**Expense Tracker**

**Course** : J2EE Business Components - ITE-5432-0NA

**Professor** : Vitalii Bohudskyi

**Semester :** 4th Semester

**Project Title** : Expense Tracker

**Team members:** Urja Tej Bhemat N01648980

Sameeha Shaik N01649346

Kalwala Siddhartha Reddy N01649317

Kledy Danlees N01652920

**Document for Phase II: MVP Development**

This document details the development of the "Expense Tracker" Minimum Viable Product (MVP), which fulfills the technical and functional requirements of the project. The application is built using the **Spring Framework** and related technologies to create a secure, multi-currency expense tracking platform for international students.

**1. Implementation of Core Features**

The application's backend is built around a secure and robust architecture.

* **Data Models**: The core entities of the application are User, Category, and Expense. The User model includes fields such as name, username, password, and role for authentication and authorization. An expense is uniquely owned by a User and categorized by a Category, establishing a clear data relationship.
* **User Management**: User registration and login functionalities are implemented with security in mind4. When a new user registers, the system validates the input, checks for existing usernames, and uses a password hashing mechanism to securely store credentials in the database.

**Spring Security** is configured to provide role-based access control, ensuring that administrators and regular users have access to different parts of the application6.

* **Expense Management (CRUD)**: The application provides a full suite of **CRUD** (Create, Read, Update, and Delete) operations for expenses. Users can create new expenses, specifying details such as the amount, category, date, and currency. They can view their list of expenses and also edit or delete them. The system includes data validation to ensure the integrity of the expense data.
* **REST APIs**: The backend exposes **RESTful web services** built with **Spring Boot's** RestController. These APIs enable communication with the frontend for all core functionalities. API governance is handled using the

**Open API Specification**, and the endpoints are documented with **Swagger**.

**2. User Interface (UI)**

The frontend is developed using

**Thymeleaf**, **HTML5**, **CSS3**, and **Bootstrap**, creating a responsive and intuitive user experience.

* **Login and Registration Pages**: These pages provide a secure entry point to the application, guiding users through the necessary steps to create an account or log in.
  + [Insert screenshot of your login page here]
  + [Insert screenshot of your registration page here]
* **User Dashboard**: This is the main interface for users. It displays a list of their expenses and provides options to filter them by date, category, or currency.
  + [Insert screenshot of the user dashboard showing a list of expenses and filter options here]
* **Expense Forms**: The application provides dedicated forms for adding and editing expenses, ensuring all required details are captured and validated.
  + [Insert screenshot of the "Add Expense" form here]
  + [Insert screenshot of the "Edit Expense" form here]
* **Admin Pages**: The admin dashboard allows administrators to manage users, including the ability to view all users, search for specific accounts, and delete them.
  + [Insert screenshot of the "View All Users" page, including search and pagination features here]