domain Model	22
Figure 3.3:	User Registration	23
Figure 3.4:	User Login	23
Figure 3.5:	Add Income	24
Figure 3.6:	User Login	24
Figure 3.7:	View Income	25
Figure 3.8:	View Expense	25
Figure 3.9:	User Registration	26
Figure 3.10:	User Login	26
Figure 3.11:	Add Income	27
Figure 3.12:	Add Expense	27
Figure 3.13:	View Report	28
Figure 3.14:	Class Diagram	28
Figure 3.15:	ERD Diagram	29
Figure 3.16:	Methodology	29
Figure 4.1:	Feature Driven Development Diagram	31
Figure 5.1:	Sign Up	36
Figure 5.2:	Login	36

Figure 5.3: Add Bank Account	37
Figure 5.4: Add Income	37
Figure 5.5: Add Expense	38
Figure 5.6: Transfer	38
Figure 5.7: Transaction	39
Figure 5.8: Calendar	39

CHAPTER 1

INTRODUCTION

1.1 Background

Throughout time, the management of expenses has emerged as a crucial element in both personal and professional spheres. As financial transactions have become more intricate and the necessity for precise monitoring and budgeting has escalated, it has become clear that conventional approaches to expense management, such as paper receipts and spreadsheets, are inadequate. In acknowledgment of this escalating demand, our team has created a ground-breaking web application for expense management, aiming to streamline and uncomplicate the task of managing expenses for individuals and organizations alike.

Before developing the web application, we conducted extensive research to identify the most common pain points and challenges users face in managing their expenses. In finding the aspects found that the major hurdles included time-consuming manual data entry, difficulty in organizing and categorizing expenses, limited access and collaboration among team members, and lack of insightful analysis to inform budgeting decisions.

To address these issues, our team set out to create a user-friendly, efficient, and comprehensive expense management system that would not only automate many of the time-consuming tasks but also provide valuable insights into spending patterns and trends. The Expense Management System application is focused on seamless

integration, intuitive user experience, and compatibility with various devices and platforms, ensuring that users can easily access and manage their expenses anytime, anywhere.

Throughout the development process, we consulted with experts in finance, user experience, and web development, ensuring that our application meets the highest standards of quality and functionality. Additionally, user testing is conducted with a diverse group of participants to gather feedback and make necessary refinements to the system.

Our expense management system web application is now ready for deployment, providing an all-in-one solution for individuals and organizations looking to take control of their finances and optimize their expense management processes. With this powerful tool in hand, users can now easily track, categorize, and analyse their expenses, collaborate with team members, and make data-driven decisions to improve their financial well-being.

1.2 Problem Statements

In traditional expense management methods, individuals and organizations must spend a significant amount of time manually entering expense data, leading to inefficiencies, inaccuracies, and delays in financial reporting and decision-making. This labour-intensive process hampers productivity and increases the likelihood of human error.

As the number of transactions grows, it becomes increasingly challenging to organize and categorize expenses accurately. This issue can lead to mismanagement of funds, incorrect budget allocation, and an inability to identify areas for cost-saving and optimization.

With the increasing use of mobile devices and a variety of platforms, expense management systems need to be compatible with different devices and operating

systems. However, many existing solutions are not designed for this level of accessibility, limiting users' ability to track and manage their expenses on the go.

1.3 Objectives

The primary objective of our expense management system web application is to provide a comprehensive and user-friendly solution that addresses the key challenges faced by individuals and organizations in managing their expenses. To achieve these objectives, we have outlined the following:

- i) Record expenses, income, and transfers.
- ii) Simplify and Automate Data
- iii) Ensure Compatibility with Multiple Devices and Platforms
- iv) Maintain Security and Data Privacy
- v) To add weekly, monthly, yearly, and invoices or billing expenses.
- vi) Improved Calendar Visuals Review all your monthly.
- vii) To give a visualized view of daily, monthly, and yearly to the user.

1.4 Scope of Project

The expense management system web application project focuses on key areas such as user management, expense tracking, budgeting, reporting, mobile compatibility, and security. It aims to streamline financial processes, enable users to track and categorize expenses, set budgets, generate reports, and access the system from various devices. Security measures will be implemented to protect user data and ensure privacy.

1.4.1 Optional Scope

The expense management system web application project will focus on delivering a user-friendly experience through intuitive menu navigation. It will be built on a well-designed architecture that ensures reliability and scalability for future enhancements.

The application will also feature a responsive layout, allowing users to access and interact with it seamlessly across different devices. These aspects will contribute to an enhanced user experience and efficient utilization of the system's features.

CHAPTER 2

SOFTWARE REQUIREMENT SPECIFICATION

2.1 User Classes and Characteristics

The expense management system will consist of two user classes: regular users and administrators.

2.1.1 Regular User

1. Regular users can log in to the system and access their dashboard.
2. Regular users can add, edit, and delete expenses, as well as add receipts and comments to their expenses.
3. Regular users can view their expenses history, including details such as date, category, and amount spent.
4. Regular users can generate reports on their expenses.

2.2 Operating Environment

The expense management system will be accessible through a web browser and will be hosted on a web server. The system will be compatible with the following web browsers:

1. Google Chrome

2. Mozilla Firefox
3. Microsoft Edge
4. Safari

2.3 Functional Requirements

The following are the functional requirements of the expense management system:

1. User registration and login functionality.
2. Expense creation, editing, and deletion functionality.
3. Budgeting and Financial Planning
4. Expense categorization functionality.
5. Reporting and Analytics.
6. User profile management functionality.

2.4 Non-Functional Requirements

The following are the non-functional requirements of the expense management system:

1. Usability
2. Performance
3. Security and Data Privacy
4. Reliability
5. Scalability
6. Maintainability
7. Portability

2.5 Design and Implementation Constraints

The expense management system is subject to the following design and implementation constraints:

- 1. Front-end Development:** The expense management system's user interface will be designed using HTML, CSS, JavaScript, and ReactJS.
- 2. Back-end Development:** The system will be developed using a combination of NodeJS and MongoDB for the backend database.
- 3. Compatibility:** The system will be accessible through a web browser and will be compatible with modern web browsers such as Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari.
- 4. Security:** To ensure the safety and privacy of user and administrator data, it is essential to establish robust security measures that effectively prevent unauthorized access.
- 5. Development Time:** The development process may be constrained by time limitations, which may impact the features and functionality that can be included in the final product.

2.6 Assumption and Dependencies of the Expense Management System

The expense management system is based on the following assumptions and dependencies:

- 1. Internet Connections:** The system assumes that the user and administrator have access to a reliable internet connection to access the server and retrieve data.
- 2. Hardware Failure:** The system assumes that the user's device and server hardware will function correctly. Any hardware failure may affect the system's availability and reliability.
- 3. User Data Input:** The system assumes that the user will input accurate and complete data. Any incorrect or incomplete data may result in errors or incorrect calculations in the system.
- 4. User Data Input:** The system assumes that the user will input accurate and complete data. Any incorrect or incomplete data may result in errors or incorrect calculations in the system.

2.7 Use-Case Diagrams

2.7.1 System Use Case Diagram

The expense management system will have two types of users: regular users and administrators. The following use case diagram outlines the different actions that each user can perform:

Regular User:

1. **Registration:** Users can create a new account within the system.
2. **Login:** Users can access the system by logging in with their respective credentials.
3. **View Expenses:** The user can view a list of their expenses.
4. **Add Expenses:** The user can add a new expense to their list.
5. **Update Expenses:** The user can edit or update an existing expense.
6. **Delete Expenses:** The user can delete an expense from their list.
7. **View Reports:** The user can view reports on their expenses.
8. **Logout:** The user can log out of the system.

2.7.2 Sign-up Use Case

The Sign-Up process is essential in the expense management system, enabling users to create an account and gain access to all application features. The following use case diagram illustrates the various actions associated with the Sign-Up process.

1. **Enter Name:** The user enters their complete name.
2. **Enter Email:** The user enters their email address.
3. **Enter Phone:** The users enter their phone number.
4. **Enter Password:** The user enters a password for their account.
5. **Confirm Password:** The user confirms their password.
6. **Create Account:** The user creates their account by submitting the form.

Once the user has created their account, they can log in to the system and access all the features available to them.

2.7.3 Login Use Case

The Sign-In/Login process is a crucial part of the expense management system, as it allows users and admin to access their accounts and use the features of the application. The following use case diagram outlines the different actions involved in the Sign-In/Login process:

1. **Enter Email:** The user/admin enters their email address.
2. **Enter Password:** The user/admin enters the password for their account.
3. **Submit Login:** The user/admin submits the form to log in to the system.

Once the user/admin has logged in, they can access their account and use the features available to them.

2.8 Use-Case Description Table

2.8.1 Sign-In/Login

Table 2.1: Use-Case 1

1	Unique Identifier	U1
2	Objective	To allow user/admin to access their account and use the features of the application
3	Priority	High
4	Source	User/Admin
5	Actors	User/Admin
6	Flow of events	<p>Basic flow:</p> <ul style="list-style-type: none"> • User/Admin will enter their email and password. • System will validate the user's credentials. • After the user logged in, the user can access the feature provided to them. <p>Alternative flow:</p>

		<ul style="list-style-type: none"> User/Admin provides an incorrect email or password. <p>The user or administrator is prompted to input their credentials again when the system shows an error message.</p>
7	Includes	None
8	Pre-condition	User/Admin must have a registered account
9	Post-condition	User/Admin is logged into the system and can access their account and features.

2.8.2 Add Expense

Table 2.2: Use-Case 2

1	Unique Identifier	U2
2	Objective	To allow the User to add a new expense to their account
3	Priority	High
4	Source	User
5	Actors	User
6	Flow of events	<p>Basic flow:</p> <ul style="list-style-type: none"> User selects the "Add Expense" option from the menu. User enters the details of the expense (date, amount, category, description) System validates the data and adds the new expense to the User's account. <p>Alternative flow:</p> <ul style="list-style-type: none"> User enters incorrect data or misses a required field. <p>The user or administrator is prompted to input their data again when the system encountered an error message.</p>
7	Includes	None
8	Pre-condition	Users must be logged into their account
9	Postcondition	A new expense is added to the User's account.

2.8.3 View Expense

Table 2.3: Use-Case 3

1	Unique Identifier	U3
2	Objective	To allow the User to view their expenses and filter them by date and category
3	Priority	High
4	Source	User
5	Actors	User
6	Flow of events	<p>Basic flow:</p> <ul style="list-style-type: none"> • User selects the "View Expenses" option from the menu. • System displays a list of all the User's expenses, sorted by date. • User can filter the expenses by date range or category. <p>Alternative flow:</p> <ul style="list-style-type: none"> • User has no expenses in their account. The system displays a message indicating that there are no expenses to view.
7	Includes	None
8	Pre-condition	Users must be logged into their accounts and have expenses in their account
9	Post-condition	User can view their expenses and filter them by date and category.

2.8.4 Edit Expense

Table 2.4: Use-Case 4

1	Unique Identifier	U4
2	Objective	To allow the User to edit an existing expense in their account
3	Priority	High
4	Source	User
5	Actors	User
6	Flow of events	<p>Basic flow:</p> <ul style="list-style-type: none"> • User selects the "Edit Expense" option from the menu.

		<ul style="list-style-type: none"> • System displays a list of the User's expenses. • User selects the expense they wish to edit. • User makes changes to the expense details (date, amount, category, description) • System validates the data and updates the expense in the User's account. <p>Alternative flow:</p> <ul style="list-style-type: none"> • User enters incorrect data or misses a required field. <p>The user or administrator is prompted to input their data again when the system encountered an error message.</p>
7	Includes	None
8	Pre-condition	Users must be logged into their accounts and have expenses in their account
9	Post-condition	The selected expense is updated with the new details in the User's account.

2.8.5 Delete Expense

Table 2.5: Use-Case 5

1	Unique Identifier	U5
2	Objective	To allow the User to delete an existing expense from their account
3	Priority	High
4	Source	User
5	Actors	User
6	Flow of events	<p>Basic flow:</p> <ul style="list-style-type: none"> • User selects the "Delete Expense" option from the menu. • System displays a list of the User's expenses. • User selects the expense to delete it. • System shows the confirmation dialog to the user. • User confirms the deletion. • System removes the expense from the User's account.

		<p>Alternative flow:</p> <ul style="list-style-type: none"> • User cancels the deletion. <p>The system returns the User to the list of expenses without deleting the selected expense.</p>
7	Includes	None
8	Pre-condition	Users must be logged into their accounts and have expenses in their account
9	Post-condition	The selected expense is removed from the User's account.

2.8.6 Create Expense Report

Table 2.6: Use-Case 6

1	Unique Identifier	U6
2	Objective	To let users, create an expense report
3	Priority	High
4	Source	User
5	Actors	User
6	Flow of events	<p>Basic flow:</p> <ul style="list-style-type: none"> • User logs into the system • User navigates to the "Create Expense Report" screen. • User enters the necessary details for the expense report (e.g., date, description, category, amount) • User saves the expense report. • System validates the expense report and saves it to the database. <p>Alternative flow:</p> <p>The user cancels the creation of the expense report</p>
7	Includes	None
8	Pre-condition	The user must be logged in
9	Post-condition	An expense report is created and saved to the database

2.8.7 View Expense Report

Table 2.7: Use-Case 7

1	Unique Identifier	U7
2	Objective	To let users, view expense reports
3	Priority	High
4	Source	User
5	Actors	User
6	Flow of events	<p>Basic flow:</p> <ul style="list-style-type: none"> • User logs into the system • User navigates to the "View Expense Report" screen. • User selects the desired expense report from the list. • System retrieves the expense report details from the database and displays them on the screen. <p>Alternative flow:</p> <ul style="list-style-type: none"> • User cancels the viewing of the expense report.
7	Includes	None
8	Pre-condition	The user must be logged in
9	Post-condition	Expense report details are displayed on the screen

2.8.8 Update Expense Report

Table 2.8: Use-Case 8

1	Unique Identifier	U8
2	Objective	To let users, update expense reports
3	Priority	High
4	Source	User
5	Actors	User
6	Flow of events	<p>Basic flow</p> <ul style="list-style-type: none"> • User logs into the system • User navigates to the "Update

		<p>Expense Report" screen.</p> <ul style="list-style-type: none"> • User selects the desired expense report from the list. • User updates the necessary details for the expense report (e.g., date, description, category, amount) • User saves the updated expense report. • System validates the updated expense report and saves it to the database. <p>Alternative flow:</p> <ul style="list-style-type: none"> • User cancels the updating of the expense report.
7	Includes	None
8	Pre-condition	The user must be logged in
9	Post-condition	The expense report is updated and saved to the database

2.8.9 Delete Expense Report

Table 2.9: Use-Case 9

1	Unique Identifier	U9
2	Objective	To let users, delete expense reports
3	Priority	High
4	Source	User
5	Actors	User
6	Flow of events	<p>Basic flow</p> <ul style="list-style-type: none"> • User logs into the system • User navigates to the "Delete Expense Report" screen. • User selects the desired expense report from the list. • User confirms the deletion of the expense report. • System deletes the expense report from the database. <p>Alternative flow:</p>

		<ul style="list-style-type: none"> User cancels the deletion of the expense report.
7	Includes	None
8	Pre-condition	The user must be logged in
9	Post-condition	The expense report is deleted from the database

2.8.10 Approve Expense Report

Table 2.10: Use-Case 10

1	Unique Identifier	U10
2	Objective	To let managers approve or reject expense reports
3	Priority	High
4	Source	Manager
5	Actors	Manager
6	Flow of events	<p>Basic flow</p> <ul style="list-style-type: none"> Manager logs into the system Manager navigates to the "Approve Expense Reports" screen. Manager selects the desired report from the list of reports. Manager reviews the details of the expense report. Manager approves or rejects the expense report. System updates the status of the report to "Approved" or "Rejected". <p>Alternative flow:</p> <ul style="list-style-type: none"> Manager cancels the approval or rejection of the expense report.
7	Includes	None
8	Pre-condition	The manager must be logged in and have the appropriate permissions to approve or reject expense reports
9	Post-condition	The expense report is either approved or rejected and the status is updated in the system.

2.9 System Requirement Chart

Table 2.11: System Requirement Chart

Requirement No.	Priority	Type	Source	Contained in USE-CASE	Description
EMR 1	High	Functional	User	U1	Users can create an expense report by entering necessary details like date, description, category, and amount.
EMR 2	High	Functional	User	U2	User can view their expense reports.
EMR 3	High	Functional	User/Admin	U3	Users and admins can edit or delete an expense report.
EMR 4	High	Functional	User/Admin	U4	Users and admins can view all expense reports.
EMR 5	High	Functional	User	U5	User can export their expense report as a CSV file.
EMR 6	High	Functional	User	U6	User can categorize their expense report by creating custom categories.
EMR 7	High	Functional	User/Admin	U7	Users and admin can generate monthly, yearly, or custom date range expense reports.
EMR 8	High	Functional	User/Admin	U8	Users and admins can search for a specific expense report by entering relevant keywords.
EMR 9	High	Functional	User/Admin	U9	Users and admins can add receipts or images as evidence for an expense report.
EMR 10	High	Non-functional	N/A	N/A	Server response time should be fast and efficient.
EMR 11	High	Non-functional	N/A	N/A	Along with the implementation phase, project code components

					will be evaluated to make sure they are operational.
EMR 12	High	Non-functional	N/A	N/A	Testing will be done on the finished integrated project code to make sure everything works as intended and is smoothly merged.
EMR 13	High	Non-functional	N/A	N/A	All user data should be encrypted.
EMR 14	High	Non-functional	N/A	N/A	Verification codes should be unique and secure.
EMR 15	High	Non-functional	N/A	N/A	One email should be associated with only one expense management system account.
EMR 16	High	Non-functional	N/A	N/A	Design a scalable architecture to handle many users and expense reports.

2.10 Other Non-functional Requirements

2.10.1 Performance Requirements

The system should respond quickly to user inputs, with each transaction taking no more than 2 seconds to process. For batch operations or large queries, the system should provide a clear progress indicator and estimated time for completion.

2.10.2 Safety Requirements

- 1. Data Security:** All sensitive data, including user credentials, financial information, and personal identification information, should be encrypted both in transit and at rest using industry-standard methods.
- 2. Access Control:** To ensure secure and controlled access, the system should incorporate role-based access control (RBAC) functionality. This implementation will restrict users to accessing only the functions and data that are relevant to their assigned roles. Unauthorized access to any part of the

system will be effectively prevented, maintaining the integrity and confidentiality of sensitive information.

3. **Authentication:** All users should be authenticated before accessing the system. Implement two-factor authentication for added security.
4. **Authorization:** The system should verify whether the authenticated user has the necessary permissions to perform a specific operation.
5. **Input Validation:** To enhance security, the system should perform thorough input validation to prevent potential attacks like SQL injection or cross-site scripting (XSS). This validation process applies to all input, including data provided by users and information received from external systems. By validating the input, the system can detect and mitigate any malicious or unauthorized attempts to manipulate or exploit vulnerabilities, ensuring the integrity and safety of the application and its data.
6. **Error Handling:** The system should not reveal sensitive information in error messages or logs. Information that could help an attacker, such as system details or account information, should be avoided.

2.10.3 Security Requirements

1. **Data Encryption:** All data, both at rest and in transit, must be encrypted using industry-standard methods.
2. **Authentication:** Implement secure user authentication, preferably with two-factor authentication (2FA).
3. **Authorization:** Use role-based access control (RBAC) to ensure users can only access and modify data pertinent to their role.
4. **Input Validation:** All user inputs must be validated to prevent attacks like SQL injection and cross-site scripting (XSS).
5. **Error Handling:** Ensure error messages do not disclose sensitive system or user information.

2.10.4 Software Quality Attributes

1. **Functionality:** The software should provide all the features and functions that it is intended to, as per the requirements.

2. **Reliability:** The software should be dependable, working reliably under defined conditions.
3. **Usability:** User friendliness should be given priority in the program, with a simple, intuitive interface that takes little training to use and navigate.
4. **Efficiency:** The software should make optimal use of system resources, including memory, processor, disk space, and network bandwidth.
5. **Maintainability:** The software should be designed for easy updates and maintenance, incorporating modular design principles, comprehensive documentation, and adherence to coding standards. This approach ensures that future modifications and maintenance tasks can be carried out smoothly, promoting the longevity and scalability of the software.
6. **Portability:** The software should be able to run in different environments, including different operating systems and hardware configurations.
7. **Scalability:** The software should be able to handle the increased load without a significant decrease in performance.
8. **Performance:** The software should respond quickly to user inputs, with minimal latency and high throughput.
9. **Security:** The software should protect data and resist attacks. This includes data encryption, user authentication, and access control.

CHAPTER 3

DESIGN AND METHODOLOGY

3.1 UML Diagrams

The design phase of the “Expense Manager” contains the following Components:

3.1.1 Use Case Diagram

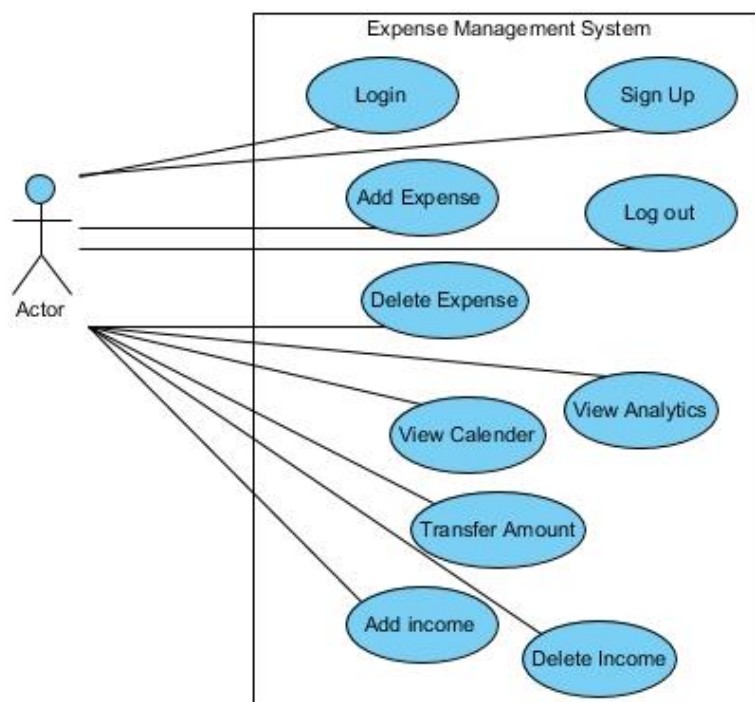


Figure 3.1: Use-Case Diagram

3.1.2 Domain Model

The collection of needs that are shared by all the systems in a product line are represented by domain models. A single product line may represent several domains, or areas of knowledge, while a single domain may cross several product lines. A domain model represents the following requirements:

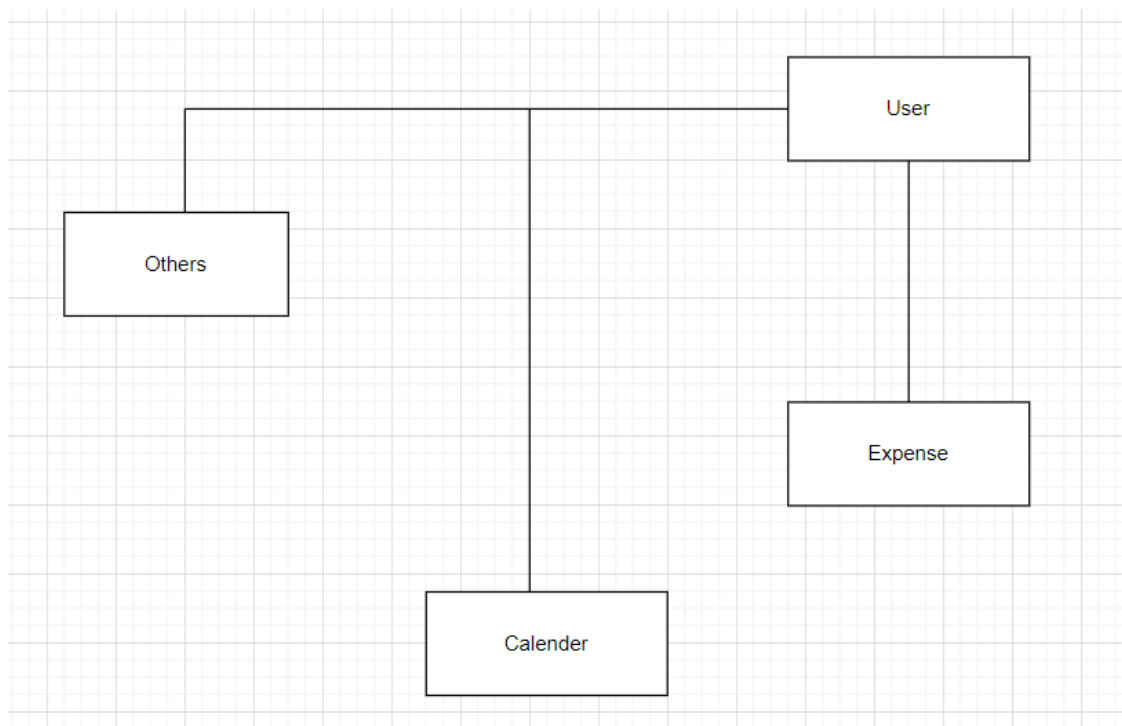


Figure 3.2: Domain Model

3.1.3 Sequence Diagram

A sequence diagram displays various concurrent processes or objects as parallel vertical lines (lifelines), and the messages that are passed between them as horizontal arrows, in the order that they occur. The following are sequence diagrams for the "Expense Manager":

3.1.3.1 User Registration

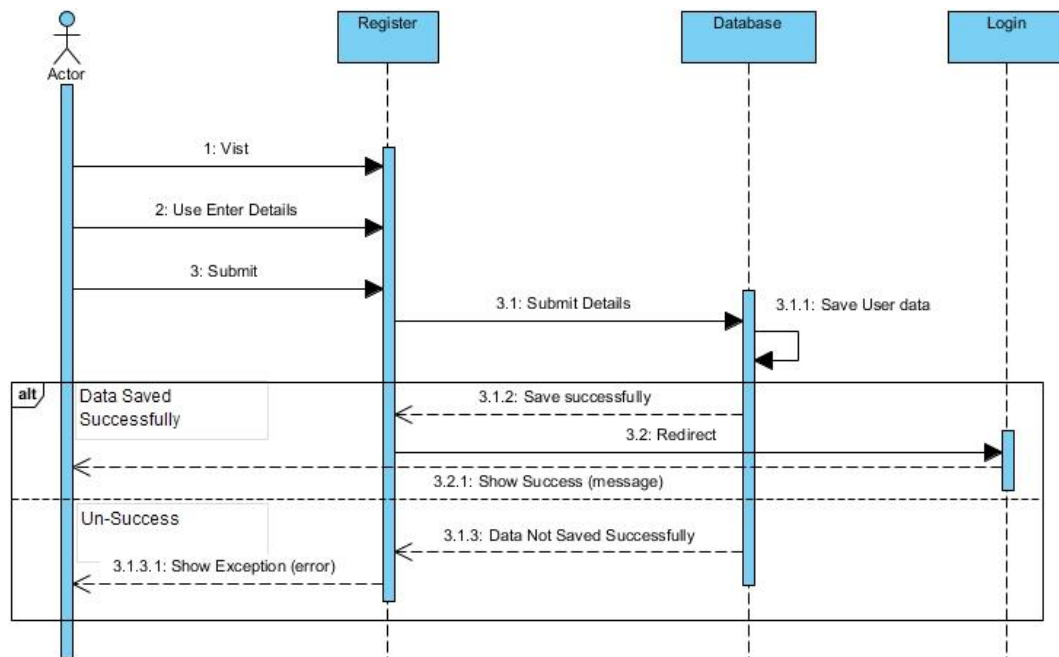


Figure 3.3: User Registration

3.1.3.2 User Login

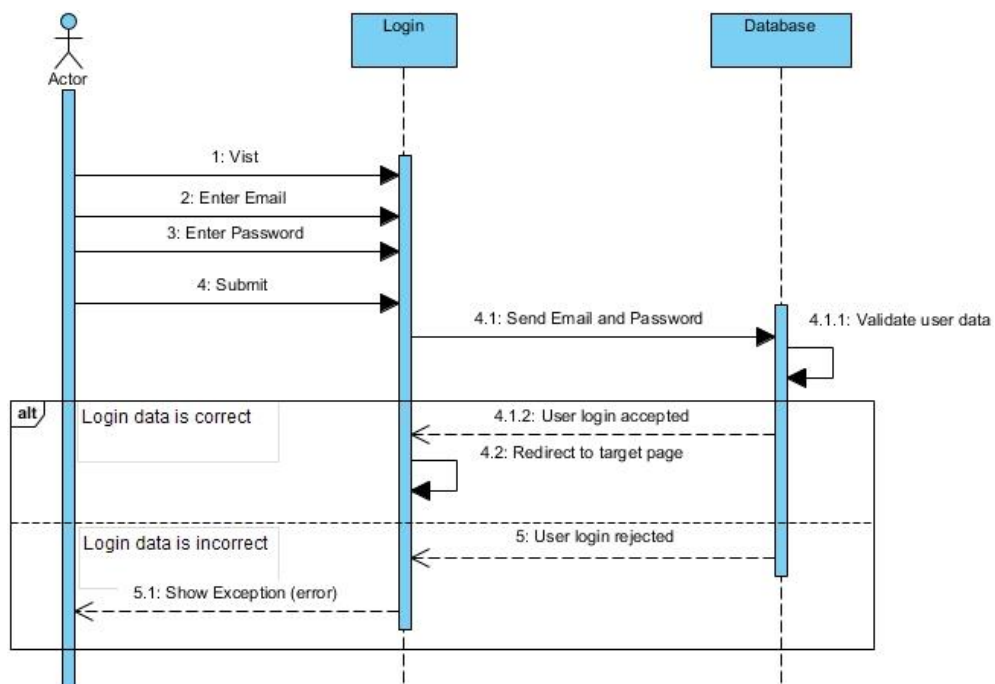


Figure 3.4: User Login

3.1.3.3 Add Income

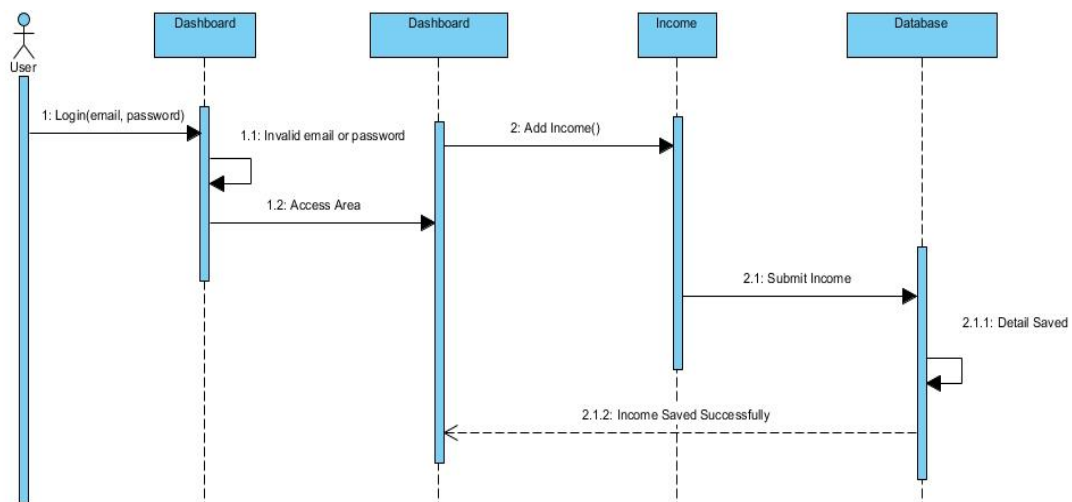


Figure 3.5: Add Income

3.1.3.4 Add Expense

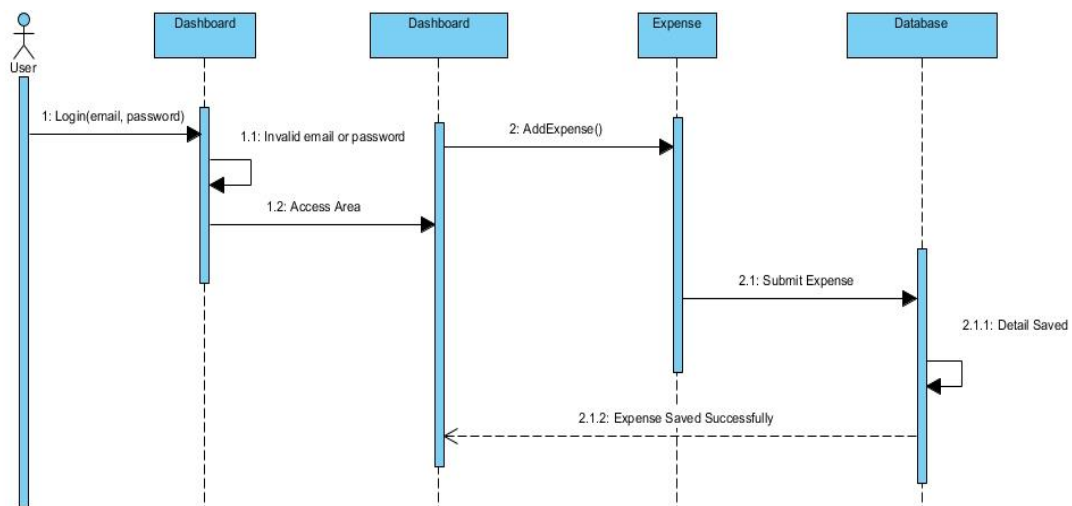


Figure 3.6: User Login

3.1.3.5 View Income

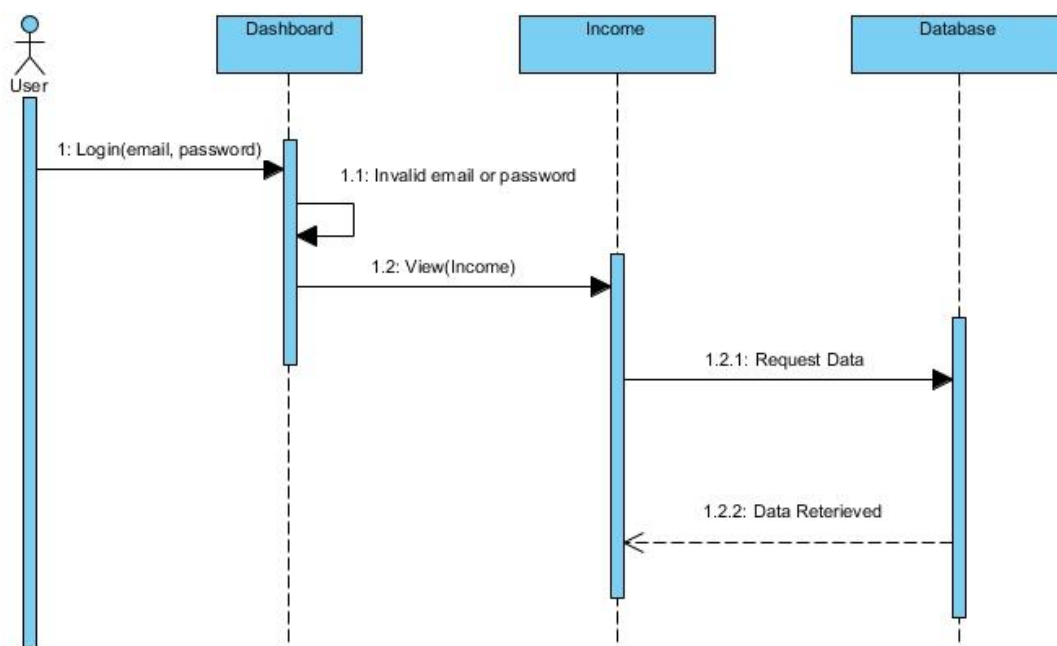


Figure 3.7: View Income

3.1.3.6 View Expense

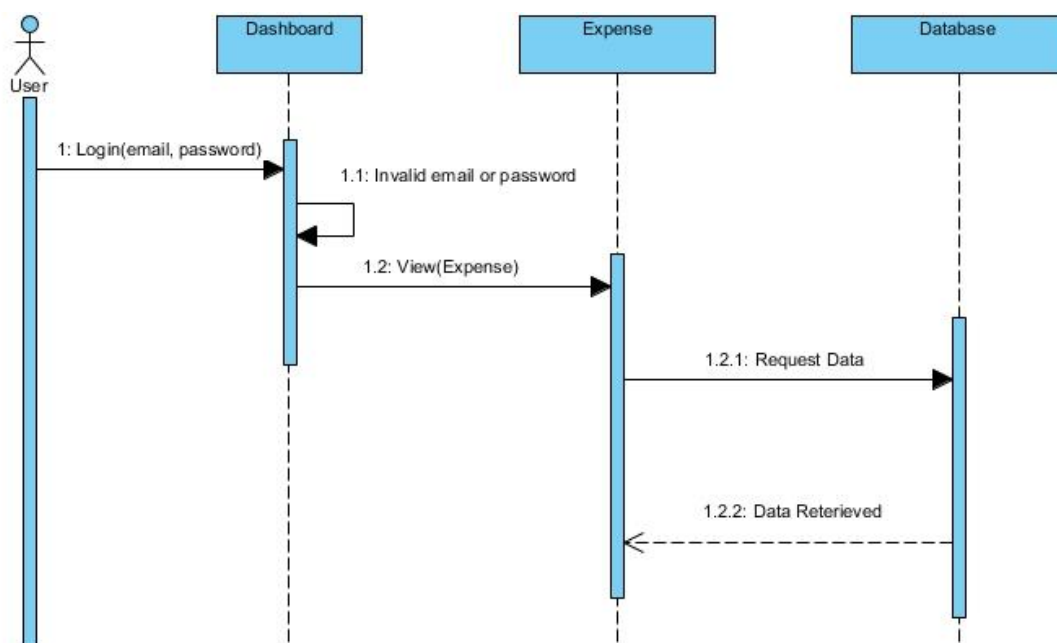


Figure 3.8: View Expense

3.1.4 Collaboration Diagram

3.1.4.1 User Registration

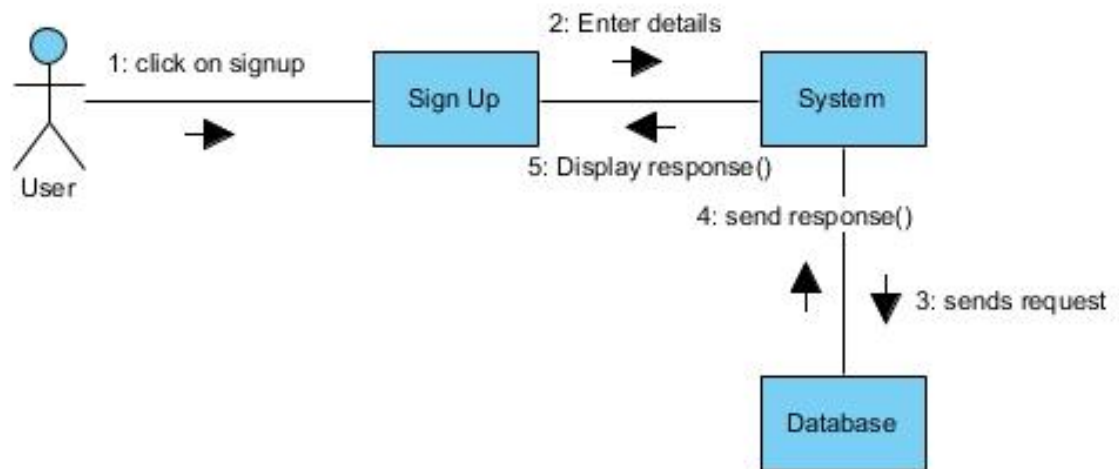


Figure 3.9: User Registration

3.1.4.2 User Login

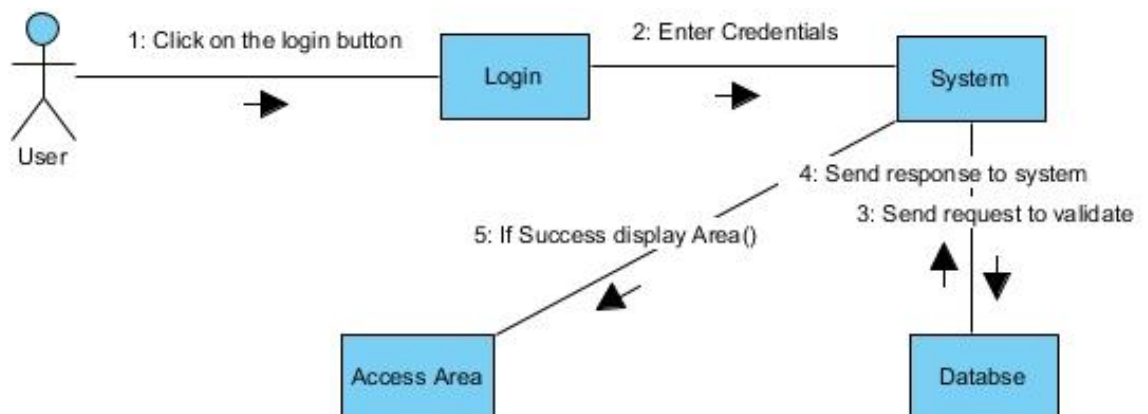


Figure 3.10: User Login

3.1.4.3 Add Income

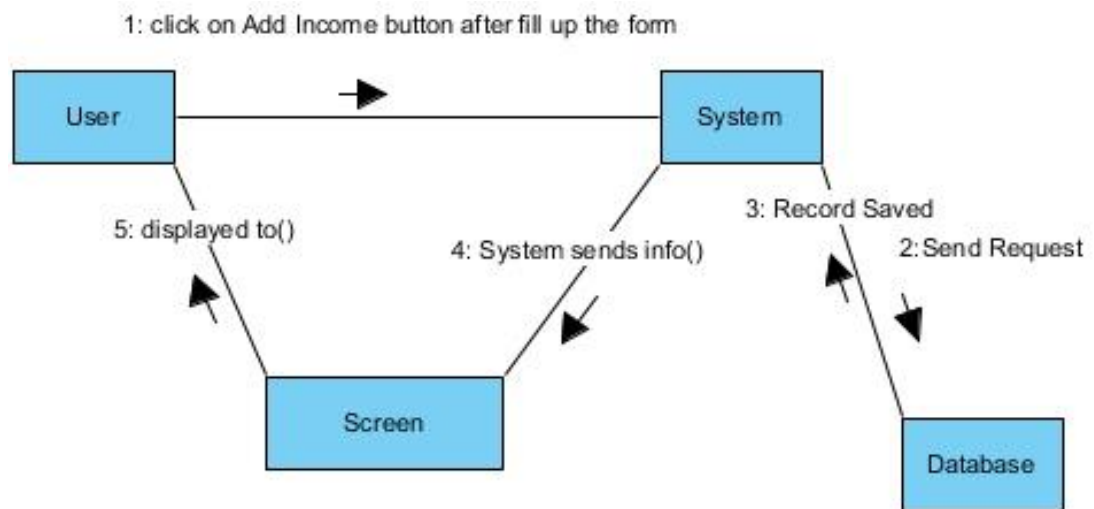


Figure 3.11: Add Income

3.1.4.4 Add Expense

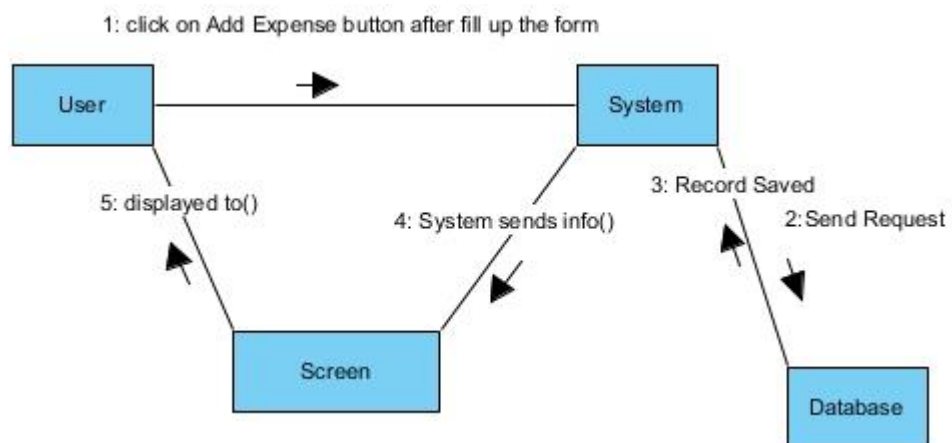


Figure 3.12: Add Expense

3.1.4.5 View Report

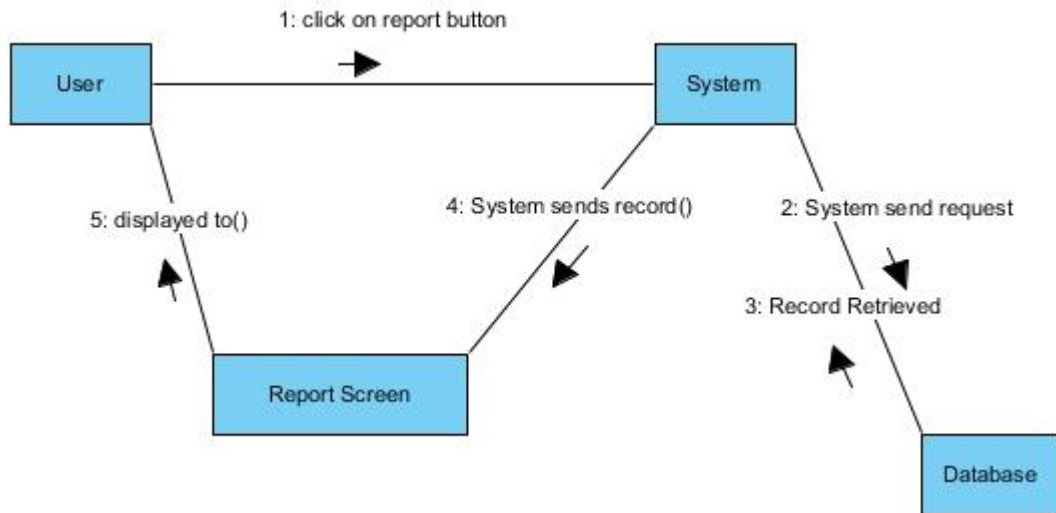


Figure 3.13: View Report

3.1.5 Class Diagram

A class diagram is a static structure diagram that shows the classes, properties, operations (or methods), and interactions between objects in a system to give a general overview of the structure of the system. The class diagram for the "Expense Manager" system is shown below:

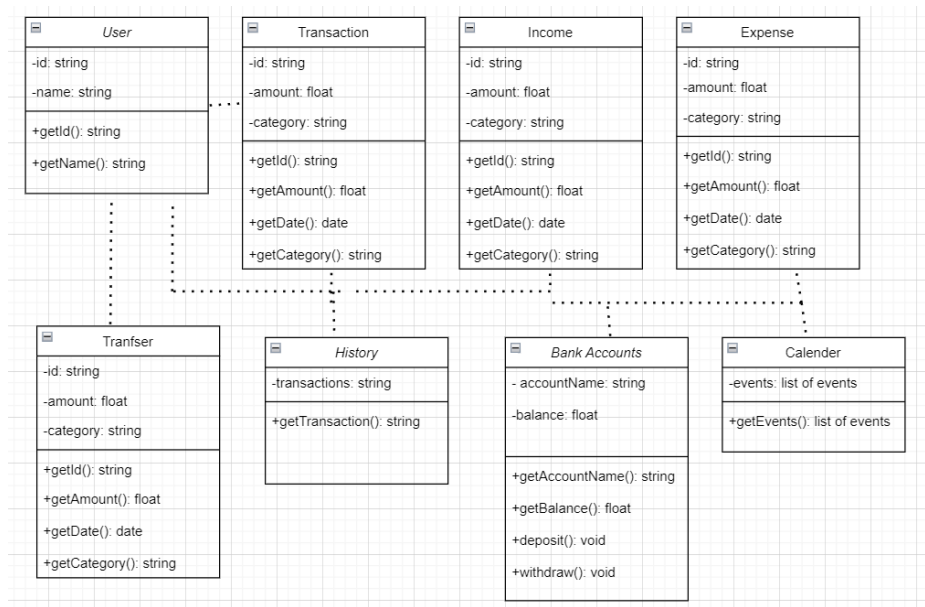


Figure 3.14: Class Diagram

3.1.6 ERD Diagram

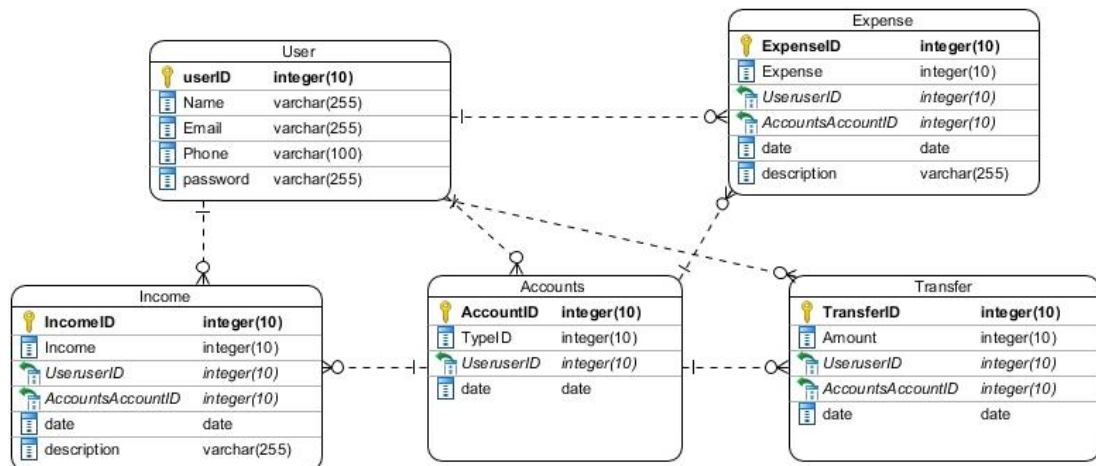


Figure 3.15: ERD Diagram

3.2 Methodology

The methodology used for this project is scrum methodology; the purpose of using scrum is because of its simplicity and iterative development.

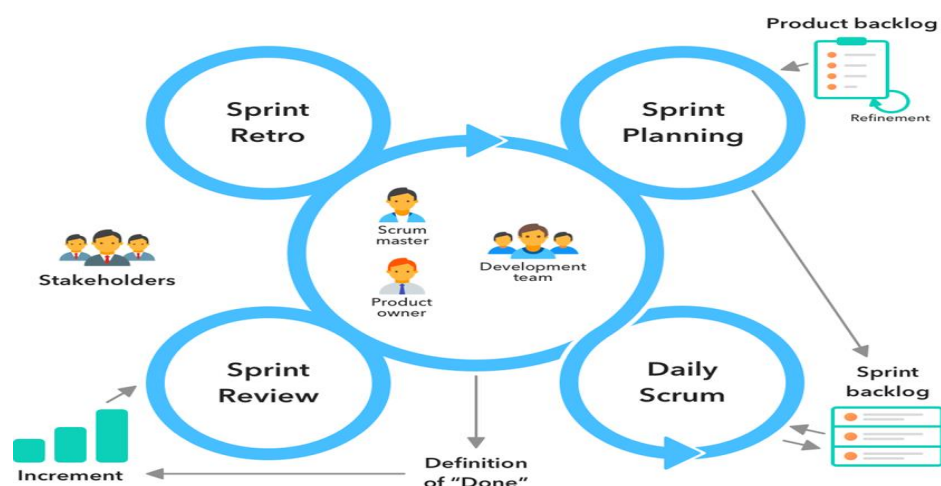


Figure 3.16: Methodology

The Scrum methodology includes further definitions and specifications but pursues the values and principles of agile. Class diagrams and sequence diagrams will

be used in the system analysis, and feedback on the initial design will be acquired from real users of the system.

3.2.1 Build Feature List

In this step, all-important features will be listed.

3.2.2 Sprint Planning

The next step after building the feature list is to produce the development plan which is called sprint planning. This phase also determines sprint duration.

3.2.3 Testing and Sprint demonstration

The team demonstrates the results of their current sprint by creating a review. The supervisor decides on further project changes after examining the demonstration.

3.2.4 Retrospective and Next Spring Planning

In this phase sprint's result will be discussed by the supervisor and determine the ways how to improve the development process. After this next sprint planning will be started.

CHAPTER 4

DATA AND IMPLEMENTATION

4.1 Implementation

During the implementation phase of the project, the designs generated in the design phase are translated into actual software development.

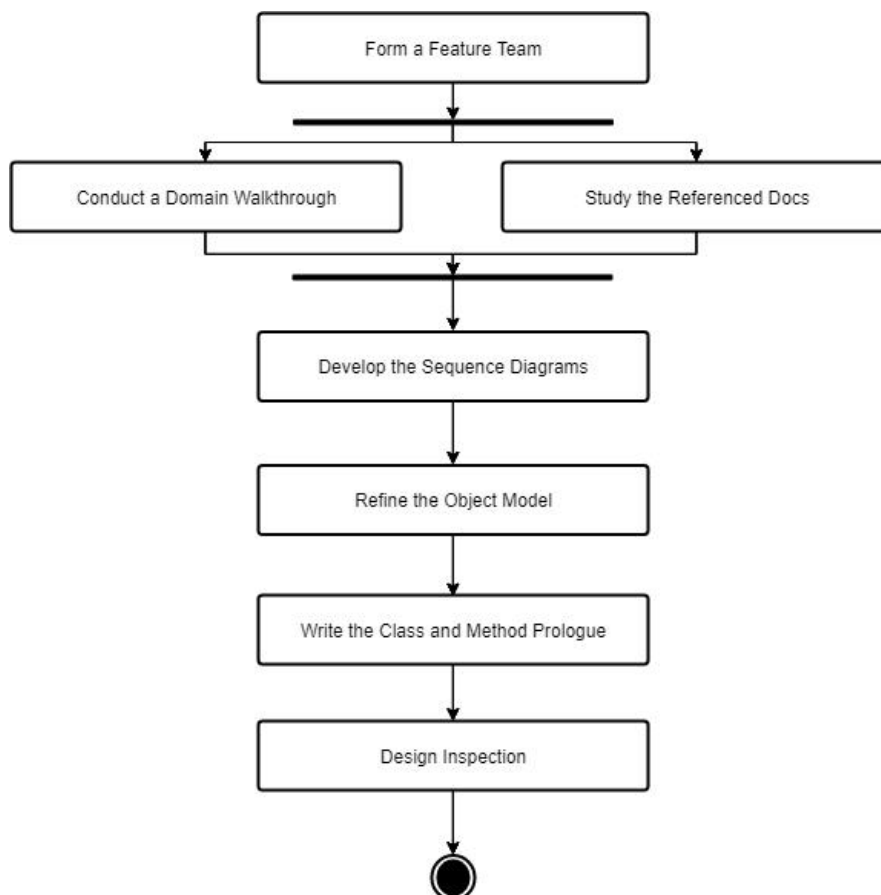


Figure 4.1: Feature Driven Development Diagram

The software development process typically comprises several phases, including planning and design, implementation, testing, documentation, and maintenance. These stages collectively contribute to the successful development, quality assurance, and ongoing support of the software product.

4.1.1 Planning and designing

This is the initial phase of the software development process. The requirements and analyses are acquired from various users. Different users have provided all the information necessary for the development of the Expense Management System.

4.1.2 Implementation

The system structure is converted into an executable program during this stage. React programming language that is utilized to transform the system structure for the expense management system. JavaScript and CSS are two other programming languages that are employed. The creation of the expense management system is made possible using these programming languages. MongoDB is the database management system utilized for this project. A certain programming language will be used to create the program.

4.1.3 Testing

This includes operating the system, assessing potential flaws, and taking appropriate corrective measures. The system is tested using unit testing and end-to-end testing, which examine the many system elements, including the user interface, data storage, and the activities of the various stakeholders.

4.1.4 Documentation

During this phase, all development-related actions must be noted and recorded. The recorded information is vital for any system updates or upgrades that may be necessary, as well as for future use as a guide while performing maintenance.

4.1.5 Deployment and Maintenance

The final presentable system is presented to the Computer Science department during this phase. After then, the product can be modified to suit the user's requirements before being placed into use. Newly found errors can be fixed, and unmet needs can be added.

4.1.6 Technical Aspects of Expense Management System

Since the Expense Management System is an embedded system, it requires both a web server and a database management system to function effectively. The system is divided into two main categories: front-end and back-end.

4.2 Hardware

Hardware contains necessary hardware such as Personal Computer (PC), Android or iOS mobiles.

4.3 Front-end

Front-end involves HTML, CSS, Bootstrap 5.0, JavaScript, and React JS.

4.4 Back-end

Back-end involves Node.js and MongoDB

CHAPTER 5

USER MANUAL

5.1 General Information

An expense manager web application is an online platform that helps individuals or organizations track and manage their expenses. It simplifies the process of expense reporting and reimbursement for businesses, and it can also be used by individuals to keep track of personal expenses.

5.2 Overview

An Expense Manager Web Application serves as a digital solution for managing, tracking, and analysing personal or business expenses. This tool is essential for businesses of all sizes, individuals, or families who aim to maintain financial discipline and transparency.

5.3 Organization of the Manual

The user manual is mentioned below.

5.4 System Configuration

Gardening Perfect operators on mobile devices or laptops. It can run on any mobile device or laptop, active Email addresses, and active internet.

5.5 User Access Level

Only registered users can order from the Expense Manager application.

5.6 Contingencies

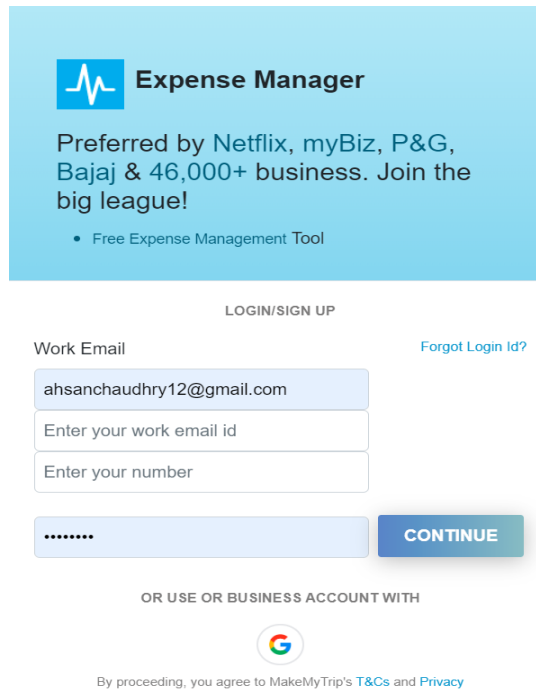
The Expense Manager will stop working if there is no internet connection. Registration procedure for users is required.

5.7 Understanding the Expense Management System

This Section includes all the components of the Expense Manager Web Application. This mainly consists of the following:

5.7.1 Sign Up

User can add name, email, phone number and password to sign up for account to login and utilize the functionalities of the expense manager.



Expense Manager

Preferred by Netflix, myBiz, P&G, Bajaj & 46,000+ business. Join the big league!

- Free Expense Management Tool

LOGIN/SIGN UP

Work Email [Forgot Login Id?](#)

ahsanchaudhry12@gmail.com


Enter your work email id

Enter your number

.....

CONTINUE

OR USE OR BUSINESS ACCOUNT WITH

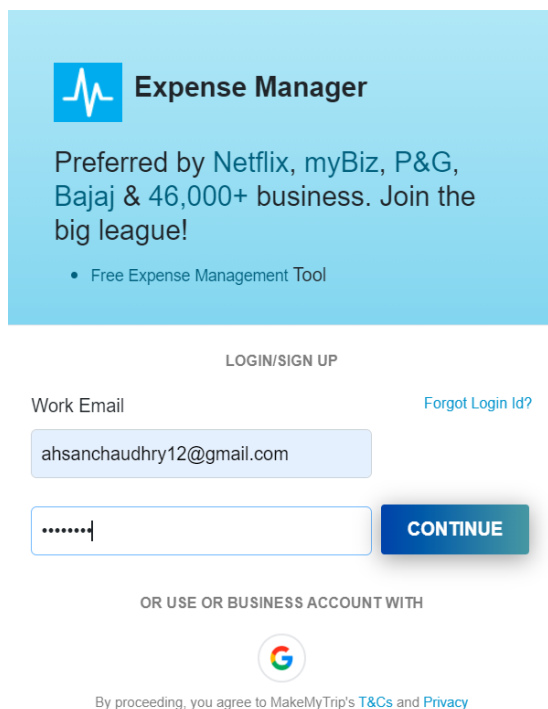


By proceeding, you agree to MakeMyTrip's [T&Cs](#) and [Privacy](#)

Figure 5.1: Sign Up

5.7.2 Login

User can add the email and password to login to the expense manager that the user had registered during the sign-up process.



Expense Manager

Preferred by Netflix, myBiz, P&G, Bajaj & 46,000+ business. Join the big league!

- Free Expense Management Tool

LOGIN/SIGN UP


Work Email [Forgot Login Id?](#)

ahsanchaudhry12@gmail.com

.....

CONTINUE

OR USE OR BUSINESS ACCOUNT WITH

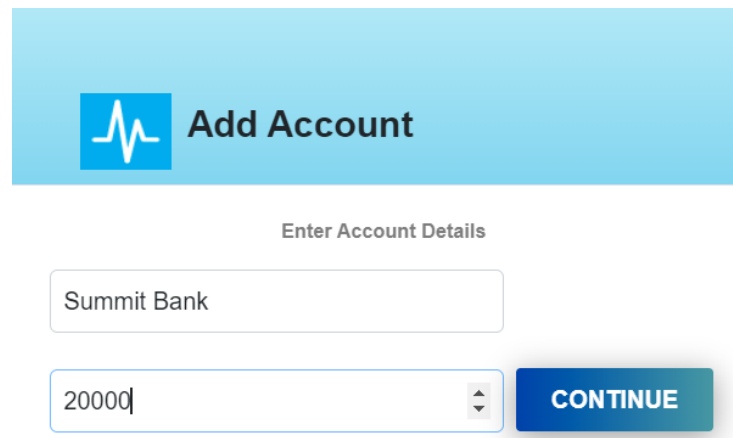


By proceeding, you agree to MakeMyTrip's [T&Cs](#) and [Privacy](#)

Figure 5.2: Login

5.7.3 Add Bank Account

User can add bank details such as name of the bank and the amount that the user wants to add to the account.



Enter Account Details

Summit Bank

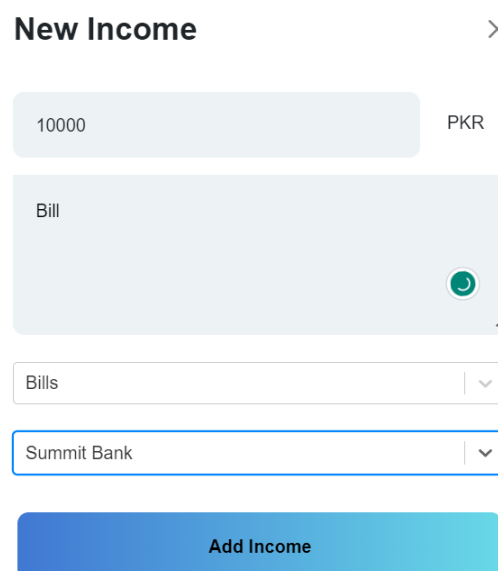
20000

CONTINUE

Figure 5.3: Add Bank Account

5.7.4 Add Income

User can add income that the user wants to add, give the description, add the category and the name of the bank account the user wants to add the income.



New Income

10000 PKR

Bill

Bills

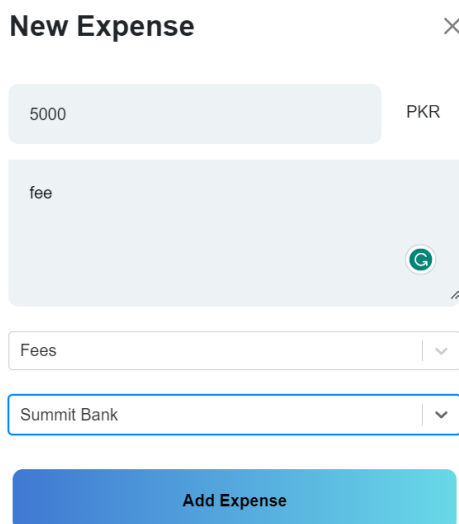
Summit Bank

Add Income

Figure 5.4: Add Income

5.7.5 Add Expense

User can add expense that the user wants to add, give the description, add the category and the name of the bank account the user wants to add the expense. The added amount would be deducted from the bank account.



New Expense X

5000 PKR

fee

Fees

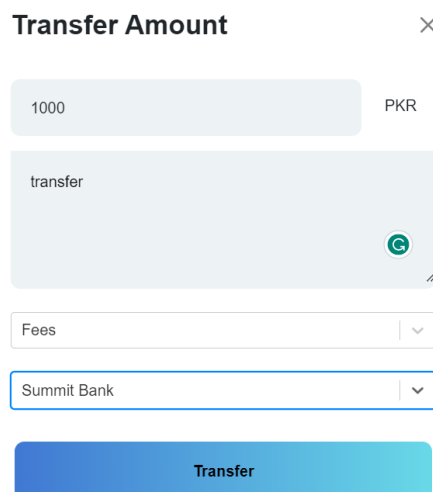
Summit Bank

Add Expense

Figure 5.5: Add Expense

5.7.6 Transfer

User can transfer the amount from one account to another, give the description, and add the category. One amount is deducted from one account and added to another account.



Transfer Amount X

1000 PKR

transfer

Fees

Summit Bank

Transfer

Figure 5.6: Transfer

5.7.7 Transaction

Transaction history is shown on clicking on the transaction tab in the expense manager system. On typing in the search field, the transactions will be filtered according to the words typed.

Description
Search 1 records...

Transaction Type
Search 1 records...

Category
Search 1 records...

Amount
Search 1 records...

Description	Transaction Type	Category	Amount
fee	expense	Fees	5000

Figure 5.7: Transaction

5.7.8 Calendar

On clicking on the calendar tab in the expense manager system, the events will be showed on the calendar with added incomes, expenses, and transfers.

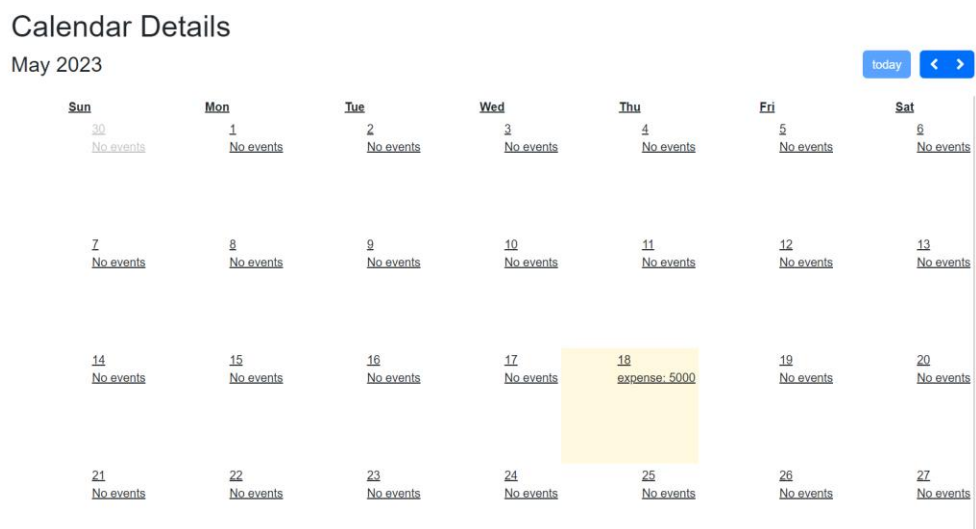


Figure 5.8: Calendar

CHAPTER 6

CONCLUSION AND RECOMMENDATIONS

6.1 Conclusion

The main purpose of this system is to provide users with a platform for easy and efficient management of their expenses, including income and expenditures. There will be no need to make daily expense sheets manually. This way the users can easily maintain their expenses by entering each income and outcome details into the system and they can generate a report of the specified time defined by the user. As this system manages the account so this system is built with the consideration that no error or breach comes during the use of this system.

6.2 Recommendation

We would like to recommend that a lot more work can be done on this project which may include extending the database, and increasing productivity, a mobile application can also be designed for Android or iOS. Besides this below are some recommendations that can be considered for to use of this system without interruption

6.2.1 Strong Internet Connection

Users must have access to a 3G or 4G network connection on their devices to ensure uninterrupted access to the system and prevent any disruptions caused by poor internet connectivity.

REFERENCES

APPENDICES

TURNITIN REPORT

BSCS-FYP-018

ORIGINALITY REPORT

16%	8%	1%	14%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Higher Education Commission Pakistan <small>Student Paper</small>	4%
2	Submitted to Edge Hill University <small>Student Paper</small>	2%
3	www.coursehero.com <small>Internet Source</small>	2%
4	Submitted to SUNY, Empire State College <small>Student Paper</small>	1%
5	Submitted to HELP UNIVERSITY <small>Student Paper</small>	1%
6	Submitted to University of North Texas <small>Student Paper</small>	<1%
7	Submitted to University of Hertfordshire <small>Student Paper</small>	<1%
8	theses.gla.ac.uk <small>Internet Source</small>	<1%
9	Submitted to Kingston University <small>Student Paper</small>	<1%

10	dl.lib.uom.lk Internet Source	<1 %
11	www.javatpoint.com Internet Source	<1 %
12	Submitted to Adventist University of Central Africa Student Paper	<1 %
13	Submitted to Coventry University Student Paper	<1 %
14	Submitted to University of Northampton Student Paper	<1 %
15	Submitted to University of Westminster Student Paper	<1 %
16	www.aanp.org Internet Source	<1 %
17	Submitted to HCUC Student Paper	<1 %
18	Submitted to KDU College Sdn Bhd Student Paper	<1 %
19	Martina Seidl, Marion Scholz, Christian Huemer, Gerti Kappel. "UML @ Classroom", Springer Science and Business Media LLC, 2015 Publication	<1 %

Submitted to Universiti Malaysia Pahang

20	Student Paper	<1 %
21	Submitted to Segi University College Student Paper	<1 %
22	Submitted to Mae Fah Luang University Student Paper	<1 %
23	dspace.daffodilvarsity.edu.bd:8080 Internet Source	<1 %
24	umpir.ump.edu.my Internet Source	<1 %
25	Submitted to Informatics Education Limited Student Paper	<1 %
26	Submitted to Vels University Student Paper	<1 %
27	core.ac.uk Internet Source	<1 %
28	norma.ncirl.ie Internet Source	<1 %
29	www.nccr-north-south.ch Internet Source	<1 %

Exclude quotes On

Exclude matches Off

Exclude bibliography On

