COURIER

**DATABASE MANAGEMENT SYSTEM**

Name : Gauri Takarkhede

MIS : 112103144

Division : 2

Batch : T3

Subject : Database Management System

**1. Problem Statement:**

In the rapidly advancing modern age, the need for efficient and timely courier services has become essential. Existing methods, such as post services and international shipping, often fall short in terms of time consumption, damage, and missing items. To address these issues, the proposed Courier Management System aims to streamline the process of sending and receiving packages through a network of branches, providing a quick and reliable solution for customers.

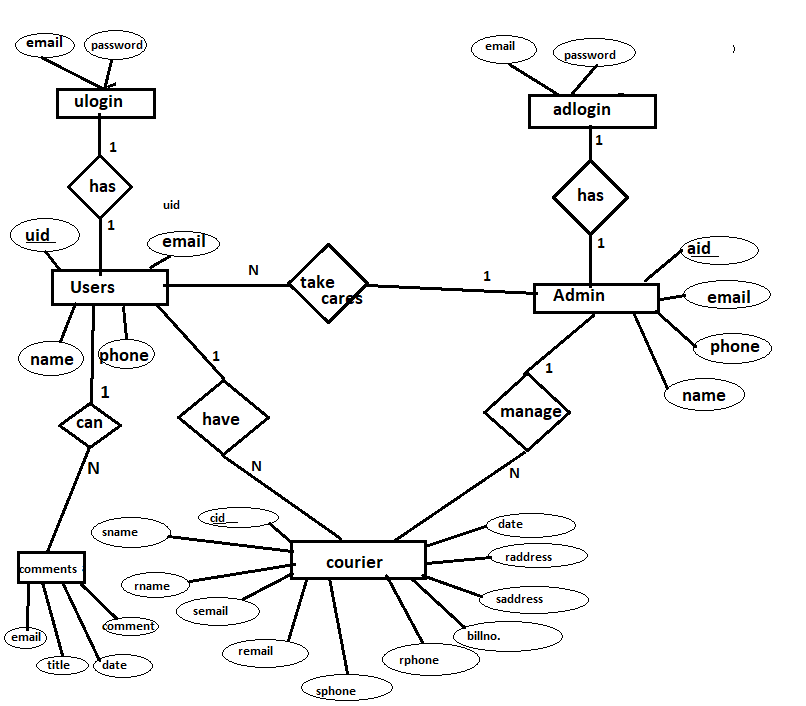
**2. Objectives:**

* Develop a Courier Management System to facilitate seamless order placement and tracking.
* Implement a user-friendly login system for both administrators and users.
* Capture and store detailed customer information and recipient details during order placement.
* Generate a unique tracking ID during the billing process for efficient package tracking.
* Provide customers and recipients with real-time status updates on their packages.
* Allow administrators to manipulate data through an admin login page and add new consignments.
* Display user profiles and pricing information based on weight categories.

**3. Functional Requirements:**

* User and Admin login functionality.
* Order processing with customer and recipient details.
* Billing process with automatic tracking ID generation.
* Real-time package tracking for customers and recipients.
* Admin capabilities to manipulate data and add new consignments.
* User profile display.
* Pricing information based on weight categories.

**4. ER Diagram:**



**5. Relational Schemas Obtained from ER:**

* User(UserID, Username, Password, UserType)
* Admin(AdminID, AdminName, AdminEmail, AdminPhone)
* Customer(CustomerID, UserID (FK), CustomerName, CustomerAddress, CustomerPhone)
* Recipient(RecipientID, RecipientName, RecipientAddress, RecipientPhone)
* Order(OrderID, CustomerID (FK), RecipientID (FK), TrackingID, OrderDate)
* Branch(BranchID, BranchName, BranchLocation)
* Schedule(ScheduleID, SourceBranchID (FK), DestinationBranchID (FK), DepartureTime, ArrivalTime)

**6. Set of Functional Dependencies:**

* User:

UserID → Username, Password, UserType

* Admin:

AdminID → AdminName, AdminEmail, AdminPhone

* Customer:

CustomerID → UserID, CustomerName, CustomerAddress, CustomerPhone

* Recipient:

RecipientID → RecipientName, RecipientAddress, RecipientPhone

* Order:

OrderID → CustomerID, RecipientID, TrackingID, OrderDate

* Branch:

BranchID → BranchName, BranchLocation

* Schedule:

ScheduleID → SourceBranchID, DestinationBranchID, DepartureTime,

ArrivalTime

**7. Normalization (Up to 3NF):**

* User:

1NF: UserID, Username, Password, UserType

2NF: (No partial dependencies)

3NF: (No transitive dependencies)

* Admin:

1NF: AdminID, AdminName, AdminEmail, AdminPhone

2NF: (No partial dependencies)

3NF: (No transitive dependencies)

* Customer:

1NF: CustomerID, UserID, CustomerName, CustomerAddress, CustomerPhone

2NF: (No partial dependencies)

3NF: (No transitive dependencies)

* Recipient:

1NF: RecipientID, RecipientName, RecipientAddress, RecipientPhone

2NF: (No partial dependencies)

3NF: (No transitive dependencies)

* Order:

1NF: OrderID, CustomerID, RecipientID, TrackingID, OrderDate

2NF: (No partial dependencies)

3NF: (No transitive dependencies)

* Branch:

1NF: BranchID, BranchName, BranchLocation

2NF: (No partial dependencies)

3NF: (No transitive dependencies)

* Schedule:

1NF: ScheduleID, SourceBranchID, DestinationBranchID, DepartureTime, ArrivalTime

2NF: (No partial dependencies)

3NF: (No transitive dependencies)