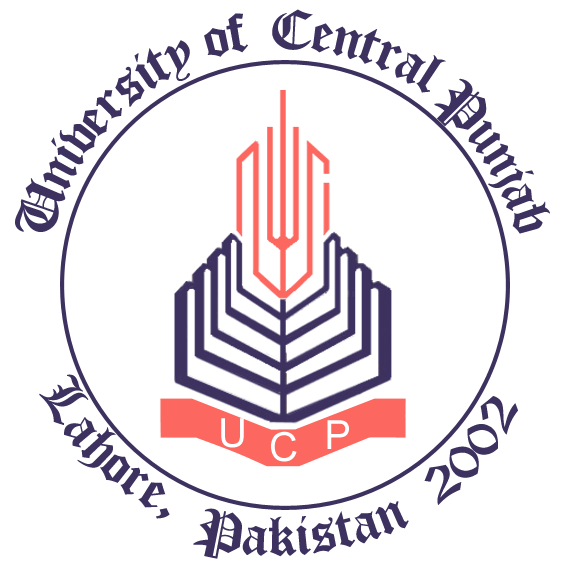
ISD TERM PROJECT

Software Requirements Specification (SRS) for

Expense Tracker



Project Advisor

**Ma’am Misbah Naz**

Presented by:

**Group ID: 0917**

L1F21BSCS0917 Taaha Hussain Khan

**Faculty of Information Technology**

**University of Central Punjab**

Software Requirements Specification

Version 1

Expense Tracker

Advisor: Ma’am Misbah Naz

Group 0917

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Member Name** | **Primary Responsibility** | **% Contribution** | **Roll No** | **Serial #** | **Sign** |
| **Taaha Hussain Khan** |  | **100 %** | **0917** | **L1F21BSCS0917** | **Taaha** |

**Table of ContentsTable of Contents...................................................................................................................................................... iii**

**1. Introduction and Background...................................................................................................... 1**

**1.1 Product (Problem Statement) ...................................................................................................... 1**

**1.2 Background.................................................................................................................................... 1**

**1.3 Scope................................................................................................................................................ 1**

**1.4 Objective(s)/Aim(s)/Target(s)....................................................................................................... 1**

**1.5 Challenges....................................................................................................................................... 1**

**1.6 Learning Outcomes........................................................................................................................ 2**

**1.7 Nature of End Product.................................................................................................................. 2**

**1.8 Completeness Criteria................................................................................................................... 2**

**1.9 Business Goals................................................................................................................................ 2**

**1.10 Related Work/ Literature Survey/ Literature Review............................................................... 2**

**1.11 Document Conventions.................................................................................................................. 3**

**2. Overall Description........................................................................................................................ 3**

**2.1 Product Features............................................................................................................................ 3**

**2.2 User Classes and Characteristics.................................................................................................. 3**

**2.3 Operating Environment................................................................................................................ 3**

**2.4 Design and Implementation Constraints..................................................................................... 4**

**2.5 Assumptions and Dependencies.................................................................................................... 4**

**3. Project Management...................................................................................................................... 4**

**3.1 Work Breakdown Structure (WBS)............................................................................................. 4**

**3.2 Develop an Activity Graph............................................................................................................ 5**

**3.3 Find the Critical Path.................................................................................................................... 6**

**3.4 Create a Gantt chart...................................................................................................................... 7**

**~~~~~~~~~~ Phase II**

**4. Functional Requirements.............................................................................................................. 7**

**4.1 Name of Use-Case 1………........................................................................................................... 8**

**4.2 Name of Use-Case 2 (and so on) ................................................................................................... 9**

**Requirements Analysis and Modeling..................................................................................................…10**

**5. Nonfunctional Requirements\*.....................................................................................................13**

**5.1 Performance Requirements.....................................................................................................................................……13**

**5.2 Safety Requirements.....................................................................................................................13**

**5.3SecurityRequirements..........................................................................................................................14**

**5.4 Additional Software Quality Attributes......................................................................................14**

**6. Other Requirements.....................................................................................................................14**

**7. Designing.................................................................................................................................. ….16**

**7.1 Complete class diagram................................................................................................................16**

**7.2 Complete Data Flow Diagram(DFD)..........................................................................................17**

**7.3 Complete ER Diagram.................................................................................................................17**

**7.4 Physical design of your database.................................................................................................18**

**7.5 Information on use of design patterns while designing the modules........................................19**

**7.6 Make a Sequence and a Collaboration diagram of following...................................................20**

**7.6.1 Scenario 1.......................................................................................................................................20**

**7.6.2 Scenario 2......................................................................................................................................20**

**8. Estimation......................................................................................................................................21**

**8.1 Cost Benefit Estimation................................................................................................................21**

**8.1.1 Return of Investment (ROI) {1st year}**

**8.3 COCOMO Estimation.....................................................................................................22**

**References................................................................................................................................... 22**

**Appendix A: Glossary.................................................................................................................22**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Taaha Hussain Khan | 04-12-2023 | Initial Draft | 1 |

# Introduction and Background

## Product (Problem Statement)

## This project aims to develop a comprehensive expense tracker application that enables users to effortlessly manage their personal finances. The application will provide a user-friendly interface for recording, categorizing, and analyzing expenses, allowing users to gain insights into their spending habits and make informed financial decisions.

## Background

## In today's fast-paced world, managing personal finances can be a daunting task. With a multitude of expenses to track, it's easy to lose sight of spending patterns and make unplanned expenditures. Traditional methods of expense tracking, such as spreadsheets or notebooks, often prove cumbersome and inefficient.

## Scope

## The scope of this project encompasses the development of a mobile application that caters to individuals seeking to manage their personal expenses. The application will provide functionalities for recording expenses, creating expense categories, generating reports, and setting spending limits.

## Objective(s)/Aim(s)/Target(s)

## The primary objectives of this project are as follows:

* To develop a user-friendly expense tracker application that seamlessly integrates into users' daily lives.
* To provide comprehensive expense tracking capabilities, enabling users to record, categorize, and analyze their expenses effectively.
* To generate insightful reports and visualizations, allowing users to understand their spending patterns and identify areas for potential savings.
* To empower users to make informed financial decisions based on their expense data.

## Challenges

The key challenges associated with this project include:

* Ensuring data accuracy and security: Implementing robust data validation and encryption mechanisms to safeguard user information.
* Designing an intuitive and user-friendly interface: Developing an interface that is easy to navigate and accessible to users of varying technical expertise.
* Integrating with financial institutions: Enabling seamless synchronization with users' bank accounts for automated expense tracking.
* Providing personalized insights: Utilizing data analytics techniques to generate tailored recommendations based on user's spending patterns.

## Learning Outcomes

Upon completion of this project, the expected learning outcomes include:

* Proficiency in mobile application development using a modern framework.
* In-depth understanding of data management and security principles.
* Expertise in user interface design and user experience optimization.
* Familiarity with data analytics techniques for personalized insights.

## Nature of End Product

The end product of this project will be a cross-platform mobile application compatible with iOS and Android devices. The application will be distributed through respective app stores and accessible to users worldwide.

## Completeness Criteria

The project will be considered complete when the following criteria are met:

* The application successfully records, categorizes, and analyzes expenses accurately.
* The application generates insightful reports and visualizations for expense analysis.
* The application provides a secure and user-friendly interface for managing finances.
* The application integrates seamlessly with users' bank accounts for automated expense tracking.

## Business Goals

The development of this expense tracker application aligns with the following business goals:

* To expand the company's product portfolio into the personal finance management domain.
* To attract new customers and increase brand awareness.
* To generate revenue through in-app purchases and premium features.
* To establish a positive reputation for providing innovative and user-centric financial solutions.

## Related Work/ Literature Survey/ Literature Review

An extensive literature review has been conducted to identify existing expense tracker applications and analyze their strengths, weaknesses, and features. This review has provided valuable insights for shaping the development of this project.

## Document Conventions

The following document conventions have been adopted throughout this SRS:

* Italicized nouns represent external systems or interfaces.
* Single quotes are used to denote specific functionalities or features.
* Bold text is used to highlight important concepts or key points.

# Overall Description

## Product Features

The Expense Tracker application will offer a comprehensive set of features to empower users to effectively manage their personal finances:

* Effortless Expense Recording: Users can seamlessly record their expenses, including the amount, date, category, and additional notes.
* Organized Expense Categorization: Expenses can be categorized into predefined or user-created categories, facilitating structured tracking of spending patterns.
* Actionable Expense Analysis: The application will generate insightful reports and visualizations, providing users with a clear understanding of their spending habits and areas for potential savings.
* Empowered Budget Management: Users can set spending limits for specific categories or overall expenses, enabling them to stay within their financial goals.
* Widespread Accessibility: The application will be available on both iOS and Android devices, ensuring compatibility with a diverse range of users.

## User Classes and Characteristics

The primary user classes identified for the Expense Tracker application are:

**Individual Users:** Individuals seeking to manage their personal finances and gain insights into their spending patterns.

**Budget-Conscious Individuals:** Individuals actively seeking to optimize their spending habits and achieve financial goals.

**Young Professionals:** Young professionals seeking a user-friendly tool to manage their finances and make informed financial decisions.

## Operating Environment

The Expense Tracker application will operate on the following platforms:

**Mobile Operating Systems**: iOS 14.0 or later, Android 5.0 or later

**Hardware Requirements**: Minimum 1GB RAM, 8GB storage space

**Network Connectivity**: Active internet connection required for synchronization and data backup

## Design and Implementation Constraints

The development of the Expense Tracker application will adhere to the following constraints:

**User-Centric Design**: The application should prioritize ease of use and accessibility for users of varying technical expertise.

**Data Security**: Robust data security measures must be implemented to protect user privacy and financial information.

**Cross-Platform Compatibility:** The application must function seamlessly across iOS and Android platforms.

**Performance Optimization:** The application should maintain optimal performance and responsiveness on a range of mobile devices.

## Assumptions and Dependencies

The development of the Expense Tracker application assumes the following:

**Availability of Third-Party Libraries**: Access to relevant libraries for data encryption, data visualization, and cross-platform development.

**Seamless Integration with Financial Institutions:** Ability to integrate with various bank and financial institutions for automated expense tracking.

**Consistency in Data Storage:** Consistent data storage mechanisms across iOS and Android platforms.

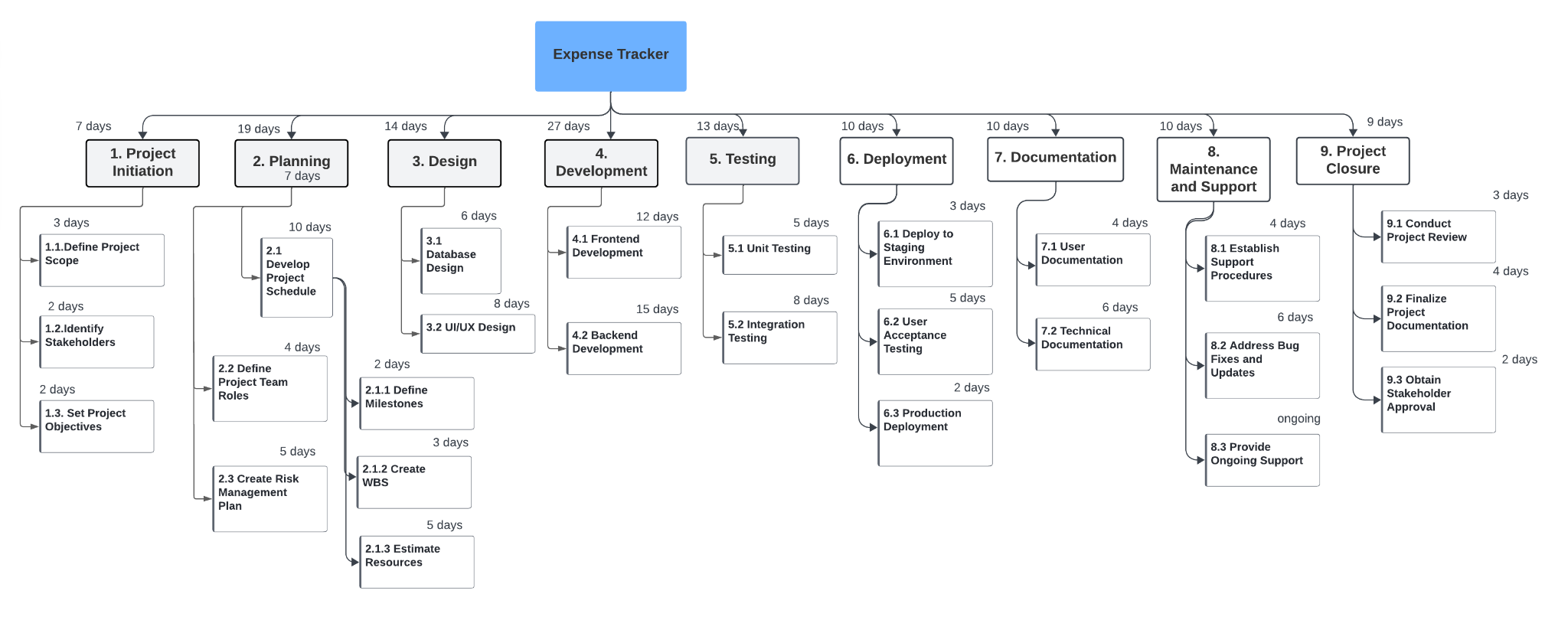
**Stable Internet Connectivity**: Reliable internet connection for data synchronization and backup.

# Project Management.

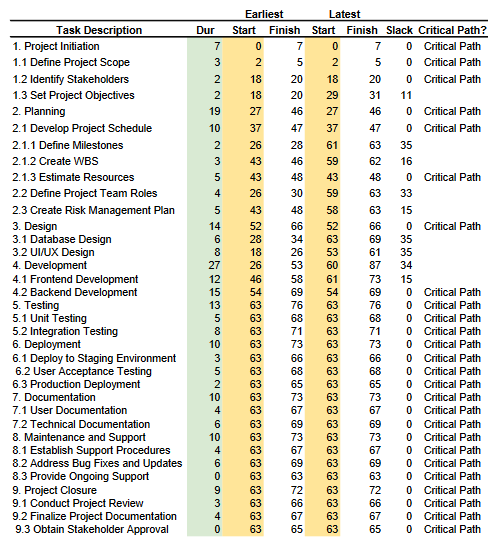
## Work Breakdown Structure (WBS).

|  |  |
| --- | --- |
| **ID** | **Activity** |
| **1.** | **Project Initiation** |
|  | Define Project Scope |
|  | Identify Stakeholders |
| 1.3 | Set Project Objectives |
| **2.** | **Planning** |
| 2.1 | Develop Project Schedule |
| 2.1.1 | Define Milestones |
| 2.1.2 | Create WBS |
| 2.1.3 | Estimate Resources |
| 2.2 | Define Project Team Roles |
| 2.3 | Create Risk Management Plan |
| **3.** | **Design** |
| 3.1 | Database Design |
| 3.2 | UI/UX Design |
| **4.** | **Development** |
| 4.1 | Frontend Development |
| 4.2 | Backend Development |
| **5.** | **Testing** |
| 5.1 | Unit Testing |
| 5.2 | Integration Testing |
| **6.** | **Deployment** |
| 6.1 | Deploy to Staging Environment |
| 6.2 | User Acceptance Testing |
| 6.3 | Production Deployment |
| **7.** | **Documentation** |
| 7.1 | User Documentation |
| 7.2 | Technical Documentation |
| **8.** | **Maintenance and Support** |
| 8.1 | Establish Support Procedures |
| 8.2 | Address Bug Fixes and Updates |
| 8.3 | Provide Ongoing Support |
| **9.** | **Project Closure** |
| 9.1 | Conduct Project Review |
| 9.2 | Finalize Project Documentation |
| 9.3 | Obtain Stakeholder Approval |

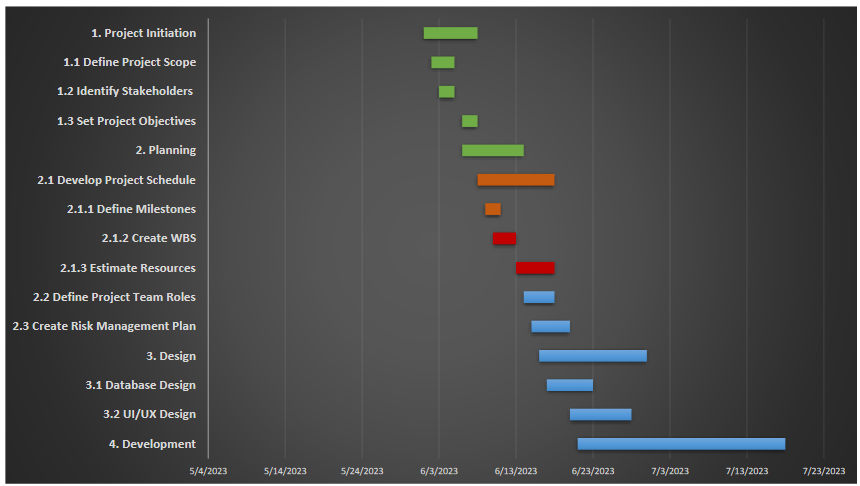
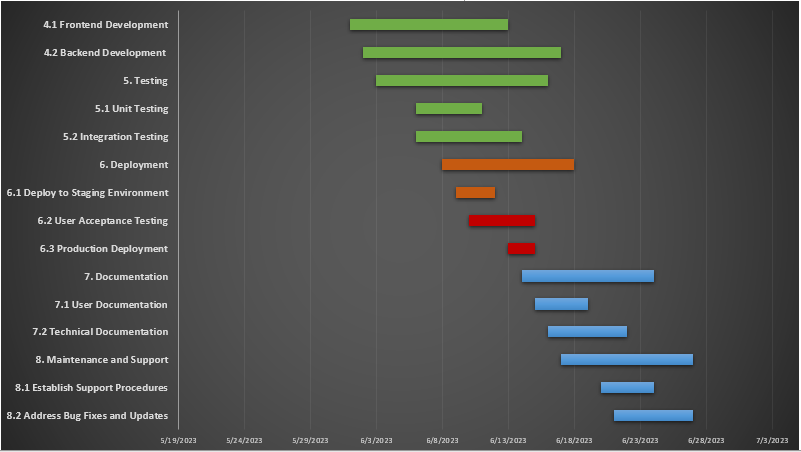
## Develop an Activity Graph.



## Find the Critical Path.



## Create a Gantt chart.

****

# ~~~~~~~~~~ Phase II ~~~~~~~~~~

# Functional Requirements

*<All functional requirements are expressed as use-cases. Fill out the following template for each use-case. Don’t really say “Use-Case 1.” State the use-case name in just a few words e.g. “Withdraw Cash from ATM”. A use-case may have multiple alternate courses of action.>*

*<Provide a Use Case Diagram before describing the use cases.>*

## Case 1: Record Expense

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-1 | |
| **Purpose** | | To record an expense effortlessly | |
| **Priority** | | High | |
| **Pre-conditions** | | * The user must be authenticated and logged into the application. * The application interface should be accessible. | |
| **Post-conditions** | | The expense is successfully recorded and saved in the system. | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | User selects “Record Expense” option | | Application displays expense recording interface |
| **2** | User enters expense details (amount, date, category, notes) | | System validates input and saves expense data |
| **3** | User confirms the expense entry | | Application notifies successful recording. |
| **Alternate Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | User corrects the invalid or missing fields | | System validates input; detects invalid or missing fields |
| **2** | User submits the corrected expense details | | System validates and saves the corrected expense data |
| **3** | User confirms the expense entry | | Application notifies successful recording |

**Table 1: UC-1**

## Use-Case 2: View Expense History

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-2 | |
| **Purpose** | | Allow users to view their past recorded expenses | |
| **Priority** | | High | |
| **Pre-conditions** | | * The user must be authenticated and logged into the application. * The application interface should be accessible. | |
| **Post-conditions** | | The users can review their expense history. | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | User navigates to “Expense History”. | | Application fetches and displays expense history |
| **2** | User scrolls or searches for specific expense | | System filters and displays relevant expense entries. |
| **Alternate Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | User has no recorded expenses | | Application displays a message indicating empty history |
| **2** | User searches for non-existent entry | | Application informs user that no matching record was found |

**Table 2: UC-2**

## Use-Case 3: Set spending Limit

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-3 | |
| **Purpose** | | Allows users to set spending limits for specific expense categories | |
| **Priority** | | Medium | |
| **Pre-conditions** | | * The user must be authenticated and logged into the application. * The application interface should be accessible. | |
| **Post-conditions** | | The expense is successfully recorded and saved in the system. | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | User navigates to “Budget Management” section | | Application displays budget settings interface |
| **2** | User selects a category and sets a spending limit. | | System validates input and updates the spending limit |
| **Alternate Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | User sets a limit higher than budget | | Application warns and suggests a lower limit. |
| **2** | User enters non-numeric value | | Application prompts user to enter a valid numeric value |

**Table 3: UC-3**

## Use-Case 4: Generate Expense Report

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-4 | |
| **Purpose** | | Allows users to set generate reports summarizing their expenses over a specified period. | |
| **Priority** | | Medium | |
| **Pre-conditions** | | * The user must be authenticated and logged into the application. * The application interface should be accessible. | |
| **Post-conditions** | | The user can view or download the generated expense report. | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | User selects to “Generate Report” section | | Application prompts user to specify report parameters. |
| **2** | User selects time period and report format. | | System generates and displays the report. |
| **Alternate Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | User requests report for empty period | | Application informs no data available. |
| **2** | User selects unsupported format | | Application prompts user to select a supported format |

**Table 4: UC-4**

## Requirements Analysis and Modeling

**1. Requirements Gathering:**

* **Stakeholder Interviews:** Engage with stakeholders such as users, product owners, and domain experts to understand their needs and expectations.
* **Surveys and Questionnaires:** Collect feedback from a broader audience to capture diverse perspectives.
* **Review Existing Documentation:** Analyze any existing documents, such as project proposals or user feedback reports, to gather insights.

**2. Requirements Documentation:**

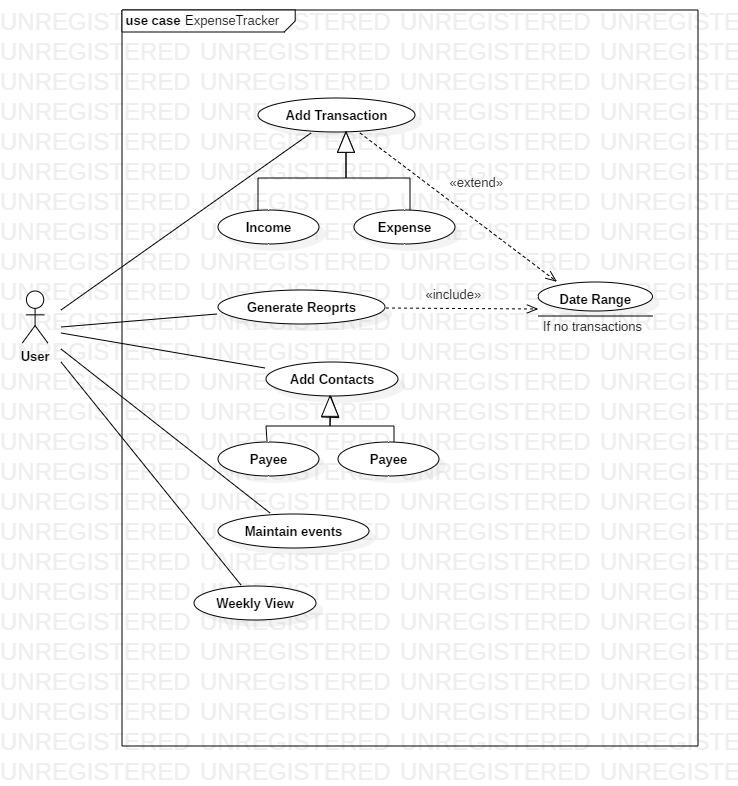
* **Functional Requirements:** Document the specific functionalities the system must provide, such as recording expenses, generating reports, and setting spending limits.
* **Non-Functional Requirements:** Specify constraints or quality attributes of the system, including performance, security, usability, and scalability.
* **Use Cases:** Define the interactions between users and the system, detailing scenarios of how users will accomplish tasks.
* **User Stories:** Describe system features from the perspective of an end user to ensure a user-centric approach.
* **Requirements Traceability Matrix:** Establish links between requirements, use cases, and test cases to ensure comprehensive coverage.

**3. Requirements Analysis:**

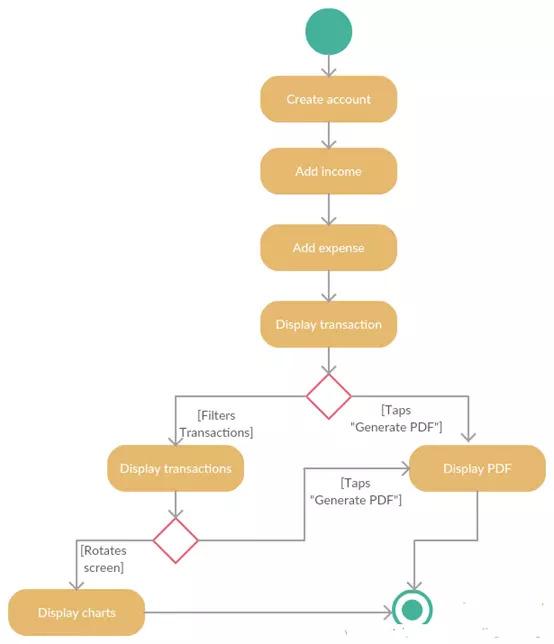
* **Prioritization:** Evaluate and prioritize requirements based on their importance and impact on the overall system functionality.
* **Feasibility Study**: Assess the technical, economic, and operational feasibility of implementing each requirement.
* **Requirement Validation:** Verify the accuracy, completeness, and consistency of requirements through reviews, walkthroughs, and prototyping.
* **Risk Analysis:** Identify potential risks associated with each requirement and develop mitigation strategies.

**4. Requirements Modeling**:

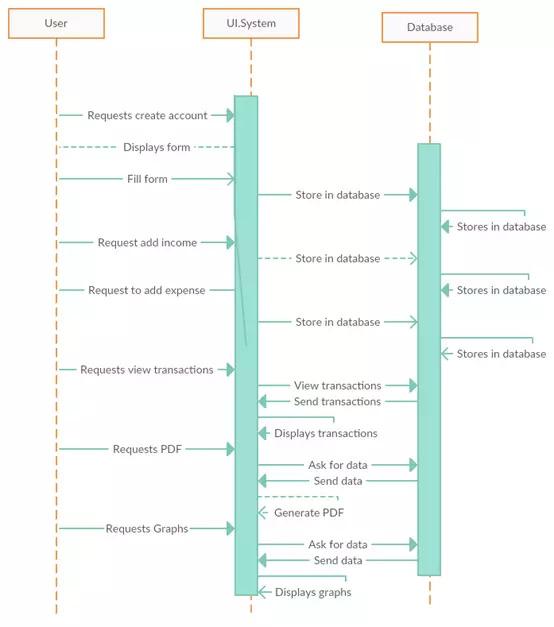
* **Use Case Diagrams:** Illustrate the interactions between users and the system, highlighting key functionalities and actors involved.



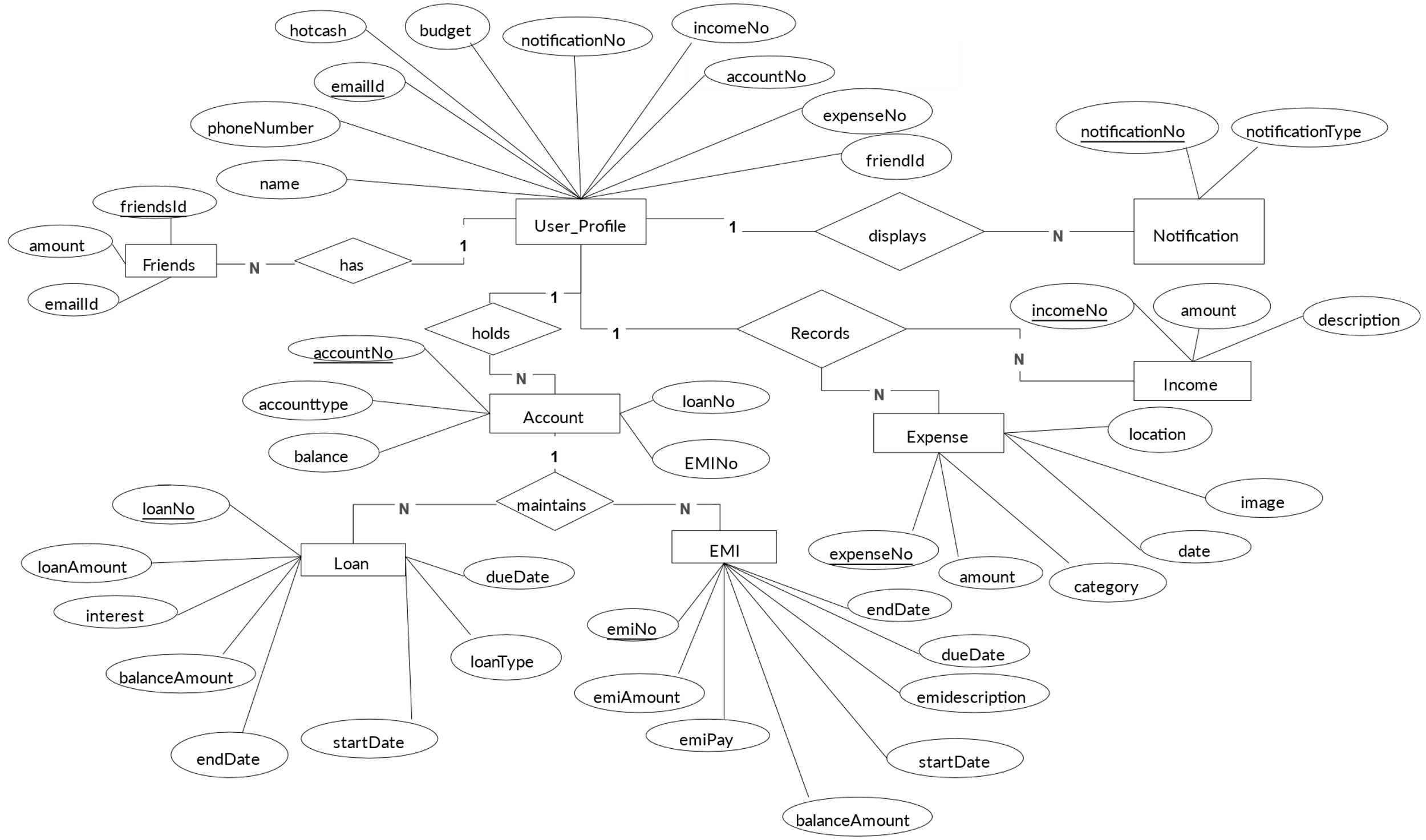
* **Activity Diagrams:** Model the flow of activities or processes within the system to understand how users navigate through various tasks.



* **Sequence Diagrams:** Visualize the sequence of interactions between system components or actors to depict the dynamic behavior of the system.



* **Entity-Relationship Diagrams**: Define the relationships between different data entities within



# Nonfunctional Requirements

## Performance Requirements

* The expense tracker system should respond to user inputs within 2 seconds under normal load conditions.
* The system should be capable of handling at least 100 concurrent users without significant degradation in performance.
* Monthly reports generation should not take more than 5 seconds per user.
* The system should maintain 99.9% uptime, allowing for scheduled maintenance windows.

## Safety Requirements

* User data, including financial information, must be encrypted both during transmission and storage to prevent unauthorized access.
* The system must have backup and recovery mechanisms in place to ensure data integrity in case of system failures.
* Compliance with relevant data protection regulations such as GDPR and HIPAA must be ensured.

## Security Requirements

* Users must authenticate themselves using a strong password or biometric authentication before accessing the expense tracker system.
* Access controls should be implemented to restrict users' access to only the data and features necessary for their roles.
* The system must undergo regular security audits and vulnerability assessments to identify and mitigate potential threats.

## Additional Software Quality Attributes

* **Usability:** The user interface should be intuitive and easy to navigate, with a minimal learning curve for new users.
* **Reliability:** The system should accurately record and calculate expenses without errors.
* Maintainability: The codebase should be well-documented and modular to facilitate future updates and enhancements.
* **Portability:** The expense tracker should be accessible from multiple devices and platforms, including web browsers and mobile apps.
* **Testability:** Automated tests should be implemented to verify the functionality of the system and ensure regression-free releases.

# Other Requirements

**6.1 Database Requirements:**

* The system shall utilize a relational database management system (RDBMS) for storing user data, expense records, and system configurations.
* Database backups shall be performed nightly and retained for at least 30 days to ensure data availability and integrity.

**6.2 External Interface Requirements:**

* The expense tracker system shall integrate with popular payment platforms such as PayPal, Stripe, and credit card providers to facilitate expense tracking and categorization.
* APIs shall be provided to allow integration with third-party accounting software for seamless data transfer and reconciliation.

**6.3 Internationalization Requirements:**

* The user interface shall support multiple languages, allowing users to select their preferred language for interacting with the system.
* Date and currency formats shall be customizable based on the user's locale to accommodate international users.

**6.4 Legal Requirements:**

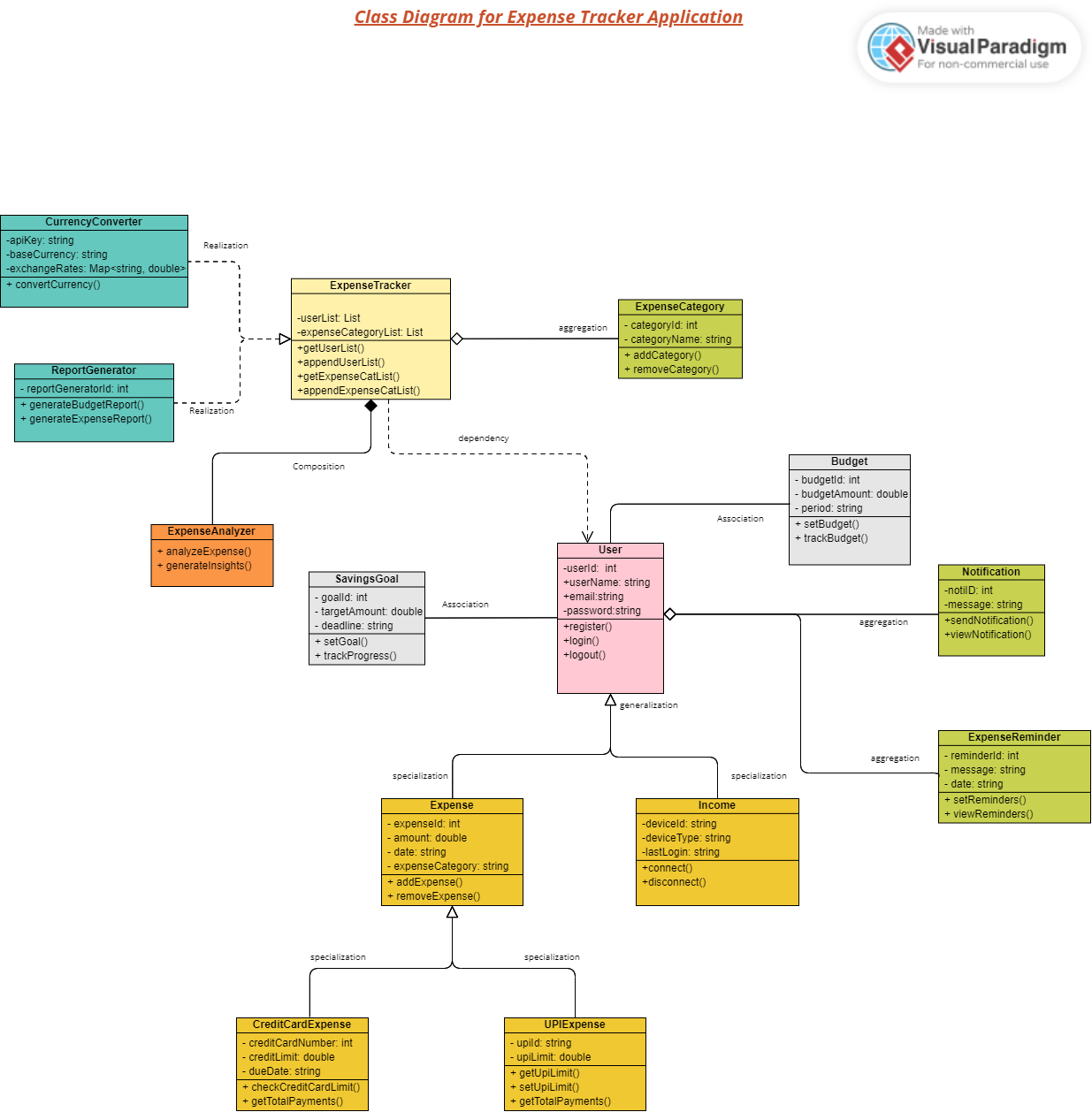
* The system shall comply with all relevant financial regulations and laws in the jurisdictions where it operates, including but not limited to tax reporting requirements.
* Terms of service and privacy policy documents shall be provided to users, outlining their rights and responsibilities when using the expense tracker system.

**6.5 Reuse Objectives:**

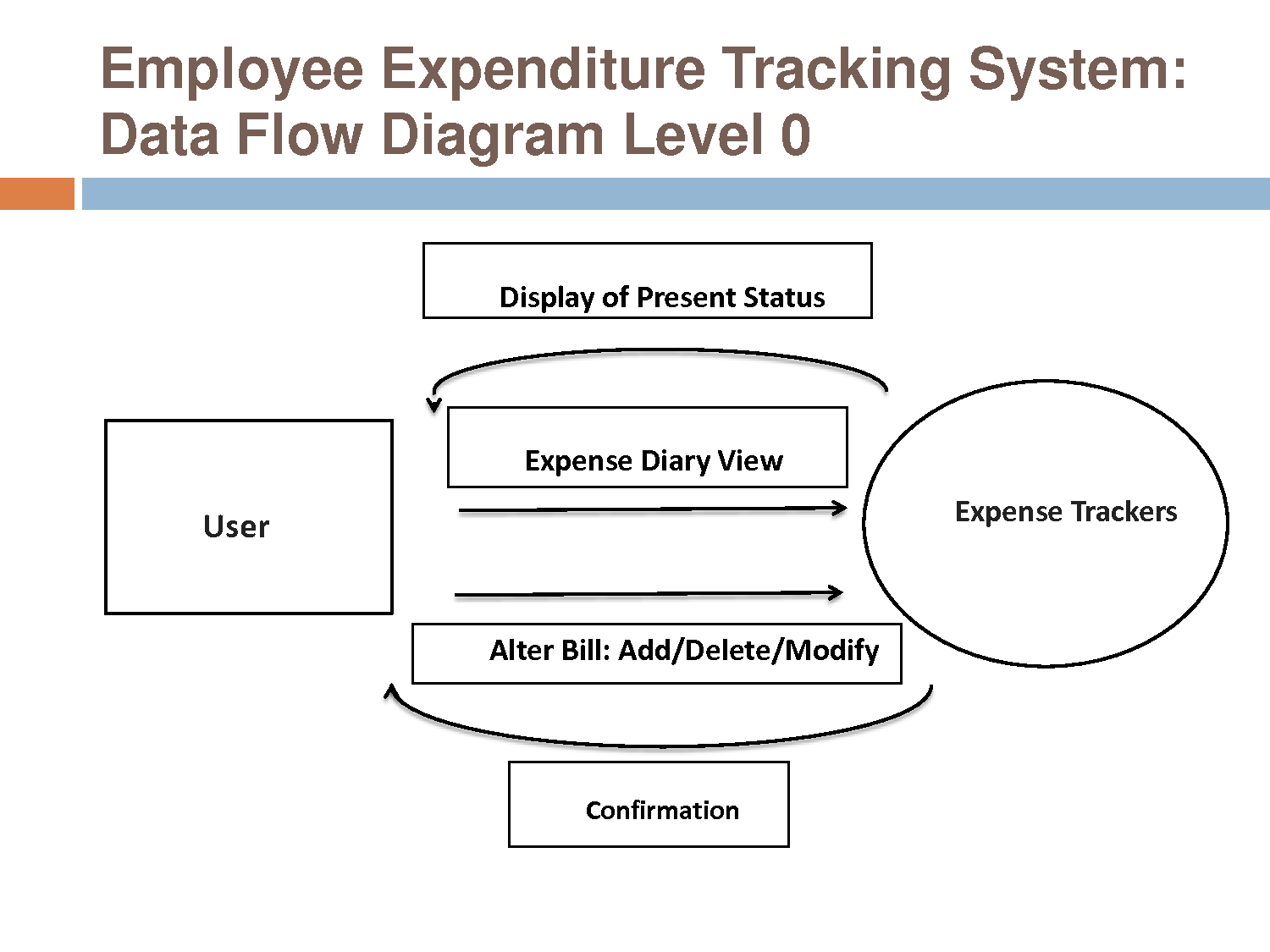
* The codebase shall be structured and documented to facilitate code reuse within the project and across future projects.
* Common modules and components shall be designed with a focus on reusability to minimize duplication of effort and promote maintainability.

1. **Designing**

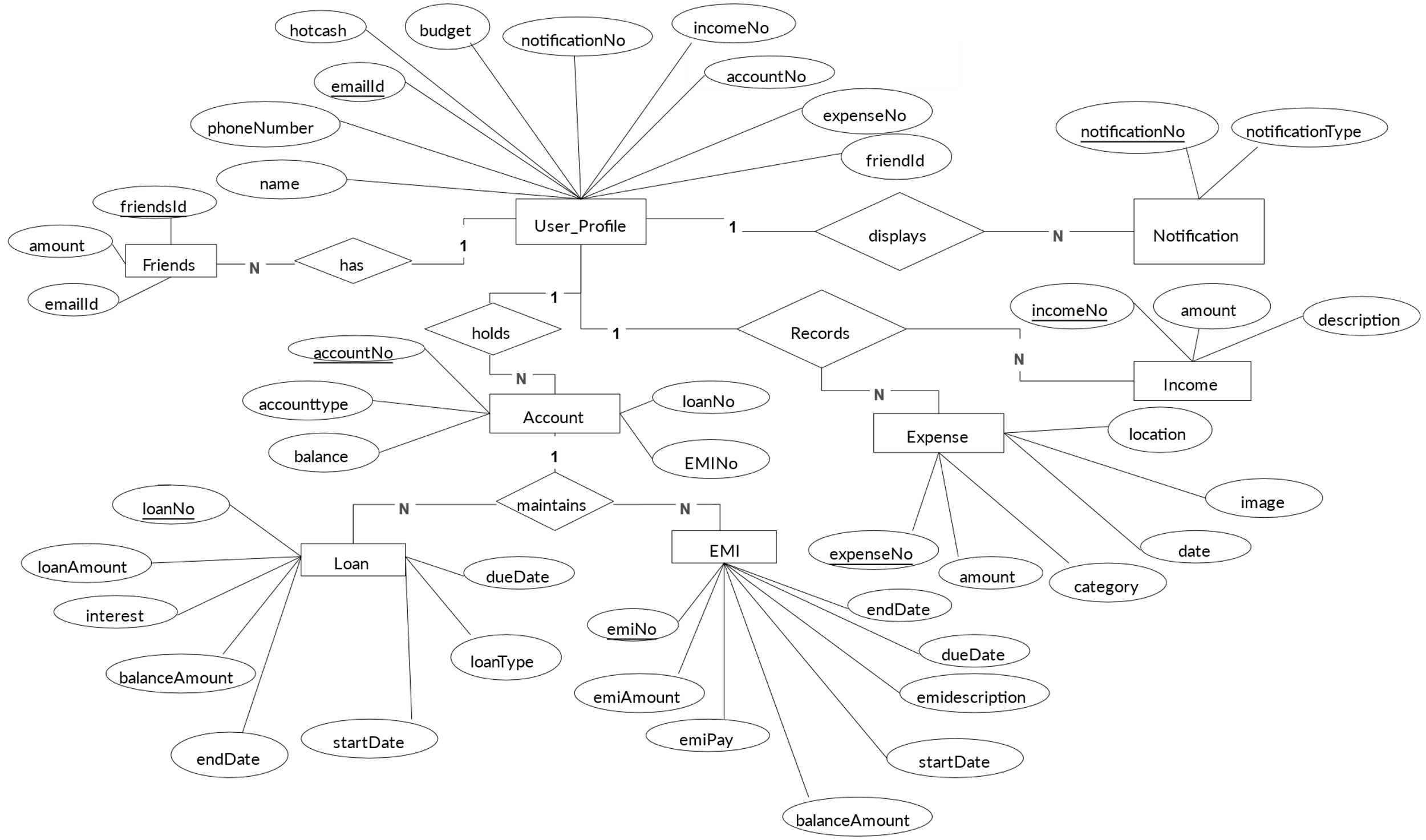
## Complete class diagram



## Complete Data Flow Diagram (DFD)

**

## Complete ER Diagram



## Physical design of your database

1. **Users**

* UserID (Primary Key, int)
* Username (varchar)
* Password (varchar)
* Email (varchar)
* FullName (varchar)
* JoinDate (datetime)

1. **Expenses**

* ExpenseID (Primary Key, int)
* UserID (Foreign Key referencing Users.UserID, int)
* CategoryID (Foreign Key referencing Categories.CategoryID, int)
* Amount (decimal)
* Description (varchar)
* Date (date)

1. **Categories**

* CategoryID (Primary Key, int)
* CategoryName (varchar)

1. **Payment**

* PaymentMethodID (Primary Key, int)
* UserID (Foreign Key referencing Users.UserID, int)
* MethodName (varchar)
* CardType (varchar)
* CardNumber (varchar)
* ExpiryDate (date)

1. **Reports**

* ReportID (Primary Key, int)
* UserID (Foreign Key referencing Users.UserID, int)
* ReportName (varchar)
* StartDate (date)
* EndDate (date)
* TotalAmount (decimal)

## Information on use of design patterns while designing the modules

In the design of the expense tracker system, several design patterns have been employed to enhance the modularity, flexibility, and maintainability of the software modules. The following design patterns have been utilized:

**Model-View-Controller (MVC) Pattern:**

* The MVC pattern separates the application into three interconnected components: Model, View, and Controller. This pattern has been adopted to facilitate the separation of concerns and improve the overall structure of the system.
* **Relevant classes:**
  + **Model:** Expense, User, Category, PaymentMethod
  + **View**: ExpenseListView, ExpenseDetailView, ReportView
  + **Controller:** ExpenseController, UserController, ReportController

**Factory Method Pattern:**

* The Factory Method pattern provides an interface for creating objects but allows subclasses to alter the type of objects that will be created. This pattern has been employed to decouple object creation from the client code and enhance extensibility.
* **Relevant classes:**
  + ExpenseFactory

**Singleton Pattern:**

* The Singleton pattern ensures that a class has only one instance and provides a global point of access to that instance. This pattern has been utilized to ensure that certain resources, such as the database connection, are instantiated only once throughout the application.
* Relevant classes:
  + DatabaseConnection

**Observer Pattern:**

* The Observer pattern defines a one-to-many dependency between objects so that when one object changes state, all its dependents are notified and updated automatically. This pattern has been applied to facilitate communication between components and enable real-time updates.
* Relevant classes:
  + ExpenseSubject, ExpenseObserver

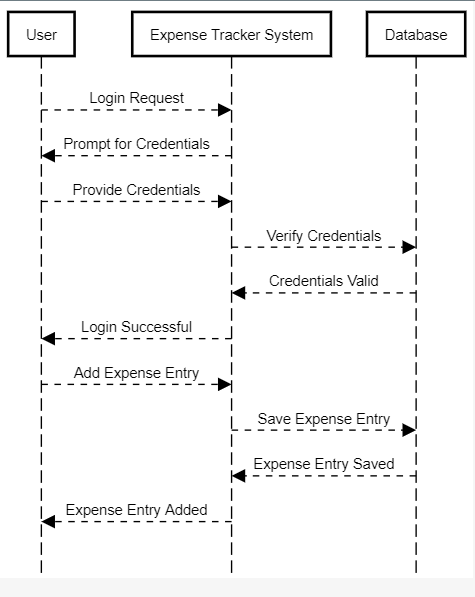
**Strategy Pattern:**

* The Strategy pattern defines a family of algorithms, encapsulates each one, and makes them interchangeable. This pattern has been utilized to enable the dynamic selection of algorithms at runtime, thereby enhancing flexibility.
* Relevant classes:
  + ExpenseSortingStrategy, DateSortingStrategy, AmountSortingStrategy

## Make a Sequence and a Collaboration diagram of following.

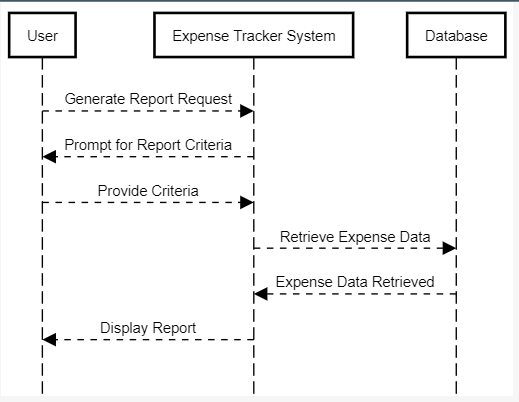
### Scenario 1

In this scenario, let's consider a user logging into the expense tracker system and adding a new expense entry.



### Scenario 2

In this scenario, let's consider a user generating a monthly expense report.



# Estimation

## Cost Benefit Estimation

**Assuming:**

Cost of development and maintenance for the first year: $100,000

Expected benefits from increased productivity and cost savings: $150,000

### Return of Investment (ROI) {1st year}

ROI = 150,000 / 100,000 = 1.5

### %gain on ROI {1st year}

*% gain on ROI = ((150,000 – 100,000) / 100,000) \* 100 = 50%*

### Payback Period in years

Payback period = 100,000 / 150,000 = 0.67 years

## FP based Estimation

Let's assume the following:

* External Inputs (EI): 30
* External Outputs (EO): 20
* External Inquiries (EQ): 15
* Internal Logical Files (ILF): 10
* External Interface Files (EIF): 5

Total Function Point (FP) can be calculated using the following formula:

FP = (EI \* 4) + (EO \* 5) + (EQ \* 4) + (ILF \* 7) + (EIF \* 5)

Plugging the values:

FP = (30 \* 4) + (20 \* 5) + (15 \* 4) + (10 \* 7) + (5 \* 5)

FP = 120 + 100 + 60 + 70 + 25

FP = 375

## COCOMO Estimation

Effort = a \* (FP)b

E = 2.4 \* (375)1.05

E 2.4 \* 423.16

E = 1015.6 person-month

# References

1. **Software Engineering: Theory and Practice**

**Authors:** Shari Pfleeger and Joanne Atlee

**Edition:** 4th Edition

**Chapters:** Selected chapters

**Publication Year**: Not specified

1. **Software Engineering: A Practitioner’s Approach**

**Author:** Roger S. Pressman

**Edition:** 7th Edition

**Chapters:** Selected chapters

**Publication Year**: Not specified