Introduction

The Courier Management System (CMS) project aims to develop a GUI-based application using Object-Oriented Programming principles in Eclipse IDE, integrated with Oracle Database. This system addresses the need for efficient courier tracking and management within logistics companies.

Objectives

1. **Development Environment**: Utilize Eclipse IDE for Java development and Oracle Database for data storage.
2. **GUI Design**: Create an intuitive user interface for package tracking, delivery management, and administrative tasks.
3. **Database Integration**: Implement seamless connectivity with Oracle Database to ensure data consistency and reliability.

Java Classes

1. **User Management**

   - **Authentication**: Implement classes to manage user authentication using username and password validation.
   - **Roles and Permissions**: Define classes to assign roles (admin, manager, courier) and manage permissions for accessing system functionalities.

2. **Package Handling**

   - **Package Class**: Create a Java class representing a package with attributes such as tracking number, sender information, recipient information, and current status.
   - **Tracking**: Develop methods to update and retrieve package status (e.g., in transit, delivered) and location using real-time data.

3. **Delivery Management**

   - **Delivery Class**: Design a class to manage delivery schedules, including pickup and drop-off locations, delivery routes, and estimated time of arrival.
   - **Optimization**: Implement algorithms or heuristics to optimize delivery routes based on factors such as distance, traffic, and delivery priority.
   - **Proof of Delivery**: Develop functionality to capture proof of delivery (e.g., recipient's signature, timestamp) and update delivery status in the system.

Database Schema

1. Tables Design

- **Users Table:** Define columns for user ID, username, password hash, role ID, and permissions.
- **Packages Table**: Create columns for package ID, tracking number, sender details, recipient details, current status, and delivery details.
- **Delivery Logs Table**: Include fields for delivery ID, package ID, delivery status, delivery timestamp, and courier information.
- **Administrative Records Table**: Design columns for administrative actions such as user activity logs and system configuration changes.

Integration with GUI (Swing in JFrame, JTable, JLabel, JTextField)

1. Swing Components

- **JFrame**: Create a main application window (JFrame) to encapsulate the entire CMS application.
- **JTable**: Display data from the database, such as package lists or user information, in a tabular format for easy viewing and editing.
- **JLabel and JTextField**: Use JLabels to display descriptive text and JTextField to accept user input for actions like package tracking or user management.

2. GUI Design and Functionality

- **Event Handling**: Implement ActionListener interfaces to handle user interactions with buttons and text fields.
- **Layout Management**: Use layout managers (e.g., BorderLayout, GridLayout) to arrange Swing components within the JFrame effectively.
- **Data Binding**: Bind data retrieved from the database to Swing components (e.g., JTable) using custom TableModels or adapters for seamless data display and manipulation

JDBC Connectivity

1. JDBC

- **Connection Management:** Establish JDBC connections to Oracle Database using DriverManager and Connection objects.
- **SQL Queries:** Execute SQL queries (e.g., SELECT, INSERT, UPDATE) within Java methods to retrieve, modify, and store data in the database.

2. Error Handling

- **Exception Handling**: Use try-catch blocks to catch SQLExceptions and handle database connectivity issues gracefully

Conclusion

This detailed implementation plan covers the development of the Courier Management System using Java, Eclipse IDE, and Oracle Database. By focusing on Java class design, efficient database schema, seamless JDBC connectivity, and GUI design using Swing components, the system aims to provide a reliable, and user-friendly solution for courier tracking and management.