SOFTWARE REQUIREMENTS SPECIFICATION

**For**

**Expense Tracker**

**Prepared by:-**

Dhana Sekar V

Dhayananth B

Kavin Kumar S

# Introduction

## Purpose

## The Expense Tracker project is designed to establish an efficient system for tracking and managing expenses, catering to both individuals and organizations. It aims to provide a user-friendly environment for recording and monitoring expenditures, streamlining the process through computer systems. The primary objective is to enhance financial transparency and decision-making by offering accurate, accessible, and easily navigable tools for expense tracking. The project will generate various reports to provide valuable insights and analysis. Utilizing visual representations like Entity-Relationship (ER) diagrams and Unified Modeling Language (UML) diagrams, it aims to deliver a robust solution for effective and user-centric expense management.

## Document Conventions

* + - Entire document should be justified.
    - Convention for Main title

Font face: Times New Roman Font style: Bold

Font Size: 14

* + - Convention for Sub title

Font face: Times New Roman Font style: Bold

Font Size: 12

* + - Convention for body

Font face: Times New Roman Font Size: 12

## Scope of Development Project

## Expense Tracker System aims to modernize expense management by transitioning from manual tracking to a Java-based web application. Tailored for individuals and businesses, it provides a user-friendly interface for tracking, categorizing, and analyzing expenses. The system is adaptable for diverse organizations, offering scalability and modularity for evolving needs.

## With Java's advantages like high performance, cross-platform compatibility, and extensive libraries, it ensures cost-effectiveness and a robust development process. The project's flexibility allows easy customization, fostering reusability and adaptability to technological advancements, making it an ideal solution for efficient financial tracking in various contexts.

## Definitions, Acronyms and Abbreviations

JAVA -> platform independence SQL-> Structured query Language ER-> Entity Relationship

UML -> Unified Modeling Language

IDE-> Integrated Development Environment SRS-> Software Requirement Specification

## References

* + - Books

 Software Requirements and Specifications: A Lexicon of Practice, Principles and Prejudices (ACM Press) by Michael Jackson

Software Requirements (Microsoft) Second EditionBy Karl E. Wiegers

Software Engineering: A Practitioner’s Approach Fifth Edition By Roger S. Pressman

* + - Websites

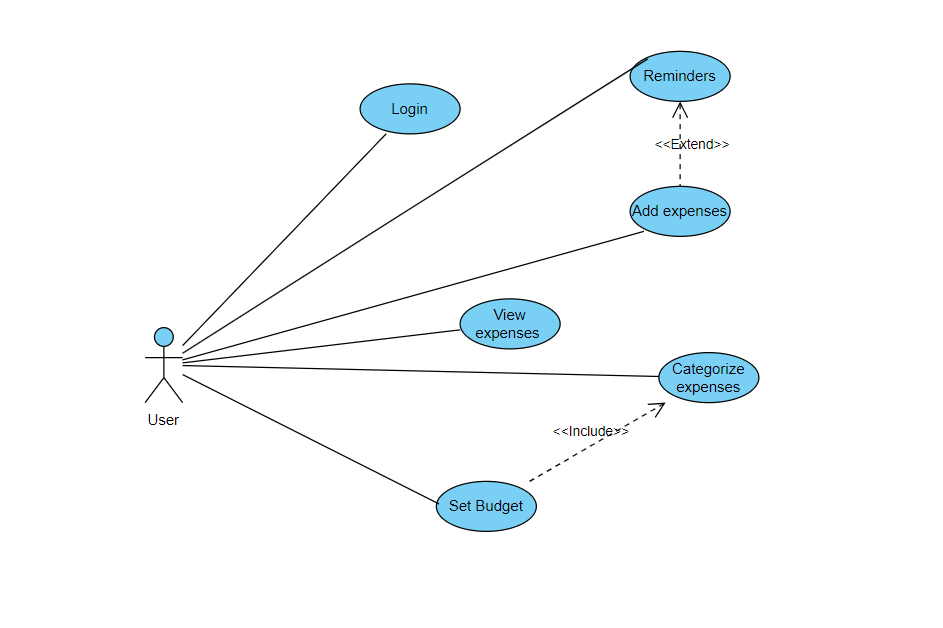
[**http://www.slideshare.net/**](http://www.slideshare.net/)

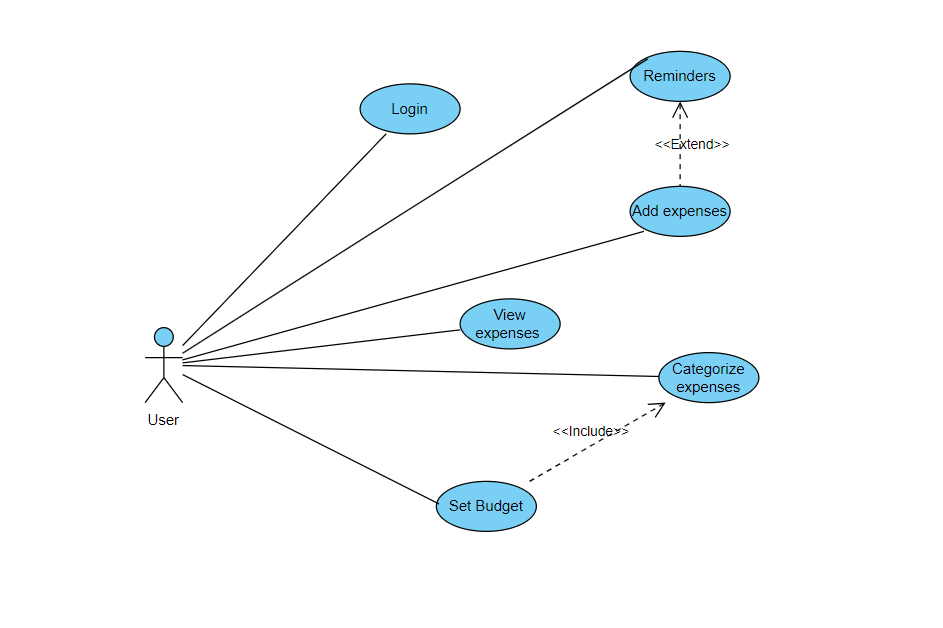
[**http://ebookily.net/doc/srs-library-management-system**](http://ebookily.net/doc/srs-library-management-system)

# Overall Descriptions

## Product Perspective

Use Case Diagram of Expense Tracker

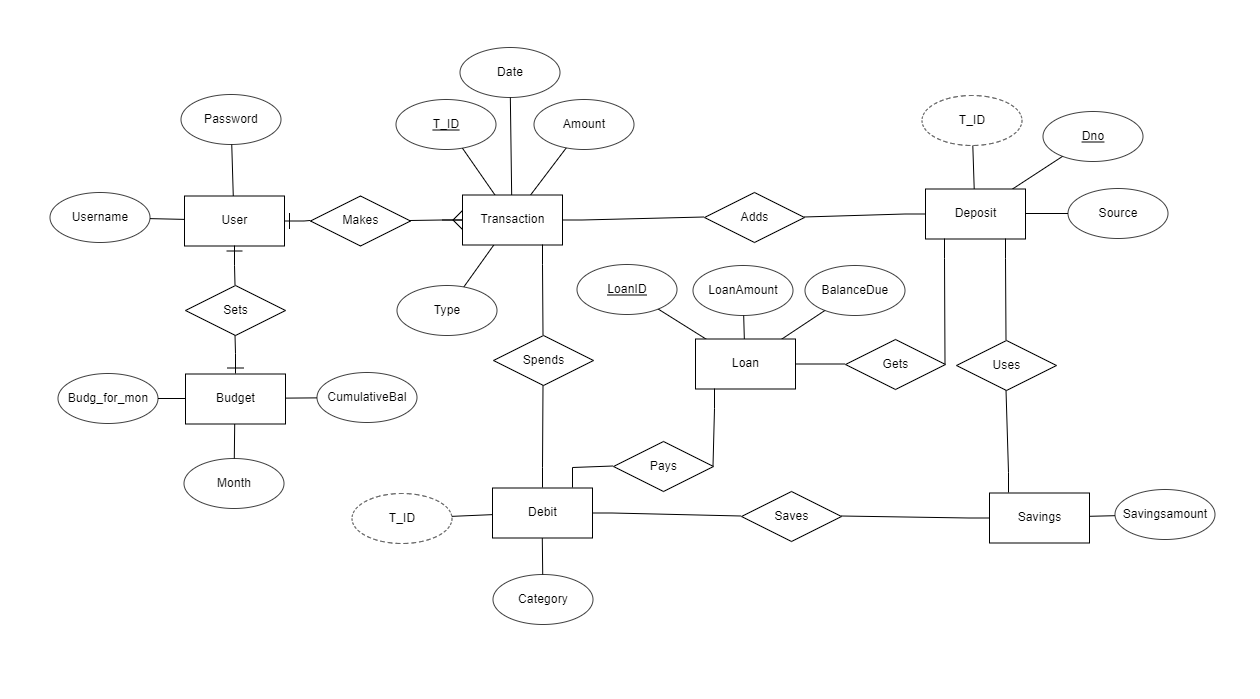




This is a broad level diagram of the project showing a basic overview. Expense Tracker Use Case Diagram: Users interact with the system, adding expenses, viewing, categorizing, and setting budgets. Optional features include generating reports and setting reminders. A streamlined design ensures efficient tracking, empowering users to manage their finances effectively.

## Product Function

Entity Relationship Diagram of Expense Tracker Application



The Online Expense Tracker System offers real-time information on expenses and user financial data. The primary objective is to automate manual processes, facilitating efficient management of expense tracking, calculations, and reporting. The system enables users to record, manage, and analyze expenses, generating reports tailored to user preferences. Acting as the administrator, the user has control over their financial activities, akin to the librarian's role in the library system. The software maintains a database of expense transactions, allowing users to retrieve and view their financial data. Valid users can access their account information, promoting transparency and user empowerment. Overall, the Expense Tracker System aims to streamline financial management, reduce manual efforts, and provide valuable insights for users.

## User Classes and Characteristics

The system provides different types of services for the users.

The features that are available to the Users are:-

* + - A user can add deposit/balance to the account.
    - Can set budget for the month
    - Can enter the expenses and categories.
    - Can view the report of the spend amount.
    - Can view existing loan and savings.

## Operating Environment

The product will be operating in windows environment. JavaFx application is compatible with various versions of Windows like Windows 7,Windows 8,Windows 10,Windows 11.

The hardware configuration include Hard Disk: 40 GB, Monitor: 15” Color monitor, Keyboard: 122 keys. The basic input devices required are keyboard, mouse and output devices are monitor, printer etc.

## Assumptions and Dependencies

The assumptions are:-

* + - The coding should be error free
    - The system should be user-friendly so that it is easy to use for the users
    - The information of all expenses and transactions must be stored in a database that is accessible by the application.
    - The system should have more storage capacity and provide fast access to the database
    - The system should provide search facility and support quick transaction
    - Users must have their correct usernames and passwords to enter into their accounts

and do actions

The dependencies are:-

* + - The specific hardware and software due to which the product will be run
    - On the basis of listing requirements and specification the project will be developed and run
    - The end users (admin) should have proper understanding of the product
    - The system should have the general report stored
    - The information of all the users must be stored in a database that is accessible by the Library System
    - Any update regarding the book from the library is to be recorded to the database and the data entered should be correct.

## Requirement

Software Configuration:-

This software package is developed using java as front end which is supported by sun micro system. Microsoft SQL Server as the back end to store the database.

Operating System: Windows NT, windows 98, Windows XP Language: Java Runtime Environment, Net beans 7.0.1 (front end) Database: My SQL Server (back end)

Hardware Configuration:-

Processor: Pentium(R)Dual-core CPU Hard Disk: 40GB

RAM: 256 MB or more

## Data Requirement

The inputs consist of the query to the database and the output consists of the solutions for the query. The output also includes the user receiving the details of their accounts. In this project the inputs will be the queries as fired by the users like add expenses and transactions . Now the output will be visible when the user requests the server to get details of their account in the form of report based on the categories.

# External Interface Requirement

## GUI

The Expense Tracker application focuses on a streamlined and user-centric interface, designed for optimal user experience in managing individual expenses. With a simple and intuitive design, users can effortlessly create, update, and view their expenses. The system enhances efficiency by providing quick report generation, allowing users to gain insights into their spending patterns. Robust search functionality enables users to find specific expenses based on type or description. The customizable interface ensures adaptability to individual preferences. The login interface ensures secure access, providing clear error messages for any credential-related issues. All modules seamlessly integrate into a unified interface, promoting a cohesive and straightforward user experience. This tailored interface caters exclusively to user needs, enhancing the efficiency and user-friendliness of the Expense Tracker application

# System Features

The users of the system should be provided the surety that their account is secure. This is possible by providing:-

* User authentication and validation of members using their unique member ID
* Proper monitoring by the administrator which includes updating account status, showing a popup if the member attempts to issue number of books that exceed the limit provided by the library policy, assigning fine to members who skip the date of return
* Proper accountability which includes not allowing a member to see other member’s account. Only administrator will see and manage all member accounts

# Other Non-functional Requirements

## Performance Requirement

The proposed system that we are going to develop will be used as the Chief performance system within the different campuses of the university which interacts with the university staff and students. Therefore, it is expected that the database would perform functionally all the requirements that are specified by the university.

* + - The performance of the system should be fast and accurate
    - Library Management System shall handle expected and non-expected errors in ways that prevent loss in information and long downtime period. Thus it should have inbuilt error testing to identify invalid username/password
    - The system should be able to handle large amount of data. Thus it should accommodate high number of books and users without any fault

## Safety Requirement

The database may get crashed at any certain time due to virus or operating system failure. Therefore, it is required to take the database backup so that the database is not lost. Proper UPS/inverter facility should be there in case of power supply failure.

## Security Requirement

* + - System will use secured database
    - Normal users can just read information but they cannot edit or modify anything except their personal and some other information.
    - System will have different types of users and every user has access constraints
    - Proper user authentication should be provided
    - No one should be able to hack users’ password
    - There should be separate accounts for admin and members such that no member can access the database and only admin has the rights to update the database.

## Requirement attributes

* + - There may be multiple admins creating the project, all of them will have the right to create changes to the system. But the members or other users cannot do changes
    - The project should be open source
    - The Quality of the database is maintained in such a way so that it can be very user friendly to all the users of the database
    - The user be able to easily download and install the system

## Business Rules

A business rule is anything that captures and implements business policies and practices. A rule can enforce business policy, make a decision, or infer new data from existing data.This includes the rules and regulations that the System users should abide by. This includes the cost of the project and the discount offers provided. The users should avoid illegal rules and protocols. Neither admin nor member should cross the rules and regulations.

## User Requirement

User Requirements for the Expense Tracker Application:

The Expense Tracker application caters to users who are both general members and administrators. Members possess basic computer and internet browsing knowledge, while administrators, functioning as system maintainers, require a deeper understanding of system internals to address potential issues like disk crashes or power failures. A user-friendly interface, comprehensive user manual, online help, and installation guides are crucial for ensuring seamless system utilization. Administrators offer essential features such as backup and recovery, password retrieval, data migration for initial user registration, data replication for redundancy, auto-recovery through frequent auto-saving, efficient file organization, and regular server maintenance and updates. These functionalities collectively contribute to a robust expense tracking system, ensuring user education, data integrity, and system reliability for both members and administrators.The admin provides certain facilities to the users in the form of:-

* + - Backup and Recovery
    - Forgot Password
    - Data migration i.e. whenever user registers for the first time then the data is stored in the server
    - Data replication i.e. if the data is lost in one branch, it is still stored with the server
    - Auto Recovery i.e. frequently auto saving the information
    - Maintaining files i.e. File Organization
    - The server must be maintained regularly and it has to be updated from time to time

# Other Requirements

## 6.1 Appendix

A: Admin, Abbreviation, Acronym, Assumptions; B: Books, Business rules; C: Class, Client, Conventions; D: Data requirement, Dependencies; G: GUI; K: Key; L: Library, Librarian; M: Member; N: Non-functional Requirement; O: Operating environment; P: Performance,Perspective,Purpose; R: Requirement, Requirement attributes; S: Safety, Scope, Security, System features; U: User, User class and characteristics, User requirement;

## 6.2 Glossary

The following are the list of conventions and acronyms used in this document and the project as well:

* + - User: A general login id assigned to most users
    - Client: Intended users for the software
    - SQL: Structured Query Language; used to retrieve information from a database
    - SQL Server: A server used to store data in an organized format
    - Layer: Represents a section of the project
    - User Interface Layer: The section of the assignment referring to what the user interacts with directly
    - Application Logic Layer: The section of the assignment referring to the Web Server. This is where all computations are completed
    - Data Storage Layer: The section of the assignment referring to where all data is recorded
    - Use Case: A broad level diagram of the project showing a basic overview
    - Class diagram: It is a type of static structure diagram that describes the structure of a system by showing the system’s cases, their attributes, and the relationships between the classes
    - Interface: Something used to communicate across different mediums
    - Unique Key: Used to differentiate entries in a database

## Class Diagram

A class is an abstract, user-defined description of a type of data. It identifies the attributes of the data and the operations that can be performed on instances (i.e. objects) of the data. A class of data has a name, a set of attributes that describes its characteristics, and a set of operations that can be performed on the objects of that class. The classes’ structure and their relationships to each other frozen in time represent the static model.

