1. Introduction

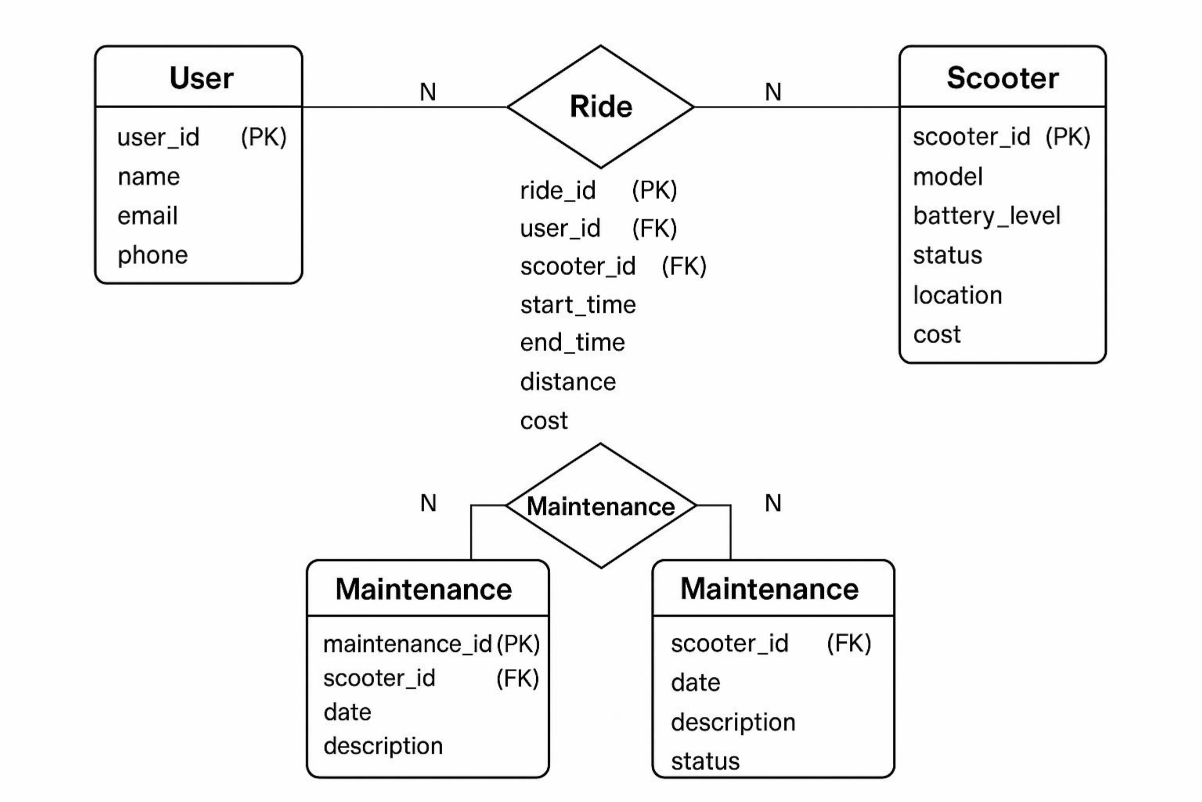
1.1 Project Description

The Scooter Tracking System project enables the management and monitoring of an electric scooter fleet through a simple web application. Users can add, list, and manage scooters, users, ride records, and maintenance logs. The system is designed for practical fleet operations, easy access to ride statistics, and efficient database management.

2. Entity Relational Model

2.1 Enhanced ER Diagram

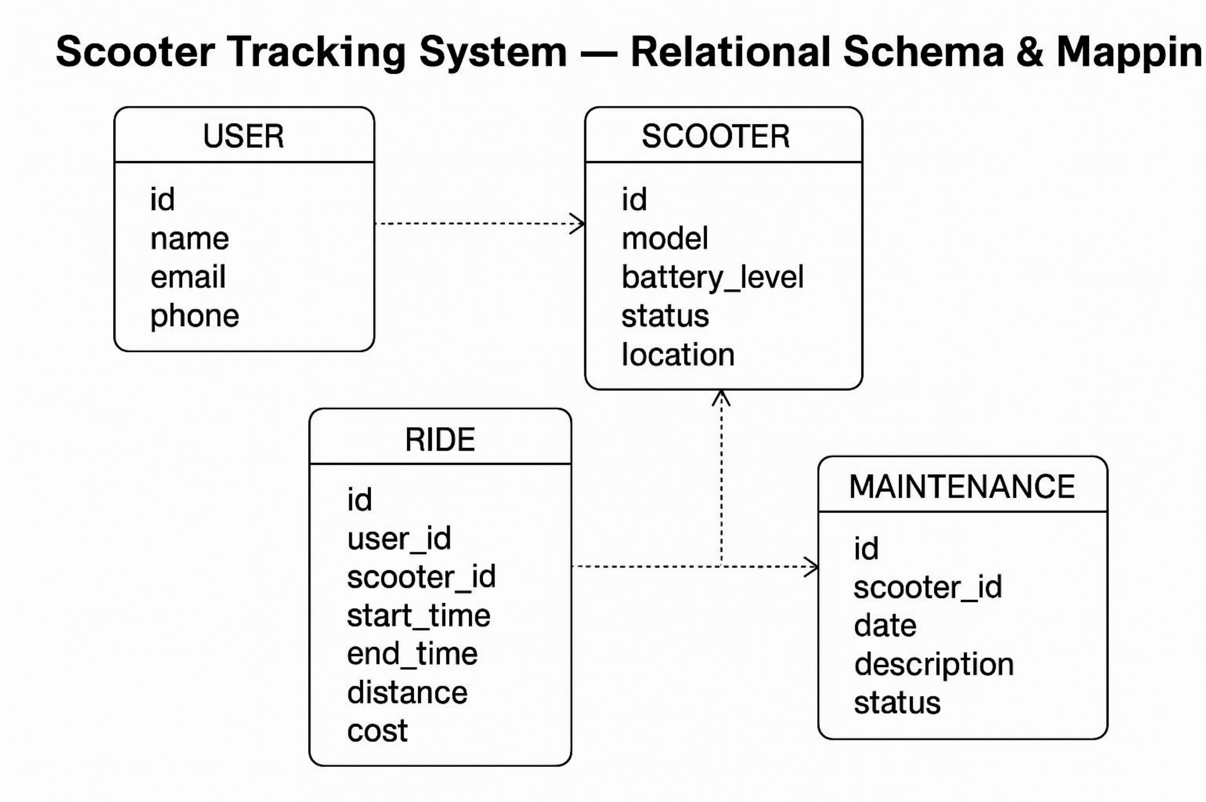
Below is the Enhanced Entity-Relationship Diagram (EER) developed for the Scooter Tracking System:**Figure 2.1: Scooter Tracking System Enhanced ER Diagram**



2.2 Relational Schema & Mapping

The relational schema and mapping for the Scooter Tracking System are presented below:

Figure 2.2: Relational Schema (Scooter Tracking System)



3.NORMALIZATION:

3.1 Functional Dependencies

User

- `id → name, email, phone`

Scooter

- id → model, battery_level, status, location

Ride

- id → user_id, scooter_id, start_time, end_time, distance, cost
- user_id → User FK
- scooter_id → Scooter FK

Maintenance

- id → scooter_id, date, description, status
- scooter_id → Scooter FK

F = { FD1: user_id → name, email, phone

FD2: scooter_id → model, battery_level, status, location

FD3: ride_id → user_id, scooter_id, start_time, end_time, distance, cost FD4: maintenance_id → scooter_id, date, description, status }

3.2 Normal Form Transformations

1. **1NF**: All data values are atomic.

| user_id | name | email | phone | | scooter_id | model | battery_level | status | location | | ride_id | user_id | scooter_id | start_time | end_time | distance | cost | | maintenance_id | scooter_id | date | description | status |

2. **2NF**: Partial functional dependencies have been removed.

   ride_id → user_id, scooter_id, ...

3. **3NF**: Transitive dependencies have been removed.

- User : user_id → name, email, phone
- Scooter : scooter_id → model, battery_level, status, location
- Ride : ride_id → user_id, scooter_id, ...
- Maintenance : maintenance_id → scooter_id, ...

4. Application

4.1 Technology Selection

- **Backend:** Python (Flask framework)
- **Database:** SQLite (local), managed with SQLAlchemy ORM
- **Frontend:** HTML (Jinja2 templates), basic CSS
- **Features:** Web-based CRUD interface for scooters, users, rides, and maintenance
- **Usage:** Accessed via web browser on local machine
- **Goal:** To provide a practical, secure, and centralized management solution for electric scooter operations