



## **PES University, Bengaluru**

Department of Computer Science & Engineering

Session: Jan - May 2023

**Subject: Object Oriented Analysis and Design with Java**

**Title: Travel Agency Management System**

**Team members:**

<i>Prashanth Sai Akurathi</i>	<i>PES1UG20CS525</i>
<i>Vishal M</i>	<i>PES1UG20CS508</i>
<i>Ekanath reddy</i>	<i>PES1UG20CS548</i>

### **Project Description**

The Travel Agency Management System is an efficient and effective solution for managing the various processes involved in a travel agency. The proposed system aims to simplify the management of customer information, travel packages, bookings, service and payments for a travel agency.

The system will be developed using the Model-View-Controller (MVC) architecture, which separates the application's data, logic, and presentation layers. The application will be developed using Java using Design principles, as the programming language and MySQL as the database management system.

The Travel Agency Management System will have the following features:

**User Management:** This module will allow the travel agency to manage Users information, including their personal details, booking history, and payment details.

**Package Management:** This module will allow the travel agency to manage the travel packages they offer, including the destinations, pricing, and availability.

**Booking Management:** This module will allow the travel agency to manage the booking process, including creating bookings, modifying bookings, and cancelling bookings.

**Payment Management:** This module will allow the travel agency to manage payment transactions, including payment processing and payment reconciliation.

The proposed system will be developed using the following technologies:

**Java Programming Language:** Java is a widely used programming language, known for its platform independence, scalability, and security features.

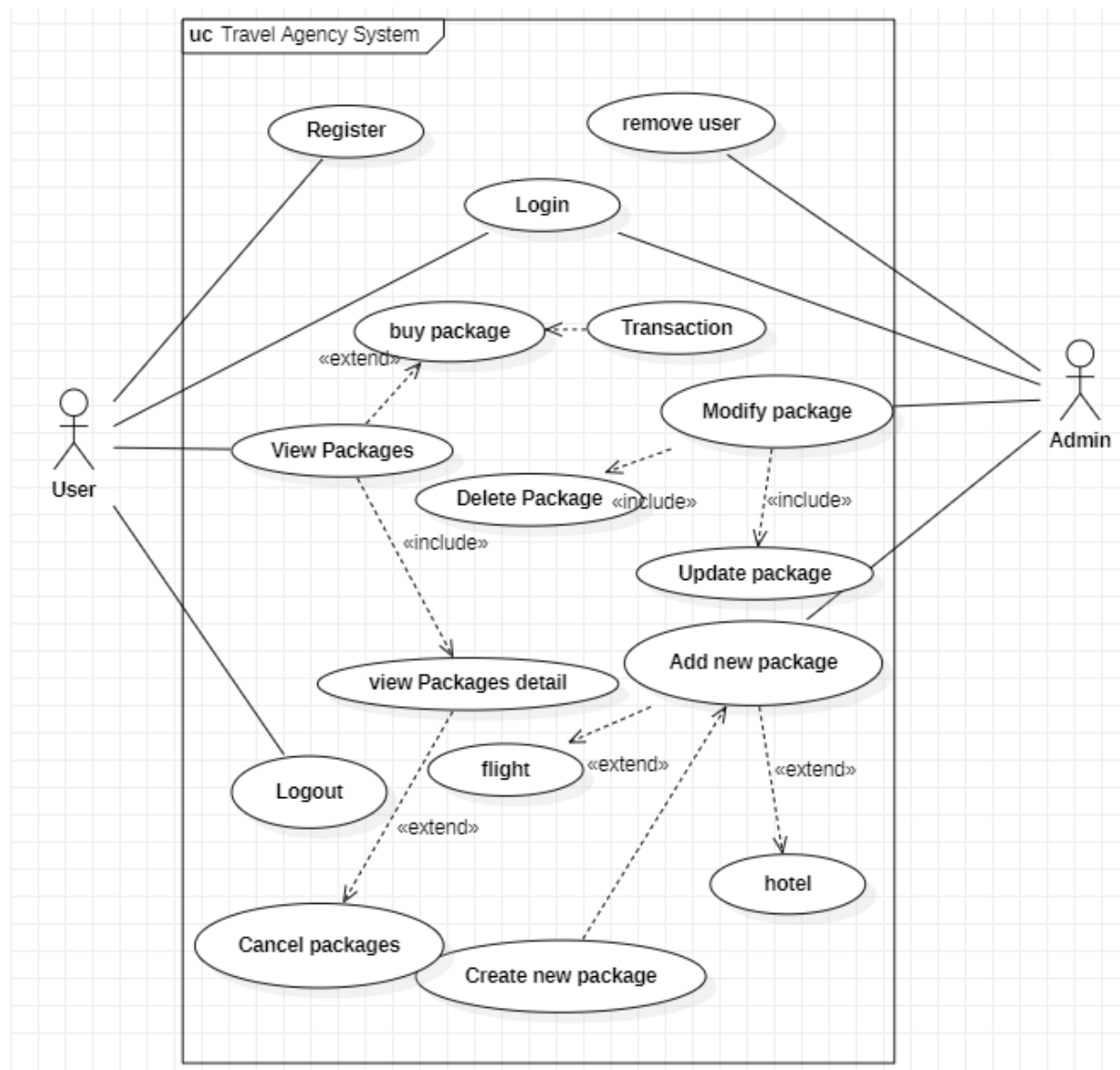
**MySQL Database Management System:** MySQL is a popular open-source database management system known for its reliability, speed, and flexibility.

**Model-View-Controller (MVC) Architecture:** MVC is a design pattern that separates the application's data, logic, and presentation layers, making the application easy to maintain and update.

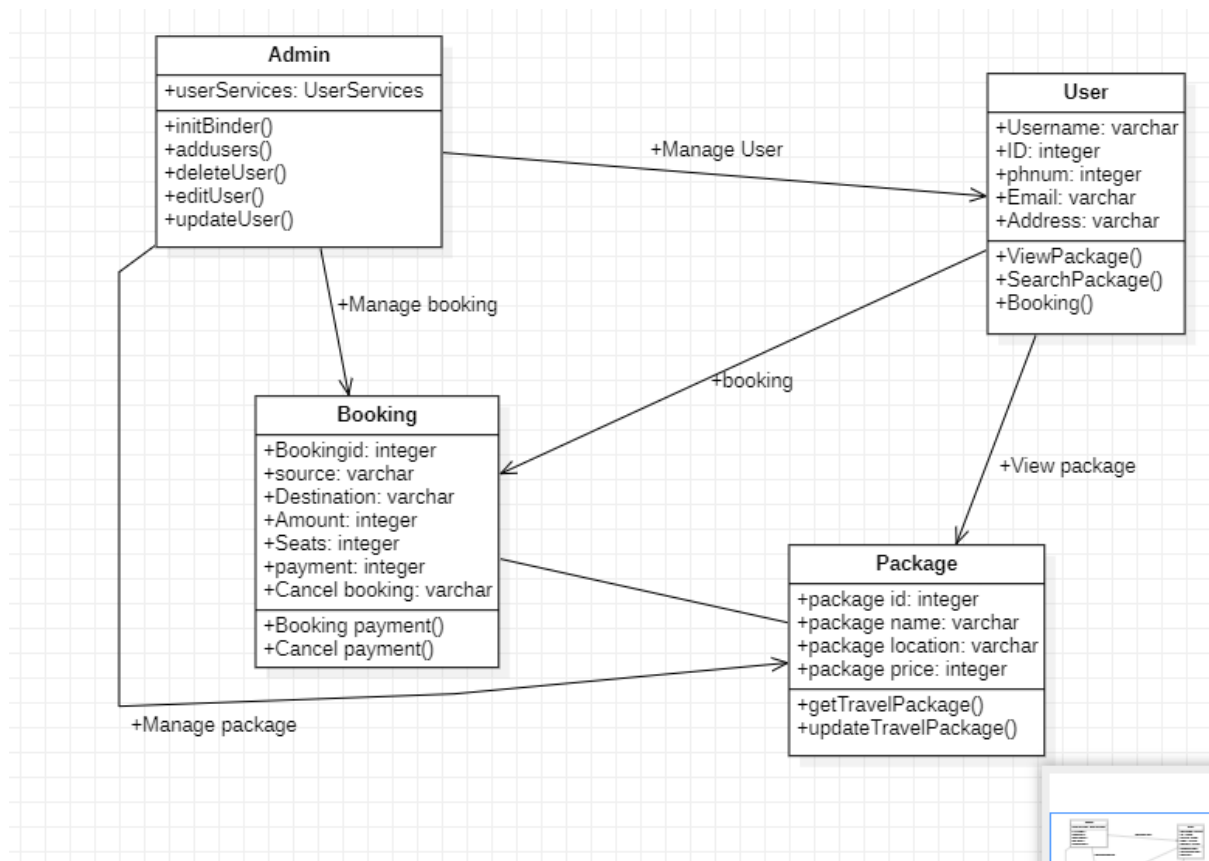
**Spring Boot Framework:** Spring Boot is a powerful framework that simplifies the development of Spring-based applications. It provides out-of-the-box features such as auto-configuration, embedded server, and dependency injection.

The Travel Agency Management System will provide the travel agency with an efficient and effective way to manage their business processes, improve customer service, and increase revenue.

**I. Use Case Diagram:**

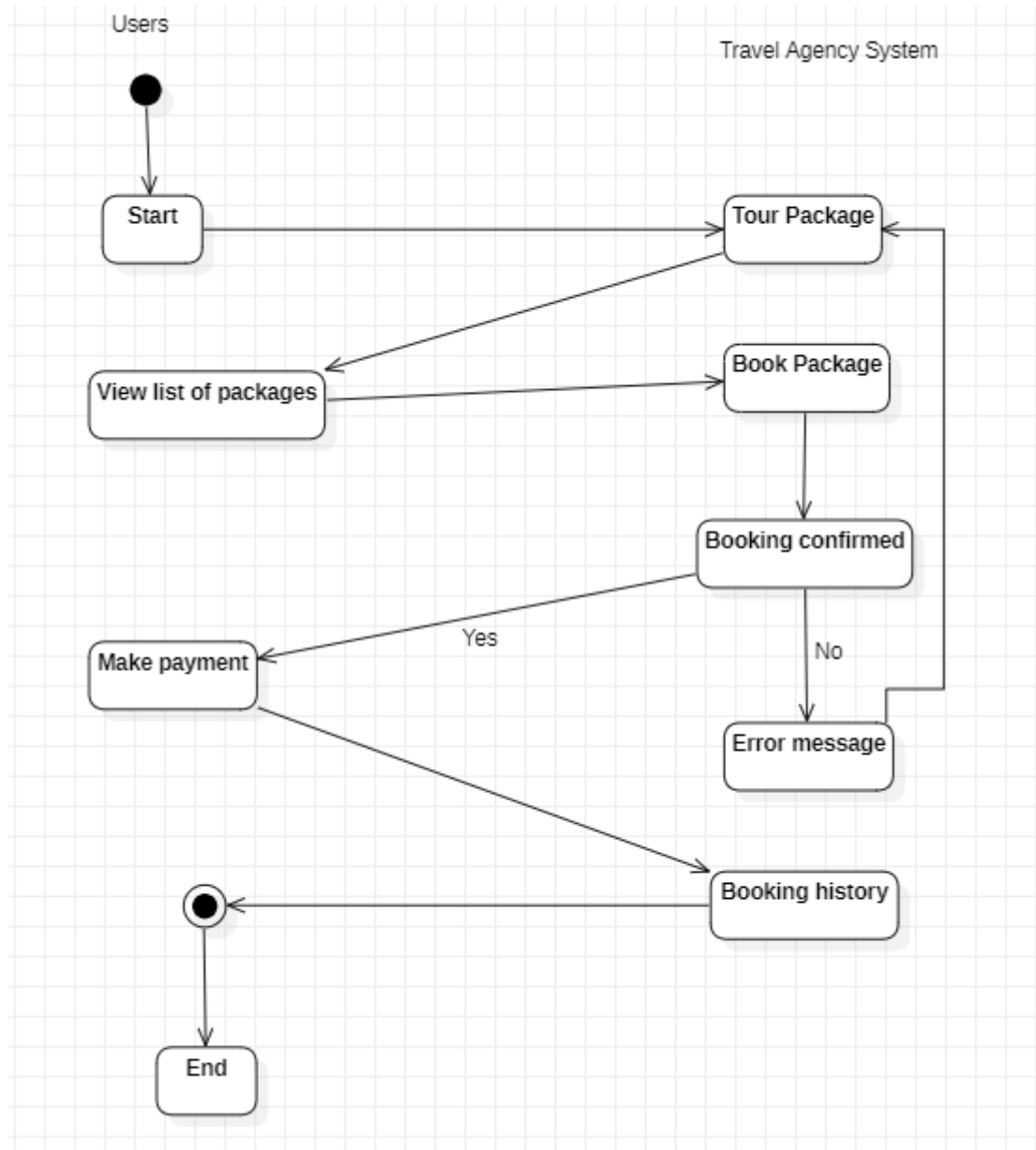


## II. Class Diagram:



### III. Activity Diagram:

**Name:** Booking package



## Architecture Pattern

Architecture Pattern used by Application is **Model View Controller (MVC)** which separates the application into three interconnected parts: the model (data), the view (user interface), and the controller (business logic).

### Design Principles:

#### 1) Single Responsibility Principle (SRP)

- ★ The User Class Has A Single Responsibility, Which Is To Represent The Data Of A User.
- ★ The TravelPackageModel has a SRP.

#### 2) Open/Closed Principle (OCP)

- UsersModel class is designed to be easily extended with new fields and methods without requiring changes to the existing code.

#### 3) Dependency Inversion Principle (DIP)

The UsersModel class depends on abstractions, such as the UsersDetailsModel, TravelPackageModel, and TransactionTravellerModel classes, rather than concrete implementations.

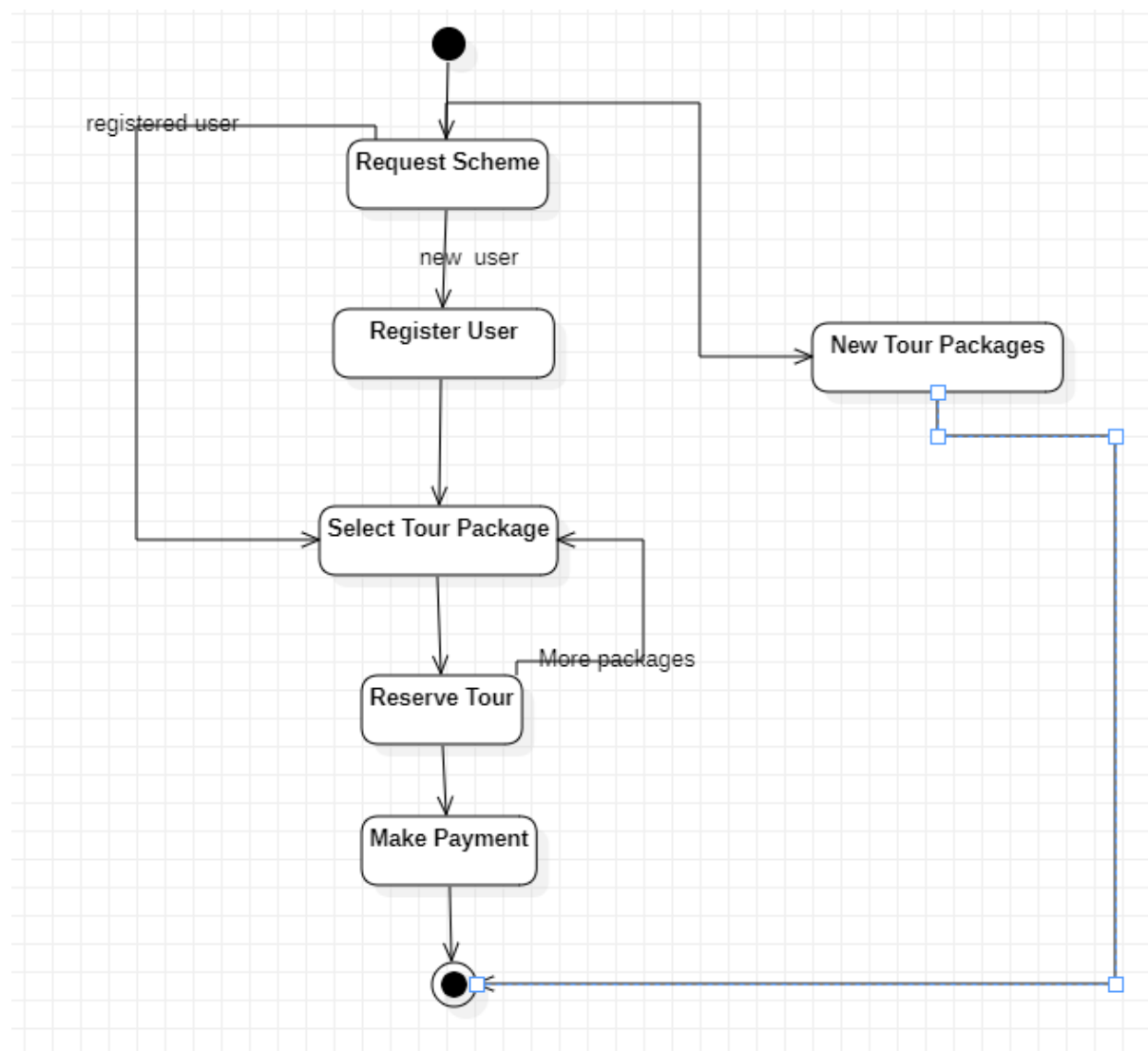
### Design Pattern :

**DAO (Data Access Object) Pattern:** This pattern is used in the TravelPackageDAOImpl and UserDAOImpl classes. It provides an abstract interface for accessing the database and decouples the application logic from the persistence layer.

**Factory Method Pattern:** The TravelPackage class is an example of the Factory Method pattern. It encapsulates the object creation logic for the TravelPackageModel class and provides a centralized location for managing the creation of new objects.

**Repository Pattern:** The TravelPackageRepository class can be seen as an implementation of the Repository pattern. It abstracts the persistence details of the TravelPackageModel from the rest of the application and provides a simple interface for accessing and managing the data.

#### IV. State Diagram:



**Team Contributions:**

PES1UG20CS525	Front end (View)and Controller
PES1UG20CS508	Model
PES1UG20CS548	Data Access Objects (Package, User)