SOFTWARE REQUIREMENTS SPECIFICATION

**For**

**EXPENSE TRACKER**

**Prepared by:-**

*S.NAVEEN KUMAR*

*S.MAHALAKSHMI*

*P.MIRNALINI*

# Introduction

## Purpose

The main objective of this document is to manage one’s business and personal finances, analyse their spending patterns, and make informed financial decisions .By using software for managing expense tracking will help to control unnecessary expenses.The main reason you should track your expenses is to identify and eliminate wasteful spending habits in your financial life. Moreover, consistently tracking your expenses will help you maintain control of your finances, and promote better financial habits like saving and investing.  
 For many small businesses, expense tracking may include fixed expenses like rent and utilities as well as fluctuating costs like labor, product orders, and advertising. It’s especially important to record one-time and variable costs, as these aren’t always accounted for in the initial budget.

## 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

Managing personal finances is essential for maintaining a healthy financial life. One effective way to achieve this is by keeping track of expenses. Tracking expenses throughout a application provides you the ability to view various expense categories and time periods. This can help you understand how much money you spend on certain area of living while staying within your budget.  
 By tracking your expenses, you can make more informed decisions than if you didn't know how much you spent. A Expensive tracker helps you budget your money so that you use it wisely. If you find that every month your expenses are more than what you earn, it is time to put your house in order and get a money manager app that keeps track of your money without any problem.  
 A Expensive tracker helps you budget your money so that you use it wisely. If you find that every month your expenses are more than what you earn, it is time to put your house in order and get a money manager app that keeps track of your money without any problem.

## 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

1. \*"Clean Code: A Handbook of Agile Software Craftsmanship" by Robert C. Martin:\* While not specifically about expense tracking, this book focuses on writing clean, maintainable, and efficient code. It's crucial when developing any software project, including expense tracking applications, to have a solid foundation in writing clean code.

2. \*"Pro Angular" by Adam Freeman:\* If you're considering using Angular as the framework for your expense tracking project, this book provides comprehensive guidance on building applications with Angular. It covers various aspects of frontend development using Angular, which could be beneficial for your project.

3. \*"Building Microservices" by Sam Newman:\* This book is a valuable resource if you plan to design and implement your expense tracking project using a microservices architecture. It covers principles, patterns, and practices for building scalable and resilient microservices-based systems.\

* Websites

<https://github.com/jakubgarfield/expenses>

<https://medium.com/search?q=expense+tracking+project>

# Overall Descriptions

## User Classes and Characteristics

*User Classes:*

* Administrator: Has full access to all features and functionalities of the tracker app, including managing user accounts, permissions, and system settings.
* Manager/Supervisor: Responsible for overseeing and monitoring the tracking activities, generating reports, and managing teams.
* Employee/User: Utilizes the tracker app to log and track various activities, such as time, expenses, tasks, or locations.
* Support/Helpdesk: This user class focuses on providing assistance and resolving any issues or inquiries related to the tracker app.

*Characteristics:*

* User-Friendly Interface: The application should have an intuitive and easy-to-use interface, allowing users to navigate and perform tasks effortlessly.
* Customizable Features: The tracker app should offer customization options to adapt to different tracking requirements, such as adding custom fields, defining workflows, or creating personalized reports.
* Data Security: As an expensive tracker app, it should prioritize data security by implementing encryption, access controls, and regular backups to protect sensitive information.
* Scalability and Performance: The application should be capable of handling a large number of users and processing a high volume of tracking data without compromising performance.
* Reporting and Analytics: The tracker app should provide comprehensive reporting and analytics features, allowing users to generate insightful reports, visualize data, and make informed decisions.
* Real-Time Tracking: The application should offer real-time tracking capabilities, allowing users to monitor activities and receive updates instantly.
* Notifications and Alerts: Users should be able to set up notifications and alerts for specific events or thresholds, ensuring timely actions and reminders.
* Data Visualization: The tracker app should provide visual representations of data, such as charts, graphs, or maps, to facilitate easy interpretation and analysis.
* Offline Mode: It might be beneficial for the application to have an offline mode, allowing users to continue tracking activities even when not connected to the internet, with data syncing once the connection is restored.
* Multilingual Support: If the tracker app is used in a multilingual environment, it should support multiple languages to cater to diverse user needs.
* Mobile Accessibility: To enhance usability and convenience, the application should have a mobile version or be accessible through mobile devices, enabling users to track activities on the go.

## Operating Environment

An expense tracker can be designed to run on various platforms, such as:

1. Desktop: The expense tracker can be developed as a desktop application that runs on operating systems like Windows, macOS, or Linux. This allows users to install and run the application directly on their computers.

2. Mobile: Another option is to develop the expense tracker as a mobile app for iOS or Android devices. This provides the convenience of accessing and managing expenses on the go.

3. Web: An expense tracker can also be built as a web application, accessible through a web browser on any device with internet connectivity. This allows for easy access and cross-platform compatibility.

## Assumptions and Dependencies

The assumptions are:-

* + - The coding should be error free
    - The application should be user-friendly so that it is easy to use for the users
    - The application should have more storage capacity and provide fast access to the database
    - The application should provide support quick transactions
    - The data entered by users is accurate and that any calculations performed by the expense tracker are based on correct input.
    - The Expense Tracker is running 24 hours a day
    - The application assumes compatibility with common devices such as smartphones, tablets, and computers.
* The application assumes compliance with relevant financial and data protection regulations.

The dependencies are:-

* The system depends on specific hardware and software configurations to operate effectively, ensuring compatibility and optimal performance.
* The project development is dependent on meeting specified requirements and adhering to project specifications to deliver a functional and reliable expense tracker.
* The end users, particularly administrators, should have a proper understanding of the expense tracker system to effectively manage and oversee financial transactions.
* The system is dependent on a reporting mechanism to generate and store relevant financial reports for users.
* The information related to user expenses must be integrated with a database accessible by the expense tracker system to ensure data accuracy and consistency.
* Any updates or changes in expense data, such as new transactions, must be accurately recorded in the database to maintain the integrity of financial records.

## Requirement

*Software Configuration:-*

* + Front-end: Java FX
  + Back-end: My SQL
  + Operating System: Windows NT, Windows 98, Windows XP
  + IDE : Eclipse

*Hardware Configuration:-*

- Processor: Pentium(R) Dual-core CPU

- Hard Disk: 500 MB or More

- RAM: 256MB 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 adding money to their account and debit money from the account. Now the output will be visible when the user requests the server to get details of their account balance, the area of expense

# External Interface Requirement

## GUI

The graphical user interface (GUI) of the expense tracking software should provide a user-friendly experience for both users and administrators. Here are some key features of the GUI:

*Quick Reports:*

Users can easily view reports such as Book Issued/Returned within a specific time frame.

*Stock Verification and Search:*

The GUI should allow users to search for books based on different criteria and perform stock verification.

*Customizability:*

The administrator should be able to customize the user interface according to their preferences.

*Standard Design:*

All modules within the software should follow a standardized template and have a simple design.

*Login Interface:*

Users can register and create their accounts. The login interface requires users to enter their username and password, displaying an error message if incorrect credentials are entered.

*Search:*

Users can search for specific books by entering the book's type or title.

*Categories View:*

The GUI should display the available book categories and allow the librarian to add, edit, or delete categories.

# 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 with the minimum balance provided by the application policy, assigning alert to members who’s account balance is reaches minimum.
* Proper accountability which includes not allowing a member to see other member’s account. Only administrator will see and manage all member account

# 5.Other Non-functional Requirements

Here are some non-functional requirements for an expense tracker system using Java and a DBMS:

1. Performance: The system should be able to handle a large number of users and transactions efficiently, with minimal response time and latency.

2. Scalability: The system should be able to handle an increasing number of users and data without compromising performance or stability. It should be easy to scale up by adding more resources or scaling out by distributing the workload across multiple servers.

3. Reliability: The system should be highly reliable, with minimal downtime and the ability to recover quickly from failures. It should have mechanisms in place to handle errors, exceptions, and data integrity issues.

4. Security: The system should ensure the confidentiality, integrity, and availability of user data. It should implement secure authentication and authorization mechanisms to prevent unauthorized access. It should also protect against common security threats such as SQL injection and cross-site scripting.

5. Usability: The system should have a user-friendly interface that is easy to navigate and understand. It should provide clear instructions and feedback to users, making it intuitive to add, view, and manage expenses.

6. Maintainability: The system should be designed and implemented in a way that makes it easy to maintain and enhance. It should follow coding best practices, use modular and reusable code, and have proper documentation to facilitate future updates and bug fixes.

7. Compatibility: The system should be compatible with different operating systems, web browsers, and devices. It should be accessible from various platforms, such as desktops, laptops, and mobile devices, to provide a seamless user experience.

8. Data Backup and Recovery: The system should have regular and automated data backup mechanisms to prevent data loss. It should also have a robust recovery plan in case of system failures or disasters.

9. Extensibility: The system should be designed in a way that allows for easy integration with other systems or modules. It should have well-defined interfaces and support for APIs, making it flexible to adapt to future changes or additions.

10. Auditability: The system should have a robust logging mechanism to track and record user activities, changes to data, and system events. This audit trail can be useful for troubleshooting, compliance purposes, and detecting any unauthorized or suspicious activities.

11. Internationalization and Localization: The system should support multiple languages and cultural preferences. It should have the ability to display user interfaces, messages, and reports in different languages and formats, catering to a diverse user base.

12. Accessibility: The system should comply with accessibility standards to ensure that users with disabilities can access and use the application. It should provide features such as keyboard navigation, screen reader compatibility, and adjustable font sizes.

13. Performance Monitoring: The system should have built-in performance monitoring tools to track and analyze system performance metrics, such as response time, CPU and memory usage, and database query performance. This information can help identify bottlenecks and optimize system performance.

14. Data Privacy: The system should adhere to data privacy regulations and protect user data from unauthorized access or disclosure. It should implement appropriate data encryption, anonymization, and data retention policies to ensure the privacy and confidentiality of user information.

15. Error Handling and Reporting: The system should have robust error handling mechanisms to gracefully handle exceptions and provide meaningful error messages to users. It should also have a mechanism to report errors and exceptions to system administrators for timely resolution.

# 6.Other Requirements

## 6.1Data and Category Requirement

There are different categories of transaction namely transport,food,clothes,entertainment,etc. Depending upon the category of user the access rights are decided.It means if the user is an administrator then he can be able to modify the data,delete, append etc. All other users except the Librarian only have the rights to retrieve the information about database. Similarly there will be different categories of transaction available. According to the categories of transaction their relevant data should be displayed. The categories and the data related to each category should be coded in the particular format.

## 6.2Appendix

A: Admin, Abbreviation, Acronym, Assumptions; C: Class, Client, Conventions; D: Data requirement, Dependencies; E:Expense; G: GUI; I:Internationalization; N: Non-functional Requirement; O: Operating environment; P: Performance,Perspective,Purpose; R: Requirement,Recovery; S: Safety, Scope, Security, System features; U: User, User class and characteristics, User requirement;

## 6.3Glossary

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

* **Expense Tracker:** A software application designed to manage and track personal and business expenses, providing insights into spending patterns.
* **User-Friendly Interface:** An interface that is easy to use and navigate, ensuring a positive user experience with minimal complexity.
* **Data Security:** Measures implemented to protect user data from unauthorized access, ensuring confidentiality and integrity.
* **Scalability:** The ability of the system to handle an increasing number of users and data without compromising performance.
* **Real-Time Tracking:** The capability of the application to provide immediate updates and monitoring of activities as they occur.
* **Multilingual Support:** The feature that allows the application to support multiple languages to cater to a diverse user base.
* **Mobile Accessibility:** The ability of the application to be accessed and used on mobile devices, enhancing usability on the go.
* **GUI (Graphical User Interface):** The visual interface of the software that allows users to interact with the application through graphical elements.
* **Auditability:** The system's capability to maintain a detailed log of user activities, changes to data, and system events for auditing purposes.
* **Compatibility:** The ability of the system to function seamlessly across different platforms, browsers, and devices.