|  |
| --- |
| Database Management project |
| Electric Scooter Rental System |
| 2025 |

|  |
| --- |
| Tarik Šuta (tsuta@student.ius.edu.ba) |

Contents

[1. Introduction 2](#_Toc194944923)

[2. Requirement Analysis 3](#_Toc194944924)

[2.1 Stakeholder Requirements 3](#_Toc194944925)

[2.1.1 Customers 3](#_Toc194944926)

[2.1.2 Maintenance Staff 3](#_Toc194944927)

[2.1.3 Operations Managers 3](#_Toc194944928)

[2.1.4 Payment Processors 3](#_Toc194944929)

[2.2 Data to be stored 4](#_Toc194944930)

[2.3 Operations to be performed 4](#_Toc194944931)

[2.4 Users of the system 4](#_Toc194944932)

[3. Database Design 5](#_Toc194944933)

[3.1 Conceptual Design 5](#_Toc194944934)

[3.2 Logical Design 5](#_Toc194944935)

[3.3 Normalization 9](#_Toc194944936)

[4. Conclusion 9](#_Toc194944937)

# 1. Introduction

The **Electric Scooter Rental System** is a database management system (DBMS) designed to streamline the operations of an urban mobility service that provides shared electric scooters to customers. As cities embrace eco-friendly transportation alternatives, efficient management of scooter inventory, user rentals, payments, and maintenance becomes crucial. This project aims to create a robust, scalable, and user-friendly database solution to support these operations.

**Purpose of the DBMS**

The primary purpose of this DBMS is to:

* **Track scooter availability** in real-time across multiple rental stations.
* **Manage user accounts**, including registrations, rentals, and payment processing.
* **Record maintenance history** to ensure scooters remain in optimal condition.
* **Generate reports** for business analytics (e.g., peak rental times, revenue trends).

By centralizing this data, the system enhances operational efficiency, improves user experience, and reduces manual errors associated with traditional record-keeping.

**Entities and Relationships**

The system’s core entities include:

* **Users** (customers who rent scooters)
* **Scooters** (individual vehicles with status, location, and battery levels)
* **Stations** (designated pickup/drop-off points)
* **Rentals** (transactions linking users and scooters)
* **Payments** (billing records tied to rentals)
* **Maintenance Records** (repair logs managed by employees)

**Key relationships**:

* Users *rent* Scooters (one-to-many).
* Scooters *are located at* Stations (many-to-one).
* Maintenance staff *service* Scooters (one-to-many).

This structured approach ensures data consistency, supports business growth, and adapts to evolving urban mobility demands.

# 2. Requirement Analysis

The requirement analysis phase establishes the foundation for the Electric Scooter Rental System by identifying and documenting the essential needs of all stakeholders, core system functionalities, and technical specifications. This analysis serves as a critical blueprint that guides the database design and application development process. By systematically examining user expectations, operational demands, and data management needs, we ensure the final system will deliver a seamless rental experience for customers while providing efficient management tools for staff and administrators.

## 2.1 Stakeholder Requirements

Stakeholder’s included in this project are:

* Sponsors of the project: companies and manufacturers of scooters and replacement batteries.
* Employees: mechanics and workers who supervise the scooters.
* Director and managers.
* Customers

### 2.1.1 Customers

* Rent scooters quickly via an app/website.
* View real-time scooter availability and locations.
* Securely pay for rentals.
* Report scooter issues.

### 2.1.2 Maintenance Staff

* Track scooter condition and battery levels.
* Receive automated alerts for damaged scooters.
* Log repairs and maintenance history.

### 2.1.3 Operations Managers

* Monitor rental trends and revenue
* Optimize scooter distribution across stations
* Generate reports on usage and profitability

### 2.1.4 Payment Processors

* Handle secure transactions (credit/debit, digital wallets).
* Issue refunds if needed.
* Prevent fraudulent activities.

## 2.2 Data to be stored

|  |  |
| --- | --- |
| Entity | Attributes |
| Users | User ID, Name, Email, Phone, Payment Method, Registration Date, Account Status |
| Scooters | Scooter ID, Model, Battery Level, Status (Available/Rented/Maintenance), Current Station |
| Stations | Station ID, Location (GPS), Address, Capacity, Available Scooters Count |
| Rentals | Rental ID, User ID, Scooter ID, Start/End Time, Start/End Station, Distance, Cost |
| Payments | Payment ID, Rental ID, Amount, Payment Method, Status (Completed/Failed/Refunded) |
| Maintenance | Maintenance ID, Scooter ID, Issue Description, Repair Status, Cost, Employee Assigned |

## 2.3 Operations to be performed

|  |  |
| --- | --- |
| Operation | Description |
| User Registration | Allow new users to sign up with personal and payment details. |
| **Scooter Search & Booking** | Users can locate and reserve available scooters via GPS. |
| **Rental Management** | Track rental start/end times, calculate costs, and update scooter status. |
| **Payment Processing** | Handle transactions, receipts, and refunds securely. |
| **Maintenance Tracking** | Log repairs, schedule servicing, and flag faulty scooters. |
| **Reporting & Analytics** | Generate insights on peak usage, revenue, and scooter performance. |

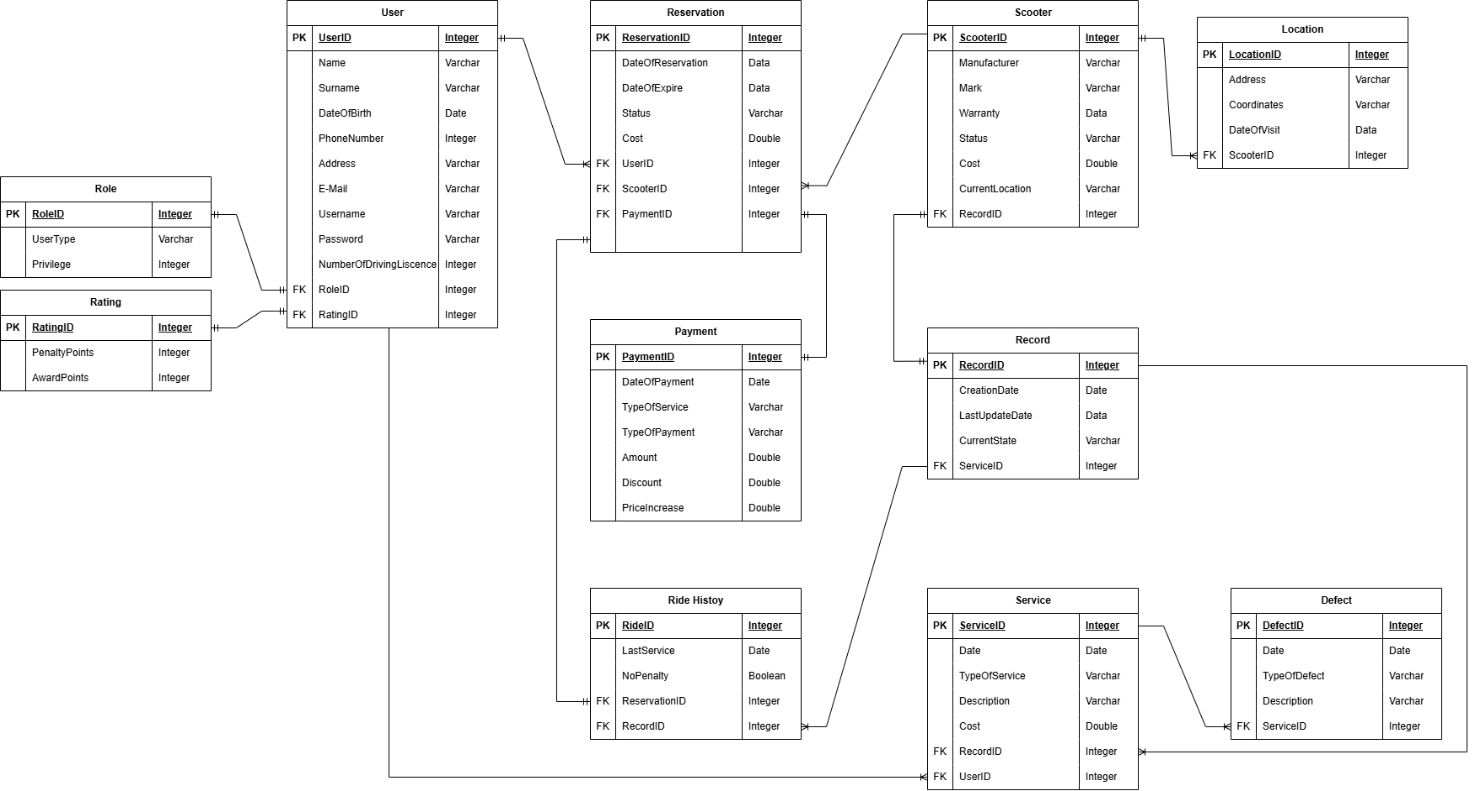
## 2.4 Users of the system

**UrbanMove** defines several types of users within this information system. Each actor has limited capabilities depending on their role in the system. The descriptions of the different types of actors present in this IS are as follows:

* **Mechanic(s)** – This type of actor is the key link in the information system chain. The system allows the mechanic to locate the positions of all scooters, inspect their condition, and record the findings in the IS. In case of a malfunction, after performing diagnostics, the mechanic reports the issue and updates the scooter’s file. This ensures the regular maintenance of the scooters.
* **Driver(s)** – This type of user can rent and use scooters via a mobile application integrated with the information system. During the rental process, the system requires specific information from the driver to grant access to the desired services. While using the system, registered drivers are monitored, and based on their usage, they can be rewarded or penalized by the service provider.
* **Company Employee(s)** – This type of actor uses the information system to perform various tasks such as customer analysis, notifying customers about malfunctions, penalizing and rewarding drivers, displaying scooter status and money spent, and reviewing different types of reports.

# 3. Database Design

## 3.1 Conceptual Design



## 3.2 Logical Design

The ER model transformed into relational tables, where each entity and relationship is represented with appropriate attributes, primary keys (PK), and foreign keys (FK). The logical design ensures that the database structure is normalized and ready for implementation.

User(

UserID INT PRIMARY KEY,

Name VARCHAR,

Surname VARCHAR,

DateOfBirth DATE,

PhoneNumber VARCHAR,

Address VARCHAR,

Email VARCHAR,

Username VARCHAR,

Password VARCHAR,

NumberOfDrivingLicense INT,

RoleID INT,

RatingID INT

)

Role(

RoleID INT PRIMARY KEY,

UserType VARCHAR,

Privilege INT

)

Rating(

RatingID INT PRIMARY KEY,

PenaltyPoints INT,

AwardPoints INT

)

Reservation(

ReservationID INT PRIMARY KEY,

ReservationDate DATE,

ExpirationDate DATE,

Status VARCHAR,

Cost DOUBLE,

UserID INT,

ScooterID INT,

PaymentID INT

)

Scooter(

ScooterID INT PRIMARY KEY,

Manufacturer VARCHAR,

Mark VARCHAR,

Warranty DATE,

Status VARCHAR,

Cost DOUBLE,

CurrentLocation VARCHAR,

RecordID INT

)

Location(

LocationID INT PRIMARY KEY,

Address VARCHAR,

Coordinates VARCHAR,

DateOfVisit DATE,

ScooterID INT

)

Payment(

PaymentID INT PRIMARY KEY,

DateOfPayment DATE,

TypeOfService VARCHAR,

TypeOfPayment VARCHAR,

Amount DOUBLE,

Discount DOUBLE,

PriceIncrease DOUBLE

)

Record(

RecordID INT PRIMARY KEY,

CreatedAt DATE,

UpdatedAt DATE,

CurrentState VARCHAR,

ServiceID INT

)

RideHistory(

RideID INT PRIMARY KEY,

LastService DATE,

NoPenalty BOOLEAN,

ReservationID INT,

RecordID INT

)

Service(

ServiceID INT PRIMARY KEY,

Date DATE,

TypeOfService VARCHAR,

Description VARCHAR,

Cost DOUBLE,

RecordID INT,

UserID INT

)

Defect(

DefectID INT PRIMARY KEY,

Date DATE,

TypeOfDefect VARCHAR,

Description VARCHAR,

ServiceID INT

)

## 3.3 Normalization

The database design has been normalized to the Third Normal Form (3NF) to ensure data integrity, reduce redundancy, and improve efficiency. Each table contains atomic attributes, and there is a clear separation of concerns with properly defined relationships. All non-key attributes are fully functionally dependent on the primary key, and transitive dependencies have been eliminated by creating separate related tables (e.g., Role, Rating, Payment, Service). This normalization approach enhances data consistency, simplifies queries, and supports scalability.

# 4. Conclusion

In conclusion, this project involved the design and implementation of a relational database system for an urban scooter rental platform. Starting from an ER model, we developed a normalized logical design that ensures data consistency, integrity, and scalability. The database supports essential operations such as user management, reservations, payments, scooter tracking, and maintenance records. The complete SQL implementation of the database can be found in the attached file urbanmove.sql. Overall, this system lays a solid foundation for building a reliable and efficient application for managing urban mobility services.