|  |  |
| --- | --- |
|  | **Personal Finance Tracker**  **Technical Design Document** |
| |  |  |  |  | | --- | --- | --- | --- | |  | **Prepared By / Last Updated By** | **Reviewed By** | **Approved By** | | **Name** | Arijit Kumar Haldar |  |  | | **Role** | CSD Intern |  |  | | **Signature** |  |  |  | | **Date** |  |  |  | |
|  |

Table of Contents

[1.0 Introduction 3](#_Toc14171023)

[1.1 Purpose of this document 3](#_Toc14171024)

[1.2 Project overview 3](#_Toc14171025)

[2.0 Solution Summary 3](#_Toc14171026)

[2.1 Scope 3](#_Toc14171027)

[2.2 Assumptions 3](#_Toc14171028)

[2.3 Dependencies 3](#_Toc14171029)

[2.4 Risks 3](#_Toc14171030)

[3.0 Schematic Diagram 3](#_Toc14171031)

[4.0 System Design 4](#_Toc14171032)

[4.1 Proposed design 4](#_Toc14171033)

[4.2 Component inventory 4](#_Toc14171034)

[5.0 Database Design 4](#_Toc14171035)

[5.1 Data Model 4](#_Toc14171036)

[5.2 Tables Structure 4](#_Toc14171037)

[6.0 Appendices 4](#_Toc14171038)

[6.1 Glossary 4](#_Toc14171039)

[6.2 Other 5](#_Toc14171040)

[7.0 Terms & Conditions 5](#_Toc14171041)

[8.0 Change Log 5](#_Toc14171042)

# Introduction

## Purpose of this document

The purpose of this document is to document the technical design, component details and Database design. This will also capture the scope, assumptions, risk, dependencies of this project.

## Project overview

The Personal Finance Tracker is a digital application designed to help users manage their finances effectively. By entering their income and expenditure details, the application generates monthly and yearly reports, calculates the percentage of savings from total income, and predicts savings based on past inputs. The application also includes financial goal tracking, enabling users to set a savings target within a given date in the future, and the system shows the percentage of completion towards achieving that goal. It provides users with valuable insights into their finances, empowering them to make informed decisions and take control of their financial future.

# Solution Summary

## Scope

*The scope of this design document is to provide a comprehensive overview of the design and functionality of the Personal Finance Tracker application. This Spring Boot and Spring Data JPA application with MySQL database is designed to help users manage their finances by providing a user-friendly interface for inputting and tracking income and expenditure details.*

*This document will cover the detailed design of the application's main features, including monthly and yearly reports, percentage savings calculation, financial goal tracking, and savings prediction based on past inputs. It will also outline the data models, architecture, and technical specifications required to build and deploy the application successfully.*

*The intended audience for this document includes anyone who are evaluating the IDP project, Personal Finance Tracker application, which is a mandatory requirement for completion of the CSD program. The design methodology for this application is based on the Spring Boot framework, Spring Data JPA, and MySQL database. Finally, this document assumes that the reader has a basic understanding of software development concepts and related technologies.*

## Assumptions

*The current iteration of the application makes the following assumptions-*

* *The application assumes that the user enters his/her income and expenditure values correctly and regularly.*
* *The application assumes that the dates entered by the users are valid and relevant to the data being entered.*
* *The application assumes that the user has a basic understanding of financial concepts and terminology, including income, expenditure, savings, and financial goals.*
* *The application assumes that the user has access to a reliable internet connection and a compatible device to access the application.*
* *The application assumes that the database connection and server configuration are set up correctly and running smoothly.*
* *The application assumes that the user will use the application within the recommended system requirements, including browser and device compatibility.*
* *The application assumes that any future changes or updates to the operating environment, such as changes to the database or server, will be compatible with the current application design.*

## Dependencies

*Dependencies:*

* *Java Development Kit (JDK 8u362\_b09) [version 1.8]*
* *Apache Maven [version 3.9.1]*
* *Spring Boot Starter Data JPA [version 2.7.11]*
* *Spring Boot Starter Web [version 2.7.11]*
* *Spring Boot DevTools [version 2.7.11]*
* *MySQL Connector/J [version 8.0.33]*
* *Lombok [version 1.18.26]*
* *Springfox Swagger2 [version 2.9.2]*
* *Springfox Swagger UI [version 2.9.2]*
* *Spring Boot Starter Test [version 2.7.11]*
* *Junit Platform Suite Engine [version 1.8.2]*
* *Log4J [version 2.17.2]*
* *SLF4J [version 1.7.36]*
* *Junit [version 5.8.2]*
* *Mockito [version 4.5.1]*
* *Embedded Apache Tomcat Server [version 9.0.74]*
* *Node [version 18.16.0]*
* *Npm [version 9.6.6]*

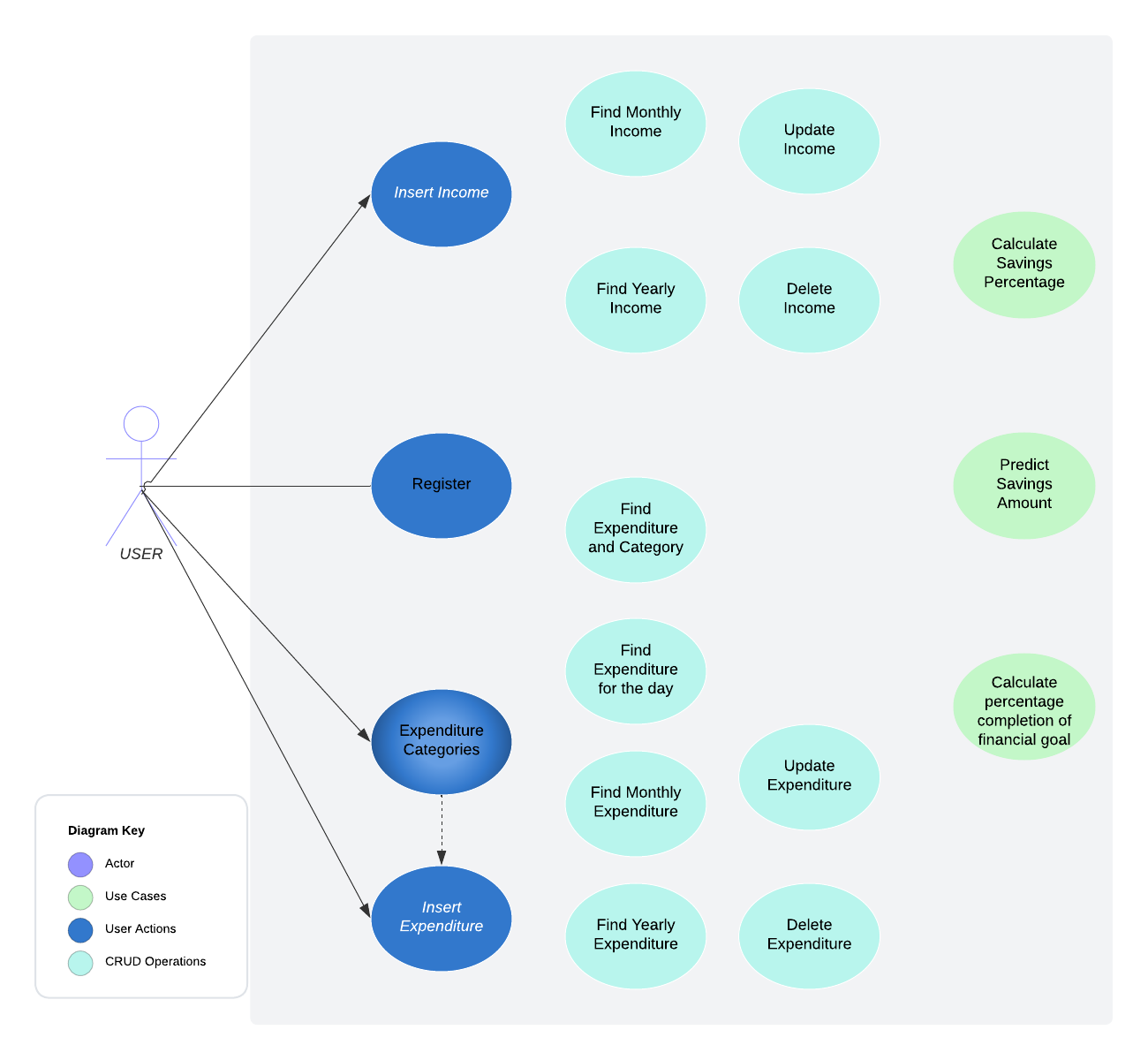
## Risks

*The risks predicted for this project are as follows:*

* *Data security breach may occur, because sensitive information is being saved in database in a plain-text format.*
* *The system may not be able to handle large volumes of data or many API requests at the same time, leading to performance issues.*

# Schematic Diagram

A schematic, or schematic diagram, is a representation of the elements of a [system](https://en.wikipedia.org/wiki/System) using abstract, graphic [symbols](https://en.wikipedia.org/wiki/Symbol) rather than realistic pictures. It gives an overview of overall system



# System Design

## Proposed design

CRUD operations

* Log In: Allows the user to log in to the system using their username and password.
* Sign Up: Allows the user to create a new account by providing their name, email address, username, and password.
* Add Income: Allows the user to add a new income transaction by specifying the amount, date, category and description of the income.
* Update Income
* Delete Income
* Add Expense: Allows the user to add a new expense transaction by specifying the amount, date, category, and description of the expense.
* Update Expense
* Delete Expense
* Add new Expenditure Category
* Delete Expenditure Category
* View Transactions: Allows the user to view their income and expense transactions in a table or list view, with filters and search options.
* Export Data: Allows the user to export their income and expense data in a CSV or PDF format.

Non-CRUD operations

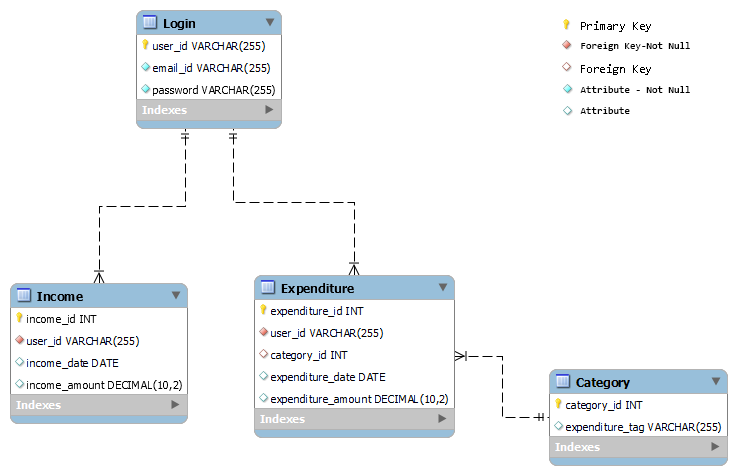
* Calculate Savings: Calculate monthly and yearly percentage of savings based on income and expenditure entered.
* Budget Forecasting: Predict savings for the next month, based on average income and expenditure from current year.
* Set Financial Goals: Allow user to set a certain savings amount for a month for vacation or an event. The tracker would calculate percentage completion towards that target.

## Component inventory

|  |  |
| --- | --- |
| *COMPONENT* | *PURPOSE* |
| Frontend | Implement the user interface using React/Angular |
| Backend | Implement business logics and communicate with the database using Java and MySQL |
| Model/Entity | Represent the data model on Java and map to database tables |
| Repository | Handle persistence and retrieval of data from the database |
| Service | Implement business logic and communicate with repositories |
| Controller | Handle incoming HTTP requests and delegates to services |
| Exceptions | Custom exception classes used for handling errors |
| Configuration | Spring Boot configuration classes |
| Database | MySQL database for storing and retrieving data |
| Unit Tests | JUnit tests for service layer using Mockito and controller layer using MockMVC |
| Logging | Logs application events and errors for debugging and auditing |
| Lombok | Provides annotations to reduce boilerplate code in Java classes |
| Postman | A tool for testing and debugging HTTP requests and responses |
| Swagger | API documentation and testing from web browser |
| React | ReactJS 18.2.0 is used for front-end |
| React Router | React Router Dom v6.11.2 is used to provide routing capability to React app |
| Reactstrap | For using Bootstrap in React |
| React-icons | Used to get icons for rendering in React components |
| Axios | Package used to perform perform Controller API calls from front-end |

# Database Design

## Data Model

**

## Tables Structure

***Login***

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Length** | **Nulls** |
| **user\_id** | VARCHAR | 255 | NO |
| email\_id | VARCHAR | 255 | NO |
| password | VARCHAR | 255 | NO |

***Income***

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Length** | **Nulls** |
| **income\_id** | INT |  | NO |
| *user\_id* | VARCHAR | 255 | NO |
| income\_date | DATE |  | YES |
| income\_amount | DECIMAL | (10,2) | YES |

***Expenditure***

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Length** | **Nulls** |
| **expenditure\_id** | INT |  | NO |
| *user\_id* | VARCHAR | 255 | NO |
| *category\_id* | INT |  | YES |
| expenditure\_date | DATE |  | YES |
| expenditure\_amount | DECIMAL | (10,2) | YES |

***Category***

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Length** | **Nulls** |
| category\_id | INT |  | NO |
| expenditure\_tag | VARCHAR | 255 | YES |

# Appendices

## Glossary

|  |  |
| --- | --- |
| **Acronyms** | **Definitions** |
|  |  |

## Other

# Terms & Conditions

***Disclaimer: Please do not circulate or distribute this document outside of Cognizant Network, We have a Zero Tolerance Policy. Kindly adhere to 100% Compliance at all times.***

# Change Log

*Please note that this table needs to be maintained even if a Configuration Management tool is used.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version Number | Changes made | | | |
| V<n.n> | *<If the change details are not explicitly documented in the table below, reference should be provided here>* | | | |
| Page no | Changed by | Effective date | Changes effected |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |