A PROJECT ON

**COURIER SERVICE MANAGEMENT SYSTEM**

SUBMITTED IN

PARTIAL FULFILLMENT OF THE REQUIREMENT

FOR THE COURSE OF DIPLOMA IN ADVANCED COMPUTING FROM CDAC



**SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY**

Hinjawadi

**SUBMITTED BY:**

Jujjavarapu Naveen

Vyas Juili Pradeep

Shruti Ashok Awate

Chaudhari Khushi Prashant

Prasad Ravi Rameshwar

**UNDER THE GUIDENCE OF:**

Mrs. Pooja Jaiswal

Faculty Member

Sunbeam Institute of Information Technology, Pune

**ACKNOWLEDGEMENT**

A project usually falls short of its expectation unless aided and guided by the right persons at the right time. We avail this opportunity to express our deep sense of gratitude towards Mr. Nitin Kudale (Center Coordinator, SIIT, Pune) and Mr. Yogesh Kolhe (Course Coordinator, SIIT, Pune).

We are deeply indebted and grateful to them for their guidance, encouragement and deep concern for our project. Without their critical evaluation and suggestions at every stage of the project, this project could never have reached its present form.

Last but not the least we thank the entire faculty and the staff members of Sunbeam Institute of Information Technology, Pune for their support.

Jujjavarapu Naveen

Vyas Juili Pradeep

Shruti Ashok Awate

Chaudhari Khushi Prashant

Prasad Ravi Rameshwar

PG-DAC

SIIT Pune

**A PROJECT ON**

**“COURIER SERVICE MANAGEMENT SYSTEM”**

SUBMITTED IN

PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE COURSE OF

DIPLOMA IN ADVANCED COMPUTING FROM CDAC



**SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY**

Hinjawadi

**SUBMITTED BY:**

Jujjavarapu Naveen

Vyas Juili Pradeep

Shruti Ashok Awate

Chaudhari Khushi Prashant

Prasad Ravi Rameshwar

**UNDER THE GUIDENCE OF:**

Mrs. Pooja Jaiswal Faculty Member

Sunbeam Institute of Information Technology, PUNE.



**CERTIFICATE**

This is to certify that the project work under the title ‘COURIER SERVICE MANAGEMENT SYSTEM’ is done by Naveen Jujjavarapu, Juili Vyas, Shruti Awate, Khushi Chaudhari, Ravi Prasad in partial fulfillment of the requirement for award of Diploma in Advanced Computing Course.

**Project Guide**

Date: 11-02-2025

**Mr. Yogesh Kolhe Course Co-Coordinator**

Table of Contents:

1. INTRODUCTION 2
2. FUNCTIONAL [REQUIREMENT 3](#_TOC_250005)
   1. [Admin Functionalities 3](#_TOC_250004)
   2. Customer Functionalities 5
   3. Warehouse Functionalities 7
   4. Delivery Agent Functionalities 8
3. [NON-FUNCTIONAL REQUIREMENTS 9](#_TOC_250003)
   * 1. Hardware and Software Interfaces 9
4. [DESIGN 11](#_TOC_250002)
   1. [Database design 11](#_TOC_250001)
5. CODING STANDARD IMPLEMENTED 15
6. [TEST REPORT 17](#_TOC_250000)
7. APENDIX A
8. Entity Relationship Diagram 19
9. Data Flow Diagram 20
10. Class Diagram 21
11. APENDIX B

UI Screenshots 22

1. REFERENCE 31

# INTRODUCTION TO PROJECT

The Courier Management System is a comprehensive web application designed to streamline the process of managing courier services. The system provides a user-friendly interface for Users, Delivery Agents, Warehouse Managers, and Admins to interact with the platform, manage shipments, track deliveries, and oversee warehouse operations.

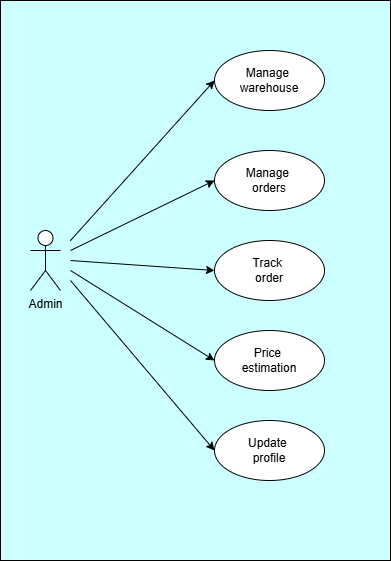
With the increasing demand for fast and efficient logistics services, businesses and individuals require a robust system to manage their shipments seamlessly. This system facilitates automation in courier operations, reducing manual work and improving overall efficiency. Users can easily schedule deliveries, track their packages in real-time.

Delivery agents benefit from real-time assignment of delivery tasks, allowing them to optimize their routes and update shipment statuses dynamically. Warehouse managers can efficiently handle inventory and shipments while ensuring that deliveries are timely and accurate. The system is equipped with comprehensive reporting tools to assist administrators in monitoring overall performance and making data-driven decisions.

# FUNCTIONAL REQUIREMENTS

## Admin Functionalities

The **Admin** is responsible for overseeing the entire system, View orders, tracking couriers, and maintaining warehouse and delivery operations.



**2.1.1 Manage Warehouses**

* Update warehouse details such as address and contact information.

**2.1.2 View Orders**

* View all courier orders, including pending, in transit, and delivered.

**2.1.3 Track Orders**

* Monitor order progress from **booking to delivery**.
* View real-time locations of parcels in transit.

**2.1.4 Price Estimation**

* Calculate price based on **weight, distance**.
* Provide estimated costs to customers before order placement.

**2.1.5 Update Profile**

* Modify admin profile details such as **name, and other details.**

## Customer Functionalities

* + 1. **Login/Register**

 Sign up using **email and password**.

 Authenticate using **JWT-based login**.

* + 1. **Place Orders**

 Enter **sender and receiver details** (name, address, phone number).

 Upload package details such as **weight**

* + 1. **Order History**

 View a list of past courier bookings.

 Filter orders by status (**Delivered, In Transit, Placed**).

* + 1. **Track Orders**
* Check real-time status of the order.
* Parcel travel warehouse to warehouse using **shortest distance path** algorithm.
  + 1. **Price Estimation**
* Estimate the shipping cost based on package weight and destination distance.
  + 1. **Update Profile**
* Modify customer profile details such as **name, and other details.**

**2.3 Warehouse Manager Functionalities**



**2.3.1 Manage Delivery Agents**

* View delivery agents.
* Assign delivery agents to specific orders.

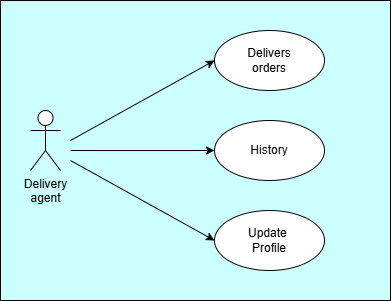
**2.3.2 Manage Deliveries**

* Accept parcels at the warehouse.
* Update order status.

**2.3.3 Update Profile**

* Modify warehouse manager profile details such as **name, and other details.**

**2.4 Delivery Agent Funtionalities**

****

**2.4.1 Deliver Orders**

 View assigned deliveries from the warehouse.

 Update system status when delivered.

**2.4.2 History**

* View delivery agent’s past deliveries.

**2.4.3 Update Profile**

* Modify delivery agent profile details such as **name, and other details.**

## Non-Functional Requirements

* 1. **Performance: The system should handle at least 1000 concurrent users.**
  2. **Scalability: The system should be scalable to accommodate future growth.**
  3. **Security: Ensure secure user authentication and data encryption.**
  4. **Availability: The system should have 99.9% uptime.**
  5. **User Interface: The interface should be intuitive and user-friendly.**

* 1. **Other Requirements**

**3.6.1 Hardware Interfaces**

**Requirements: Intel Core i5 or higher (or AMD equivalent),**

1. **GB RAM, 512 GB SSD or larger.**



## 3.6.2 Software Interfaces

* **Operating Systems**: MS Windows 13, Ubuntu 22.04.
* **Database**: MySQL.
* **Server**: Embedded Tomcat.
* **Browsers**: Compatible with modern web browsers.

## System Design

**3.6.4 Architecture**

* **Front-End**: Developed using React.js, providing an

intuitive UI for users.

* **Back-End**: Built using Spring Boot and Spring Security, handling all

processing and logic.

* **Database**: Uses MySQL for structured data storage and

retrieval.

* **Server**: Embedded Tomcat for hosting the application.

# DESIGN

## Database Design

The following table structures depict the database design.

**Table 1: Users**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key Type/Constraint** | **Column Name** | **Data Type** | **Length** | **Allow Null (1=Yes; 0=No)** |
| **3** | **id** | **BIGINT** | **-** | **0** |
| **0** | **first\_name** | **VARCHAR** | **50** | **0** |
| **0** | **last\_name** | **BIT(1)** | **50** | **0** |
| **0** | **password** | **VARCHAR** | **-** | **0** |
| **0** | **email** | **VARCHAR** | **40** | **0** |
| **0** | **role** | **ENUM** | **-** | **0** |
| **1** | **address\_id** | **BIGINT** |  | **1** |
| **0** | **contact\_number** | **VARCHAR** | **10** | **0** |

**Table 2: Addresses**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key Type/Constraint** | **Column Name** | **Data Type** | **Lengt h** | **Allow Null (1=Yes; 0=No)** |
| 3 | id | BIGINT | - | 0 |
| 0 | city | VARCHAR | 255 | 1 |
| 0 | flat\_no | VARCHAR | 255 | 1 |
| 0 | landmark | VARCHAR | 255 | 1 |
| 0 | pincode | int | - | 0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key Type/Constraint** | **Column Name** | **Data Type** | **Lengt h** | **Allow Null (1=Yes; 0=No)** |
| 0 | state | VARCHAR | 255 | 1 |
| 0 | street\_name | VARCHAR | 255 | 1 |

**Table 3: Orders**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key Type/Constraint** | **Column Name** | **Data Type** | **Lengt h** | **Allow Null (1=Yes; 0=No)** |
| 3 | id | BIGINT | - | 0 |
| 0 | contact\_number | VARCHAR | 10 | 1 |
| 0 | Delivery\_date | DATE | - | 1 |
| 0 | order\_time | DATE | - | 1 |
| 0 | price | DOUBLE | - | 0 |
| 0 | Receiver\_name | VARCHAR | 255 | 1 |
| 0 | status | ENUM | - | 1 |
| 0 | tracking\_id | VARCHAR | 255 | 1 |
| 0 | weight | DOUBLE | - | 0 |
| 1 | from\_warehouse\_id | BIGINT | - | 0 |
| 1 | delivery\_agent  \_id | BIGINT | - | 1 |
| 1 | To\_warehouse\_id | BIGINT | - | 1 |
| 1 | sender\_id | BIGINT | - | 0 |

**Table 4: Routes**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key Type/Constraint** | **Column Name** | **Data Type** | **Lengt h** | **Allow Null (1=Yes; 0=No)** |
| 3 | id | BIGINT | - | 0 |
| 0 | Arrival\_date | DATE | - | 1 |
| 0 | Dispatch\_date | DATE | - | 1 |
| 0 | status | ENUM | - | 1 |
| 1 | from\_wid | BIGINT | - | 0 |
| 1 | order\_id | BIGINT | 10 | 0 |
| 0 | to\_wid | BIGINT | - | 0 |

**Table 5: warehouse**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key Type/Constraint** | **Column Name** | **Data Type** | **Lengt h** | **Allow Null (1=Yes; 0=No)** |
| 3 | id | BIGINT | - | 0 |
| UNI | location\_id | BIGINT | - | 1 |
| UNI | manager\_id | BIGINT | - | 1 |

**Table 6: \_delivery\_agents**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key Type/Constraint** | **Column Name** | **Data Type** | **Lengt h** | **Allow Null (1=Yes; 0=No)** |
| **3** | **id** | **BIGINT** |  | **0** |
| **1** | **user\_id** | **BIGINT** | **-** | **1** |
| **1** | **Warehouse\_id** | **BIGINT** | **-** | **1** |

**E-R Diagram,Dataflow diagram and Class Diagram:**

Go to Appendix A

# CODING STANDARDS IMPLEMENTED

## Naming and Capitalization

Below summarizes the naming recommendations for identifiers in Pascal casing is used mainly (i.e. capitalize first letter of each word) with camel casing (capitalize each word except for the first one) being used in certain circumstances.

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | **Case** | **Examples** | **Additional Notes** |
| Class | Pascal | User, Order, UserController | Class names should be based on "objects" or "real things" and should generally be **nouns**. No ‘\_’ signs allowed. Do not use type  prefixes like ‘C’ for class. |

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Camel | SignUp, SignIn,  addReviews | Methods should use **verbs** or verb  phrases. |
| Parameter | Camel | firstName, lastName, email, password | Use descriptive parameter names. Parameter names should be descriptive enough that the name of the parameter and its type can be used to determine its meaning in  most scenarios. |
| Interface | Pascal with "I" prefix | UserRepository, OrderRepository  ,  MenuRepository | Do not use the ‘\_’ sign |
| Annotation | Pascal | SpringBootAppli  cation | Use @ at start of annotation |
| DTOs | Camel | ApiResponseDTO, SignUpReqDTO, OrderDetailsRes DTO | Use to transfer data between the processes |
| Exception Class | Pascal with "Exception"  suffix | ResourceNotFoun dException |  |

## Comments

* Comment each type, each non-public type member, and each region declaration.
* Use end-line comments only on variable declaration lines.

End-line comments are

comments that follow code on a single line.

* Separate comments from comment delimiters (apostrophe) or // with one space.
* Begin the comment text with an uppercase letter.
* End the comment with a period.
* Explain the code; do not repeat it.

# TEST REPORT

**Another group called Linux did the testing and the report of the testing is given hereunder.**

# GENERAL TESTING:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SR-NO** | **TEST CASE** | **EXPECTED RESULT** | **ACTUAL RESULT** | **ERROR MESSAGE** |
| 1 | Register Page | Register  successfully message | OK | Nothing |
| 2 | Login Page | Pop-up will come | Ok | Invalid Username or password |
| 3 | Place-order by customer | Add order to DB | Ok | Failed to add order |
| 4 | Price estimation | Calculated price will be displayed | Ok | Nothing |
| 5 | Tracking | Give a status of particular  order | Ok | Nothing |

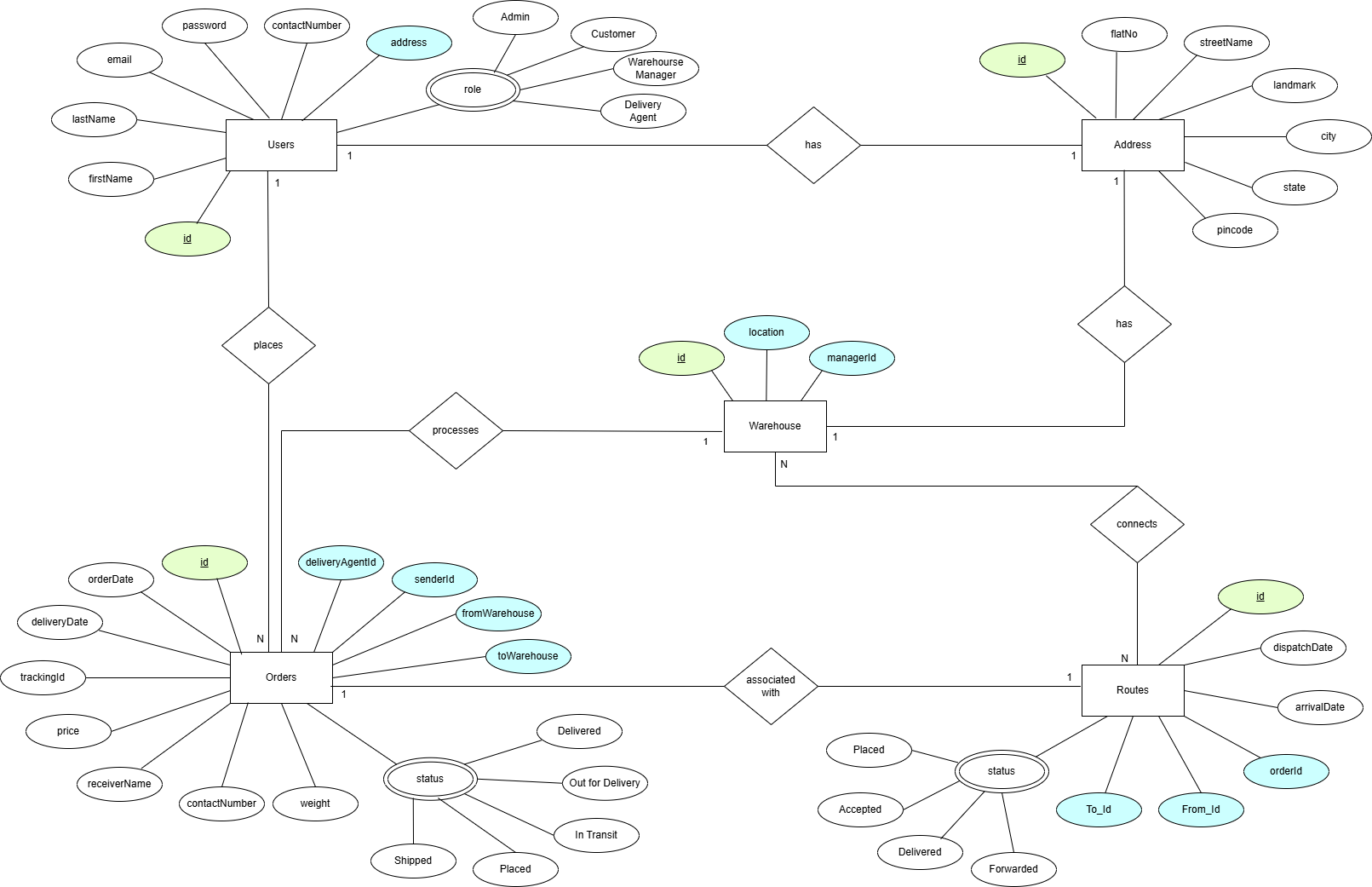
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **6** | **Customer Order history** | **Order history list render successfully** | **Ok** | **Failed to fetched orders** |
| **7** | **Warehouse manage-deliveries** | **View,accept,forward, assign delivery agent** | **Ok** | **Nothing** |
| **8** | **Warehouse manage delivery agent** | **View and add delivery agent** | **ok** | **Nothing** |
| **9** | **Warehouse profile** | **Update the profile** | **Ok** | **Error fetching profile** |
| **10** | **Generate Token after first**  **login** | **Token generated successfully** | **Ok** | **Failed to generate token** |
| **11** | **View all**  **orders list by admin** | **Seeing the list of all orders** | **Ok** | **Nothing** |
| **11** | **View all warehouse list by admin** | **Seeing the list of all warehouse and update it** | **Ok** | **Nothing** |
| **12** | **View all deliveries assign to delivery agent** | **Seeing the list of all deliveries and deliver it.** | **ok** | **Fail to fetch the deliveries** |
| **13** | **View all deliveries previously assigned (History)** | **Seeing the list of all previous delivered orders** | **ok** | **Nothing** |
| **12** | **Logout** | **It will logout from user profile.** | **Ok** | **Nothing** |

--

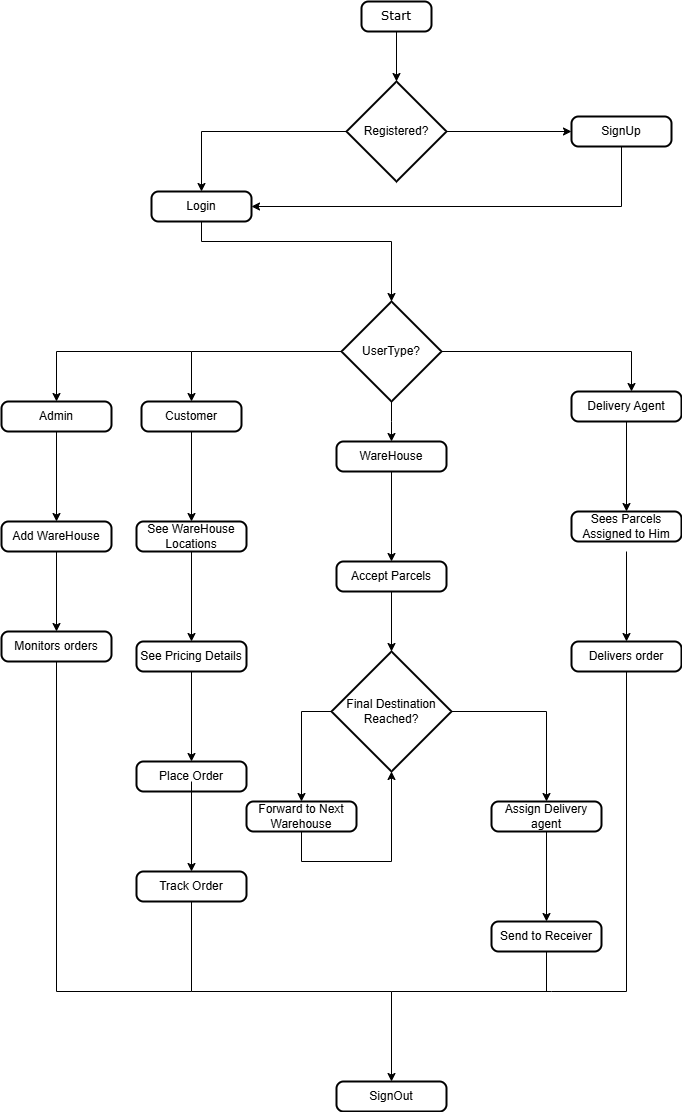
-

1. **Appendix A**

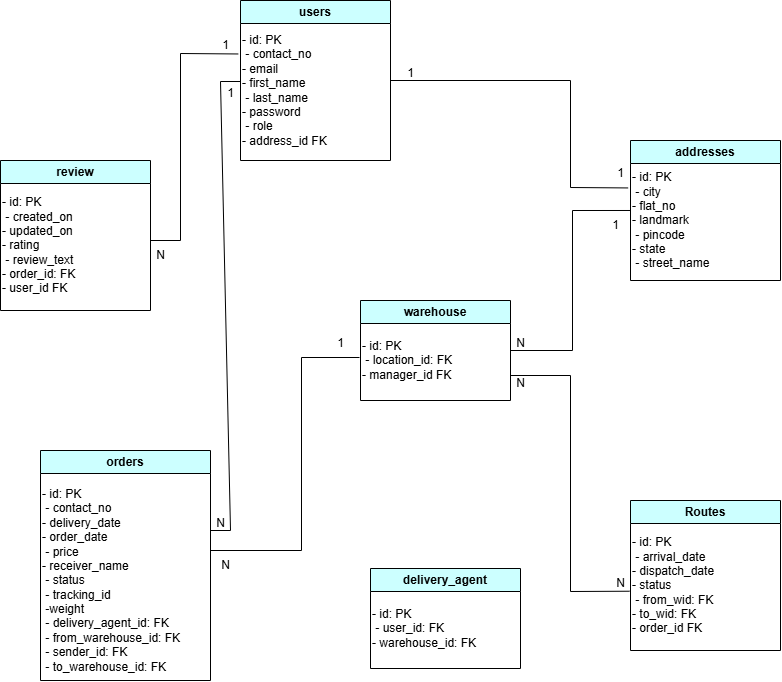
**Entity Relationship Diagram:**



**Data Flow Diagram:**



**Class Diagram:**

****

1. **Appendix B   
     
   Customer pages:**

**Login:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Registration:**

**A screenshot of a login form

AI-generated content may be incorrect.**

**Home Page:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Place Order:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Price Estimation:**

A screenshot of a computer

AI-generated content may be incorrect.

**History:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Update Profile:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Admin Pages:**

**Orders:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Warehouse:**

**A screenshot of a warehouse logistic

AI-generated content may be incorrect.**

**Track Order**

**A screenshot of a login page

AI-generated content may be incorrect.**

**Price Estimation**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Profile:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Warehouse Pages:**

**Delivery Agent:**

**A screenshot of a delivery agent

AI-generated content may be incorrect.**

**Manage Deliveries:**

**A screenshot of a login

AI-generated content may be incorrect.**

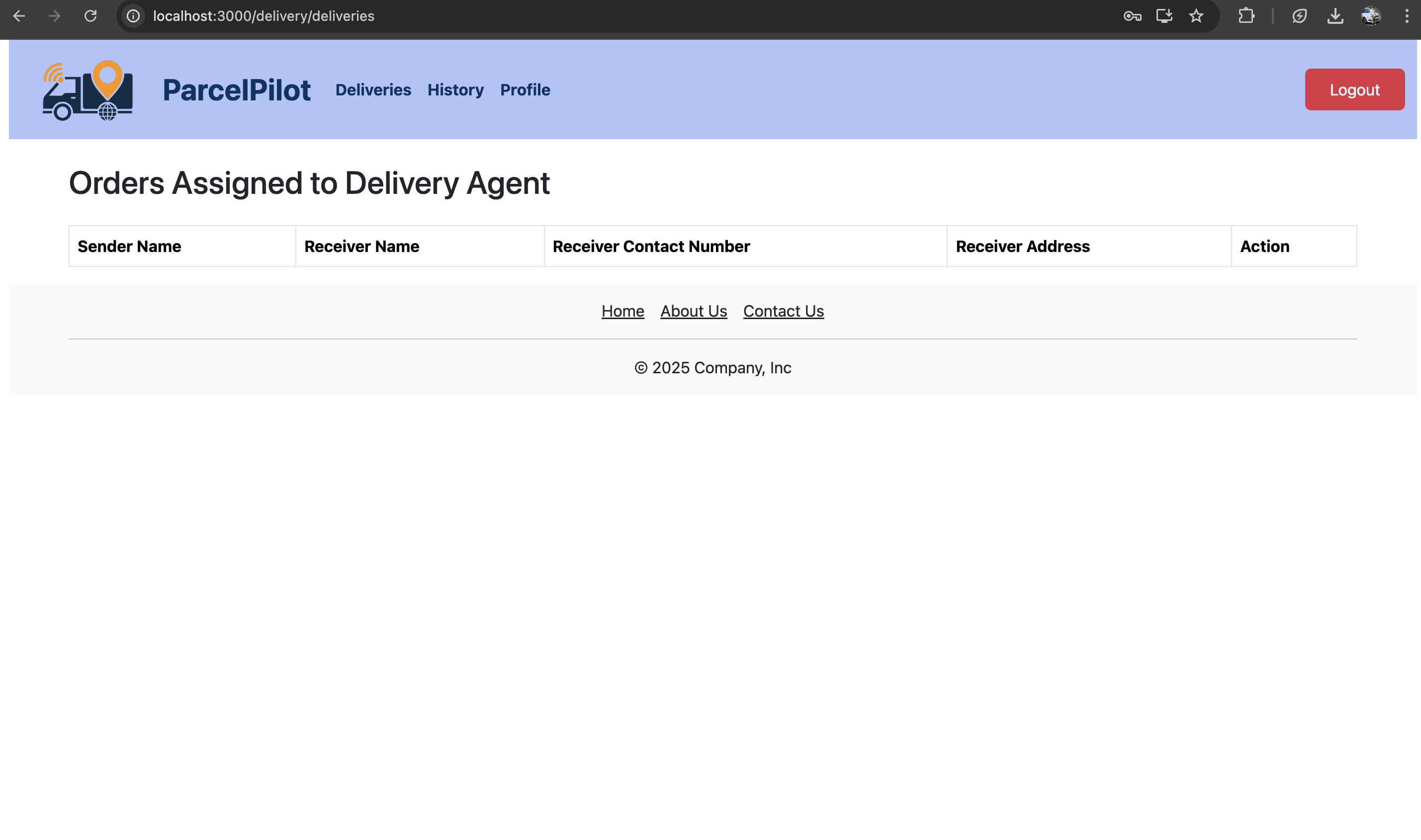
**Profile:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Delivery Agent Pages:**

**Deliveries:**

d

**History:**

A screenshot of a computer

AI-generated content may be incorrect.om

**Profile:**

**A screenshot of a computer

AI-generated content may be incorrect.**

# REFERENCES

## Spring Boot Documentation

URL: <https://spring.io/projects/spring-boot>

## React.js Documentation

URL: https://reactjs.org/docs/getting-started.html

## Redux Documentation

URL: https://redux.js.org

## Java Programming Language

URL: <https://www.oracle.com/java/>

## MySQL Workbench Documentation

URL: <https://dev.mysql.com/doc/workbench/en/>

## Spring Boot with React and Redux

URL: https://[www.baeldung.com/spring-boot-react-and-redux](http://www.baeldung.com/spring-boot-react-and-redux)

## Java Persistence API (JPA) Documentation

URL: https://[www.eclipse.org/eclipselink/documentation/2.7/](http://www.eclipse.org/eclipselink/documentation/2.7/)

## Swagger Documentation for Spring Boot

URL: <https://springdoc.org/>

## MDN Web Docs

URL: <https://developer.mozilla.org/>

## React Redux Integration Guide

URL: https://react-redux.js.org/