**Course Name:** Programming for Big Data

**Course Code:** PROG8420-23S-Sec1

**Professor Name:** Jomis John

**Technical Design Document**

**Members:**

Ravali Gandhavarapu

Mohammed Arbaaz Khan

Vijaya Pasupuleti

Venkata Varun Pechetty

**Technical Design Document: Transwift Inc**

**1. Introduction**

The Logistics Management System is a software application designed to optimize and streamline transportation operations within the logistics domain. It aims to enhance the efficiency, cost-effectiveness, and reliability of transportation processes, including route planning, vehicle scheduling, delivery tracking, driver management, cost calculation, and notifications.

**2. System Architecture**

The system will be implemented using a multi-tier architecture, consisting of the following layers:

**2.1 Presentation Layer**

* User Interface (UI): An Angular-based web application will provide an intuitive and user-friendly interface for users to interact with the system.
* User Authentication and Authorization: Implement user authentication and role-based access control to ensure secure access to system functionalities.

**2.2 Application Layer**

* Backend Services: Develop a Python-based backend using the Flask framework. It will handle the business logic, data processing, and interaction with the database.
* APIs: Expose RESTful APIs to enable communication between the presentation layer and the backend services.
* Authentication and Authorization: Implement authentication and authorization mechanisms to secure the APIs.

**2.3 Data Layer**

* Database: Utilize a relational database management system Microsoft SQL Server to store and manage the application data.
* Tables and Relationships: Define appropriate tables and establish relationships to store information related to routes, vehicles, drivers, deliveries, costs, and user accounts.

**3. User Interface Design**

The user interface of the Transwift application focuses on simplicity and intuitive navigation as a single page application after the user login. The different tabs like Driver, Vehicle, Route, Cost Estimates. Each tab will let the users perform various activities like add vehicle, assign vehicle, add driver, assign vehicle, specific route, cost calculations.

* Dashboard: Helps to navigate different tabs such as driver, vehicle, route.
* Driver Module: Add and assign drivers to vehicles and deliveries based on availability and requirements.
* Vehicle Module: Add and assign vehicles for deliveries based on make, model and capacity.
* Route Module: Allow users to add origin and destinations to find out a feasible route.

**3.1 Data Mapping Diagram**

**A picture containing text, diagram, screenshot, rectangle

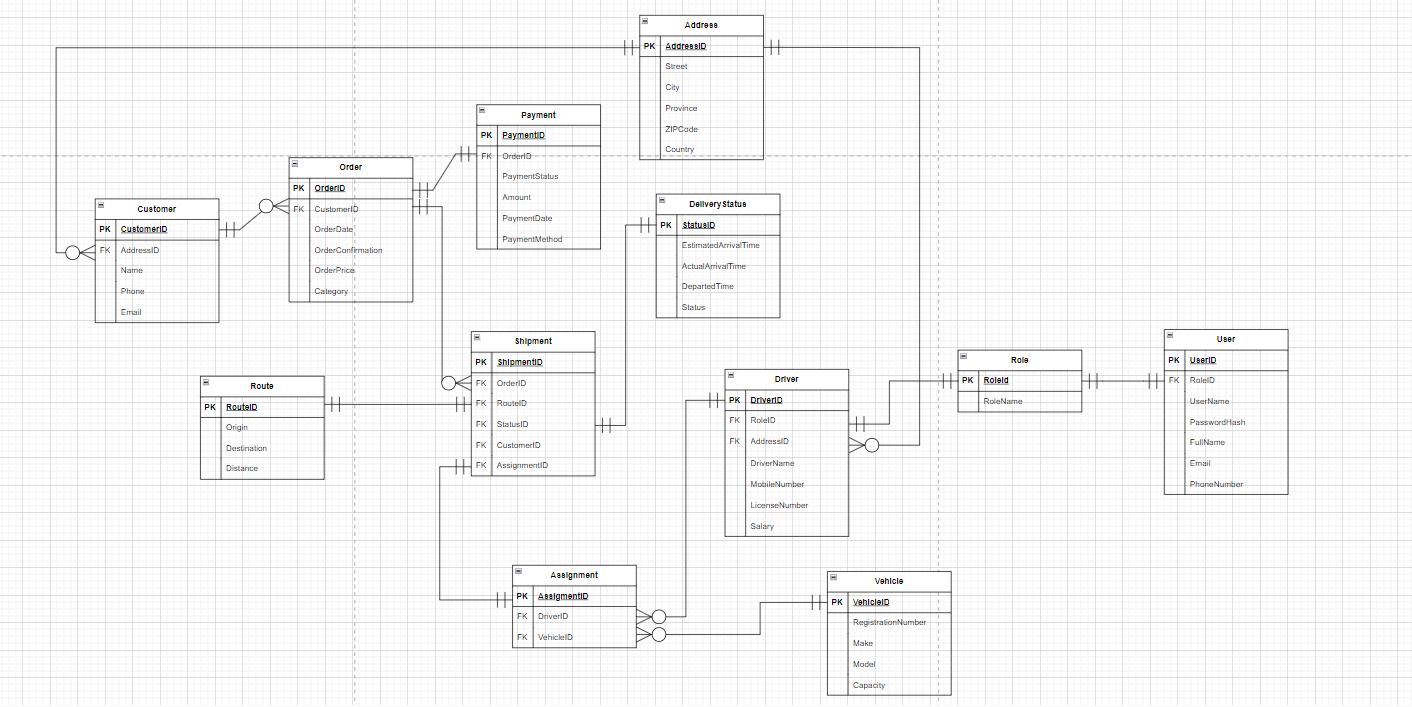
Description automatically generated**

**4. Authentication and User Roles**

* User Registration: Allow users to register and create their accounts.
* User Login: Enable users to authenticate and access the system.
* User Roles: Implement role-based access control to assign different permissions based on user roles (admin, manager, driver).

**5. Database Design**

**5.1 Entity – Relationship Diagram**



**5.2 Database Access**

* Utilize the Object-Relational Mapping (ORM) framework of SQLAlchemy to interact with the database.
* Implement appropriate data models and define relationships between the models.

**6. Implementation Plan:**

The development process for the logistics management application will follow the Agile  
methodology and it includes the following milestones:

Milestone 1: Setting up the development environment, including installing necessary software and configuring the project structure - July 09

Milestone 2: Implementing the frontend UI components and integrating them with the backend APIs for basic functionality – July 16

Milestone 3: Implementing user registration, authentication, driver assignment features - July 23

Milestone 4: Assigning a vehicle, adding the route, and tracking the delivery status of the shipment - July 30

Milestone 5: Conducting comprehensive testing, bug fixing, and performance optimization - August 06

Milestone 6: Final deployment and production readiness - August 13