

**UE22CS352B - Object Oriented Analysis & Design**

Telecom Billing System

***Submitted by:***

***Name :*** KAVURU MARUTHI SRINIVAS (PES1UG22CS280)

J V SHANMUGA TEJA (PES1UG22CS261)

KATTERPALLI PRABHATH REDDY (PES1UG22CS278)

LAKSHMIPALLI AKHILESWARA REDDY (PES1UG22CS301)

Semester Section

**6 E**

# Facultly Name

**January - May 2025**

##### DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

FACULTY OF ENGINEERING

##### PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013) 100ft Ring Road, Bengaluru – 560 085, Karnataka, India

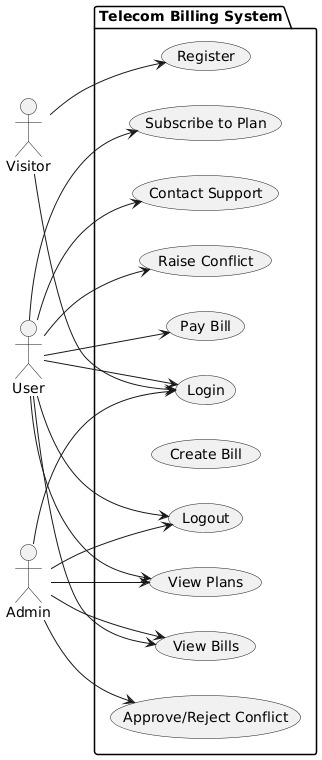
##### Problem Statement: TELECOM BILLING SYSTEM

**Key Features:**

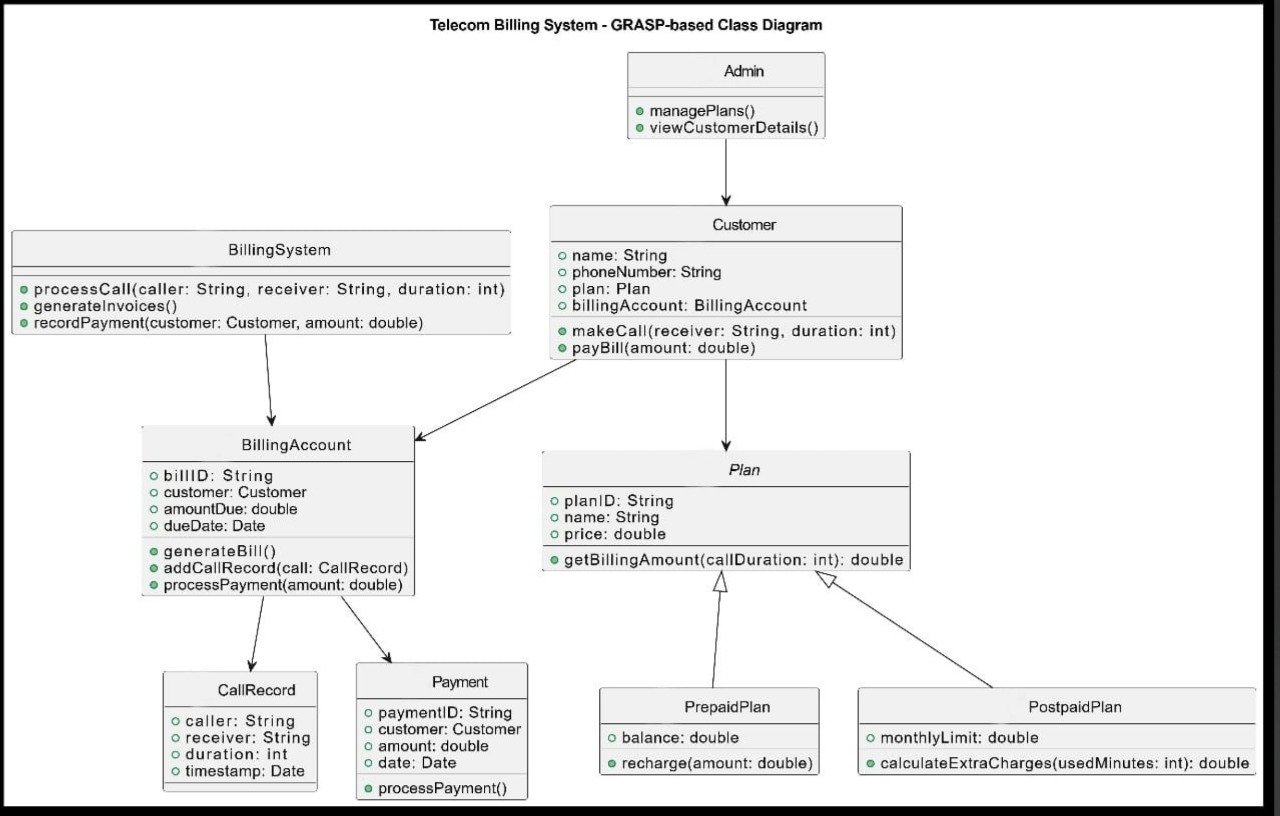
1. **USER MANAGEMENT AND AUTHENTICATION(JWT BASED TOKEN)**
2. **BILLING AND SUBSCRIPTION MANAGEMENT**
3. **PLAN MANAGEMENT**
4. **CONTACT US FUNCTIONALITY**
5. **CONSISTENT UI AND THYMELEAF TEMPLATES CSS STYLING**
6. **MAVEN AND DOCKER INTEGRATION**

## Models:

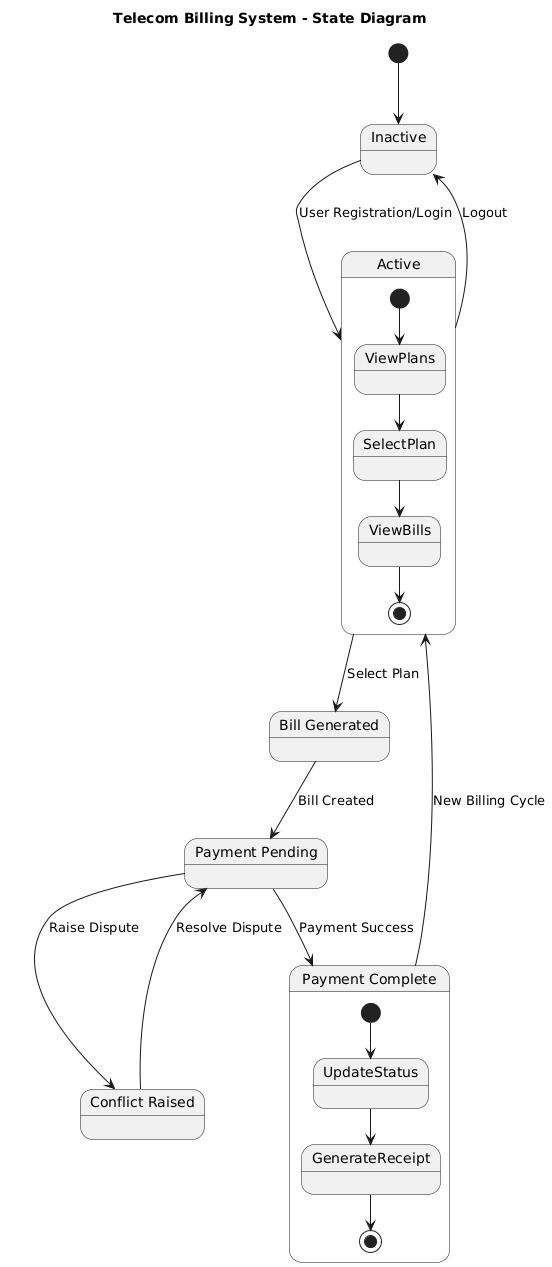
Use Case Diagram:



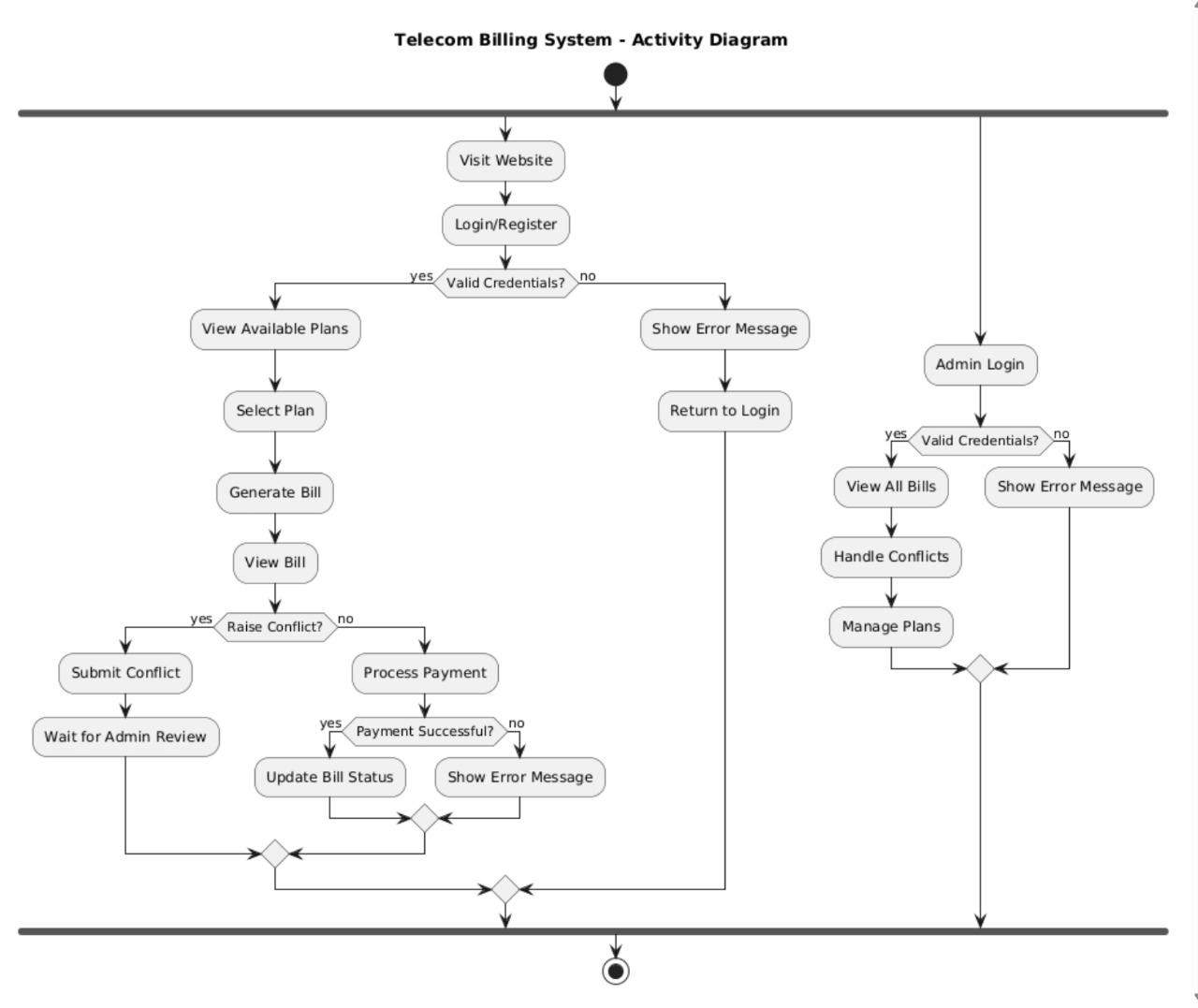
**Class Diagram:**



# State Diagram:



**Activity Diagrams:**



Architecture Patterns, Design Principles, and Design Patterns:

# Architecture Patterns

# Model – View – Controller Pattern (MVC)

# Models: Entity classes like User, Bill, Plan

# Views: Thymeleaf templates in resources/templates

# Controllers: LoginController, BillingController, ContactController, MainController, PlanController, RegistrationController, TokenController

# Design Principles

# Design Patterns

# Authentication Use Cases (Login/Register/Logout)

# Pattern: MVC Pattern

# Implementation: LoginController handles requests

# Pattern: Singleton Pattern

# Implementation: Session management for user authentication

# Plan Management Use Cases

# Pattern: Repository Pattern

# Implementation: PlanRepository interface for data access

# Pattern: DTO Pattern

# Implementation: Plan entity for data transfer

# Bill Generation Use Cases

# Pattern: Factory Pattern

# Implementation: Bill creation in BillingController

# Pattern: Observer Pattern

# Implementation: Bill status updates

# Payment Processing Use Cases

# Pattern: Strategy Pattern

# Implementation: Different payment methods

# Pattern: State Pattern

# Implementation: Bill status transitions (Pending → Successful)

# Conflict Resolution Use Cases

# Pattern: Chain of Responsibility

# Implementation: Conflict handling flow (User → Admin)

# Pattern: Command Pattern

# Implementation: Approve/Reject actions

# User Management Use Cases

# Pattern: Repository Pattern

# Implementation: UserRepository for data access

# Pattern: Decorator Pattern

# Implementation: User role-based access control

# Frontend Templates

# Pattern: Template Pattern

# Implementation: Thymeleaf templates for views

# Pattern: Composite Pattern

# Implementation: HTML component structure

# Data Access

# Pattern: Repository Pattern

# Implementation: JPA repositories

# Pattern: DAO Pattern

# Implementation: Entity classes with JPA

### Github link to the Codebase: https://github.com/Maruthi-srinivas/OOAD

# Screenshots

## UI:

## 

## 

## 

## 

## Admin :

## 

## 

Individual contributions of the team members:

|  |  |
| --- | --- |
| **Name** | **Module worked on** |
| **SHANMUGA TEJA** | CONFLICT MANAGEMENT, CLOUD DEPLOYMENT, SESSION MANAGEMENT |
| **K PRABHATH REDDY** | USER AUTHENTICATION, HOME PAGE, UI |
| **K MARUTHI SRINIVAS** | BILLS MANAGEMENT, PAYMENT, CONTACT US |
| **L AKHILESWARA REDDY** | PLANS MANGEMENT |