**SOFTWARE REQUIREMENTS SPECIFICATION**

**FOR**

**EXPENSE TRACKING**

PREPARED BY: YALINI S,

SRINIDHI PS,

SURRYA BHAARATHI SA.

# Introduction:

# Purpose:

# This document aims to outline the project requirements for the Expense Tracker. The application is created to offer users an effective means of documenting, organizing, and examining their expenses. It provides a user-friendly and uncomplicated text-based interface. Expense trackers serve as a tool for individuals and businesses to monitor their expenditures, helping them understand where their money goes. These tools are valuable for financial management, aiding in the organization of finances and gaining insights into spending patterns.

# 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

# The scope of an expense tracker encompasses a comprehensive set of features, functionalities, and objectives that the application or system strives to accomplish. It entails a meticulous outline of what the expense tracker is specifically designed to do, such as tracking and categorizing expenses, generating financial reports, setting budget limits, and providing real-time expenditure analysis. Furthermore, the scope identifies the underlying problems that the expense tracker aims to address, such as financial disorganization, overspending, or difficulty in budget management.

# Lastly, the scope highlights the numerous benefits that the expense tracker offers, including improved financial awareness, better expense control, increased savings, and enhanced overall financial well-being.

# 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" by Michael Jackson

# "Requirements Engineering: Fundamentals, Principles, and Techniques" by Klaus Pohl and Chris Rupp

# "The Pragmatic Programmer: Your Journey to Mastery" by Andrew Hunt and David Thomas

# Website – (<https://goodbudget.com/>)

# (<https://www.expensify.com/>)

# Overall Descriptions

# 2.1 User Classes and Characteristics

# The system provides different kind of services based on the user's desires.

# Expense Entry

# Users can manually input expenses, including details such as amount, date, and category. Expenses should be stored persistently for future reference.

# Categories

Expenses support a predefined set of categories like groceries, utilities, entertainment and Income support a predefined set of categories like card, wallet, savings.

* View Expenses and Monthly Summary

Display a list of recorded expenses with details such as date, amount, and category and provide a summary of total expenses for the current month category wise.

**2.2 Operating Environment**

There are also expense tracker software programs that are designed to be installed and run directly on desktop or laptop computers. These programs usually provide more advanced features and customization options compared to mobile or web-based options.

**2.3 Assumptions and Dependencies**

**Assumptions:**

* Users of the expensive tracker are willing to spend a higher amount of money for advanced features and functionalities.
* The expensive tracker offers additional benefits such as longer battery life, more accurate tracking, and advanced analytics.
* There is a market demand for high-end trackers among fitness enthusiasts, professional athletes, and individuals who prioritize tracking accuracy and comprehensive data analysis.
* The expensive tracker is built using high-quality materials and components to ensure durability and longevity.
* The expensive tracker includes a user-friendly and intuitive interface to provide a seamless tracking experience

**Dependencies:**

* The expensive tracker relies on advanced sensor technology and algorithms to accurately track and analyze data such as heart rate, steps taken, distance covered, calories burned, and sleep patterns.
* The expensive tracker requires a reliable wireless connection to synchronize data with a mobile app or computer software for data visualization and analysis.
* The expensive tracker may depend on a dedicated mobile app or software platform developed by the manufacturer, which needs to be regularly updated and supported with bug fixes and feature enhancements.
* The expensive tracker may need to be charged regularly, depending on the battery life, and hence relies on an electrical power supply for charging.
* The expensive tracker may have dependencies on third-party services (e.g., GPS satellites for location tracking or weather data providers for weather-related features) to offer enhanced functionalities.
* The expensive tracker's manufacturing and distribution depend on a supply chain for sourcing components, assembly, and logistics.
* The expensive tracker's success relies on effective marketing and promotion to reach the target audience and convince them of its value proposition.

# 2.4 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: MS SQL Server (back end).

# Hardware Configuration:-

# Processor: Pentium(R)Dual-core CPU

# Hard Disk: 40GB

# RAM: 256 MB or more

# 2.5 Data Requirement

# In this expense tracking project, users can input queries into the database. The output of the queries includes solutions to the user's expense tracking needs. Additionally, when the user requests it, the output also includes details of their accounts, such as the current time, date, and the expenses they have recorded. The server will display the output when the user requests to see the details of their account along with the graphical representation of the same.

# External Interface Requirement

# 3.1: Graphical User Interface (GUI)

# The software must offer a robust graphical interface for both users and administrators, facilitating tasks such as book creation, updates, and detailed viewing. Key features include:

# Quick access to reports, enabling users to check Book Issued/Returned within specific time frames.

# Functionality for stock verification and search based on various criteria.

# Customizability of the user interface for administrators to adapt to specific needs.

# Seamless integration of all software modules into the standardized graphical user interface.

# A simple design approach ensures a uniform and user-friendly experience across different interfaces for the expensive tracker system.

# 1.Login:

# Efficiently manage user access by allowing registration for new users. Once registered, users can log in securely using their username and password. In case of entry errors, a user-friendly error message should guide them.

# 2. Search :

# Empower users to locate valuable items efficiently through the search functionality. Members and administrators can input specific details such as the type or title of the item they are looking for, enabling a swift and precise search experience within the expensive tracker system.

# 3.Categories :

# Implement a robust categorization system within the expensive tracker, allowing administrators to organize resources effectively. This feature enables the addition, editing, or deletion of categories, ensuring systematic management and easy retrieval of valuable items within the tracking system.

**4. System Features**

# The expense tracker system places a high emphasis on user security through the implementation of crucial features. Users undergo authentication using distinctive credentials, and member validation is achieved through their unique member ID. To uphold privacy, the system enforces restricted access, enabling members to exclusively view their individual expense records. Meanwhile, the administrator maintains exclusive access, ensuring accountability and security while overseeing all member accounts within the expense tracking system.

# 5.Other Non-functional Requirements

# 

# Performance: The system should be able to handle a large number of transactions without

# significant slowdown or performance degradation.

# Reliability: The system should be highly reliable, ensuring that expenses are accurately recorded

# and stored without data loss or corruption.

# Security: The expense tracker should have robust security measures in place to protect sensitive

# financial data, including encryption, access control, and authentication mechanisms.

# Usability: The user interface should be intuitive and user-friendly, allowing users to easily

# navigate and record their expenses without needing extensive training.

# Scalability: The system should be able to accommodate a growing number of users and handle

# increased transaction volumes without compromising performance.

# Availability: The expense tracker should be accessible and available to users at all times,

# minimizing downtime for maintenance or upgrades.

# Compatibility: The system should be compatible with various platforms and devices, ensuring

# seamless access and functionality across different operating systems and browsers.

# Auditability: The expense tracker should provide a comprehensive audit trail, allowing users to

# track and review the history of their expenses and any modifications made.

# Backup and Recovery: The system should have regular data backups in place, with

# mechanisms for data recovery in case of system failures or data loss incidents.

# Compliance: The expense tracker should comply with relevant regulations and standards, such

# as data protection laws and financial reporting requirements.

**6. Other Requirements**

**6.1 Data and Category Requirement**

A well-designed expense tracker application necessitates a systematic organization of data,

incorporating crucial elements like expense amount, date, category, description, and payment

method, complemented by income data encompassing attributes such as amount, date, and

source. User profile information, comprising name, email, password, and an optional profile

picture, forms a critical component. The app should empower users to tailor their experience

by creating custom expense and income categories. To provide a comprehensive financial

tracking solution, it should include features like budgeting tools, analytical reports, receipt

attachments, reminders, and robust security measures. These elements collectively contribute

to an effective and user-friendly application, ensuring the customization, analysis, and security

necessary for a seamless financial management experience.

**6.2 Appendix**

A: Admin, Abbreviation, Acronym, Assumptions; B: Bank, Business rules; C: Conventions; D:

Data requirement, Dependencies;E:Expenditure; G: GUI; K: Key; L: Loss; 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;

**6.3 Glossary**

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

Administrator: A login id representing a user with user administration privileges to the software

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