Smart City Parking Management System Report

Team Members

عمر هانى ابراهيم محمد بشر 21010891

محمد حسن محمد قبارى حسن عبد الجواد 21011115

محمد محمد ابراهيم على حسن 21011211

ساهر طارق أنور زايد أحمد 19015763

Application Source Code

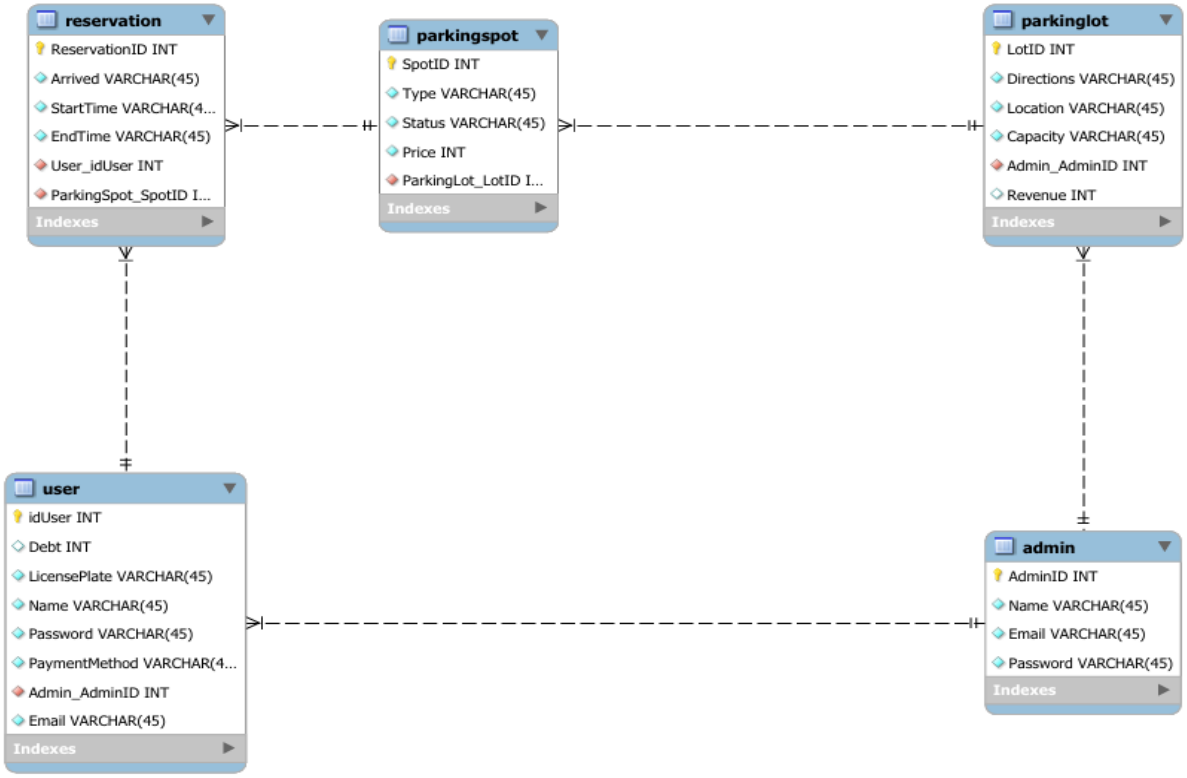
project repo link

1. Introduction

The Smart City Parking Management System aims to provide efficient parking solutions within urban environments by leveraging modern technology. The system's primary goals include reducing parking search time, optimizing space utilization, and enhancing the user experience.

2. Database Design

ERD Diagram



Database Schema

The database schema consists of six key tables:

1. **Admin**
    a. Manages administrative users.
    b. Key attributes: AdminID, Name, Email, Password.
2. **User**
    a. Stores user details.
    b. Key attributes: idUser, Name, Email, Password, LicensePlate, Debt, PaymentMethod, Admin_AdminID.
3. **ParkingLot**
    a. Represents parking lots managed by the system.
    b. Key attributes: LotID, Directions, Location, Capacity, Revenue, Admin_AdminID.
4. **ParkingSpot**
    a. Contains information about individual parking spots.
    b. Key attributes: SpotID, Type, Status, Price, ParkingLot_LotID.

5. **Reservation**
    a. Tracks user reservations.
    b. Key attributes: ReservationID, StartTime, EndTime, Arrived, User_idUser, ParkingSpot_SpotID.

- **Admin** has a one-to-many relationship with **ParkingLot**, indicating that each parking lot is managed by one admin.
- **User** has a one-to-many relationship with **Reservation**, where each user can have multiple reservations.
- **ParkingSpot** has a many-to-one relationship with **ParkingLot**, indicating that each parking spot belongs to one parking lot.
- **Reservation** has a many-to-one relationship with **User** and **ParkingSpot**, representing that each reservation is made by one user for one parking spot.

SQL Generation Script

A comprehensive SQL script to create the database schema, along with the insertion of sample data, has been prepared. The script ensures:

- Proper indexing for query optimization.
- Referential integrity through foreign key constraints.
- Sample data for Admins, Users, Parking Lots, and Reservations.

-- MySQL Script generated by MySQL Workbench -- Sat Dec 28 07:41:22 2024 -- Model: New Model Version: 1.0 -- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0; SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0; SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

-- Schema smartparkingcity

-- Schema smartparkingcity

CREATE SCHEMA IF NOT EXISTS smartparkingcity DEFAULT CHARACTER SET utf8mb3 ;
USE smartparkingcity ;

-- Table smartparkingcity.admin

CREATE TABLE IF NOT EXISTS smartparkingcity.admin (AdminID INT NOT NULL AUTO_INCREMENT, Name VARCHAR(45) NOT NULL DEFAULT '', Email VARCHAR(45) NOT NULL DEFAULT '', Password VARCHAR(45) NOT NULL DEFAULT '', PRIMARY KEY (AdminID), INDEX idx_admin_signin (Email ASC, Password ASC) VISIBLE) ENGINE = InnoDB AUTO_INCREMENT = 5 DEFAULT CHARACTER SET = utf8mb3;

-- Table smartparkingcity.parkinglot

CREATE TABLE IF NOT EXISTS smartparkingcity.parkinglot (LotID INT NOT NULL AUTO_INCREMENT, Directions VARCHAR(45) NOT NULL DEFAULT '', Location VARCHAR(45) NOT NULL DEFAULT '', Capacity VARCHAR(45) NOT NULL DEFAULT '', Admin_AdminID INT NOT NULL DEFAULT '1', Revenue INT NULL DEFAULT '0', PRIMARY KEY (LotID), INDEX fk_ParkingLot_Admin1_idx (Admin_AdminID ASC) VISIBLE, INDEX idx_parking_lot_id (LotID ASC) VISIBLE, CONSTRAINT fk_ParkingLot_Admin1 FOREIGN KEY (Admin_AdminID) REFERENCES smartparkingcity.admin (AdminID)) ENGINE = InnoDB AUTO_INCREMENT = 6 DEFAULT CHARACTER SET = utf8mb3;

-- Table smartparkingcity.parkingspot

CREATE TABLE IF NOT EXISTS smartparkingcity.parkingspot (SpotID INT NOT NULL AUTO_INCREMENT, Type VARCHAR(45) NOT NULL DEFAULT 'Standard', Status

VARCHAR(45) NOT NULL DEFAULT 'Available', `Price` INT NOT NULL DEFAULT '0', `ParkingLot_LotID` INT NOT NULL DEFAULT '1', PRIMARY KEY (`SpotID`), INDEX `fk_ParkingSpot_ParkingLot1_idx` (`ParkingLot_LotID` ASC) VISIBLE, INDEX `idx_parking_spot_id` (`SpotID` ASC) VISIBLE, CONSTRAINT `fk_ParkingSpot_ParkingLot1` FOREIGN KEY (`ParkingLot_LotID`) REFERENCES `smartparkingcity`.`parkinglot` (`LotID`)) ENGINE = InnoDB AUTO_INCREMENT = 15 DEFAULT CHARACTER SET = utf8mb3;

-- Table `smartparkingcity`.`user`

CREATE TABLE IF NOT EXISTS `smartparkingcity`.`user` (`idUser` INT NOT NULL AUTO_INCREMENT, `Debt` INT NULL DEFAULT '0', `LicensePlate` VARCHAR(45) NOT NULL DEFAULT '', `Name` VARCHAR(45) NOT NULL DEFAULT '', `Password` VARCHAR(45) NOT NULL DEFAULT '', `PaymentMethod` VARCHAR(45) NOT NULL DEFAULT 'Credit Card', `Admin_AdminID` INT NOT NULL DEFAULT '1', `Email` VARCHAR(45) NOT NULL DEFAULT '', PRIMARY KEY (`idUser`), UNIQUE INDEX `idx_user_email` (`Email` ASC) VISIBLE, INDEX `fk_User_Admin_idx` (`Admin_AdminID` ASC) VISIBLE, INDEX `idx_user_signin` (`Email` ASC, `Password` ASC) VISIBLE, INDEX `idx_user_id` (`idUser` ASC) VISIBLE, CONSTRAINT `fk_User_Admin` FOREIGN KEY (`Admin_AdminID`) REFERENCES `smartparkingcity`.`admin` (`AdminID`)) ENGINE = InnoDB AUTO_INCREMENT = 8 DEFAULT CHARACTER SET = utf8mb3;

-- Table `smartparkingcity`.`reservation`

CREATE TABLE IF NOT EXISTS `smartparkingcity`.`reservation` (`ReservationID` INT NOT NULL AUTO_INCREMENT, `Arrived` VARCHAR(45) NOT NULL DEFAULT 'No', `StartTime` VARCHAR(45) NOT NULL DEFAULT '', `EndTime` VARCHAR(45) NOT NULL DEFAULT '', `User_idUser` INT NOT NULL DEFAULT '1', `ParkingSpot_SpotID` INT NOT NULL DEFAULT '1', PRIMARY KEY (`ReservationID`), INDEX `fk_Reservation_User1_idx` (`User_idUser` ASC) VISIBLE, INDEX `fk_Reservation_ParkingSpot1_idx` (`ParkingSpot_SpotID` ASC) VISIBLE, CONSTRAINT `fk_Reservation_ParkingSpot1` FOREIGN KEY (`ParkingSpot_SpotID`) REFERENCES `smartparkingcity`.`parkingspot`

(SpotID), CONSTRAINT fk_Reservation_User1 FOREIGN KEY (User_idUser) REFERENCES smartparkingcity.user (idUser)) ENGINE = InnoDB AUTO_INCREMENT = 7 DEFAULT CHARACTER SET = utf8mb3;

SET SQL_MODE=@OLD_SQL_MODE; SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS; SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

SET FOREIGN_KEY_CHECKS = 0; SET FOREIGN_KEY_CHECKS = 1;

DELIMITER $$

CREATE PROCEDURE InsertNewUser(IN p_Name VARCHAR(45), IN p_Email VARCHAR(45), IN p_Password VARCHAR(45), IN p_LicensePlate VARCHAR(45), IN p_PaymentMethod VARCHAR(45), IN p_Debt INT, IN p_AdminID INT) BEGIN INSERT INTO smartparkingcity.User (Name, Email, Password, LicensePlate, PaymentMethod, Debt, Admin_AdminID) VALUES (p_Name, p_Email, p_Password, p_LicensePlate, p_PaymentMethod, p_Debt, p_AdminID); END$$

DELIMITER ;

-- Add a unique index for Email in the User table -- CREATE INDEX idx_user_email ON smartparkingcity.User (Email);

DELIMITER $$

CREATE PROCEDURE UserSignIn(IN p_Email VARCHAR(45), IN p_Password VARCHAR(45), OUT p_UserID INT) BEGIN SELECT idUser INTO p_UserID FROM smartparkingcity.User WHERE Email = p_Email AND Password = p_Password LIMIT 1;

```
-- If no match, set p_UserID to NULL
IF p_UserID IS NULL THEN
    SET p_UserID = NULL;
END IF;
```

END$$

DELIMITER ;

-- Create a composite index for Email and Password in the User table

-- CREATE INDEX idx_user_signin ON smartparkingcity.User (Email, Password);

DELIMITER $$

CREATE PROCEDURE AdminSignIn(IN p_Email VARCHAR(45), IN p_Password VARCHAR(45), OUT p_AdminID INT) BEGIN SELECT AdminID INTO p_AdminID FROM smartparkingcity.Admin WHERE Email = p_Email AND Password = p_Password LIMIT 1;

```
-- If no match, set p_AdminID to NULL
IF p_AdminID IS NULL THEN
    SET p_AdminID = NULL;
END IF;
```

END$$

DELIMITER ;

-- Create a composite index for Email and Password in the Admin table -- CREATE INDEX idx_admin_signin ON smartparkingcity.Admin (Email,Password);

DELIMITER $$

CREATE PROCEDURE GetUserInfoByID(IN p_UserID INT, OUT p_Name VARCHAR(45), OUT p_Email VARCHAR(45), OUT p_LicensePlate VARCHAR(45), OUT p_PaymentMethod VARCHAR(45), OUT p_Debt INT) BEGIN SELECT Name, Email, LicensePlate, PaymentMethod, Debt INTO p_Name, p_Email, p_LicensePlate, p_PaymentMethod, p_Debt FROM smartparkingcity.User WHERE idUser = p_UserID; END$$

DELIMITER ;

-- Create an index for idUser in the User table -- CREATE INDEX idx_user_id ON smartparkingcity.User (idUser);

DELIMITER $$

CREATE PROCEDURE GetAllLotsProfiles() BEGIN SELECT LotID, Directions, Location, Capacity, Revenue FROM smartparkingcity.ParkingLot; END$$

DELIMITER ;

-- Create an index for LotID in the ParkingLot table -- CREATE INDEX idx_parking_lot_id ON smartparkingcity.ParkingLot (LotID);

DELIMITER $$

CREATE PROCEDURE GetAllUsers() BEGIN SELECT idUser, Name, Email, LicensePlate, PaymentMethod, Debt FROM smartparkingcity.User; END$$

DELIMITER ;

DELIMITER $$

CREATE PROCEDURE RemoveUserByID(IN p_UserID INT, IN p_AdminID INT) BEGIN DELETE FROM smartparkingcity.User WHERE idUser = p_UserID AND Admin_AdminID = p_AdminID; END$$

DELIMITER ;

DELIMITER $$

CREATE PROCEDURE ReserveSpot(IN p_SpotID INT, IN p_UserID INT, IN p_AdminID INT, IN p_StartTime DATETIME, IN p_EndTime DATETIME) BEGIN INSERT INTO smartparkingcity.Reservation (ParkingSpot_SpotID, User_idUser, User_Admin_AdminID, StartTime, EndTime) VALUES (p_SpotID, p_UserID, p_AdminID, p_StartTime, p_EndTime); END$$

DELIMITER ;

-- Create an index for SpotID in the ParkingSpot table -- CREATE INDEX idx_parking_spot_id ON smartparkingcity.ParkingSpot (SpotID);

DELIMITER $$

CREATE TRIGGER UpdateParkingSpotStatus AFTER INSERT ON smartparkingcity.Reservation FOR EACH ROW BEGIN DECLARE currentTime DATETIME; SET currentTime = NOW();

```
-- Update parking spot status based on current time
IF currentTime BETWEEN NEW.StartTime AND NEW.EndTime THEN
    UPDATE smartparkingcity.ParkingSpot
    SET Status = 'Reserved'
    WHERE SpotID = NEW.ParkingSpot_SpotID;
```

```
ELSE
    UPDATE smartparkingcity.ParkingSpot
    SET Status = 'Available'
    WHERE SpotID = NEW.ParkingSpot_SpotID;

END IF;


END$$

DELIMITER ;

DELIMITER //

CREATE EVENT UpdateSpotStatusAndDebt ON SCHEDULE EVERY 1 MINUTE DO BEGIN -- Update ParkingSpot status to 'Available' for expired reservations UPDATE smartparkingcity.ParkingSpot PS INNER JOIN smartparkingcity.Reservation R ON PS.SpotID = R.ParkingSpot_SpotID SET PS.Status = 'Available' WHERE R.EndTime < NOW();

-- Apply penalty for users who did not arrive
UPDATE smartparkingcity.User U
INNER JOIN smartparkingcity.Reservation R
ON U.idUser = R.User_idUser
SET U.Debt = 500
WHERE R.EndTime < NOW() AND R.Arrived = 'No';


END; //

DELIMITER ;

DELIMITER $$

CREATE PROCEDURE ReserveParkingSpot( IN p_SpotID INT, IN p_LotID INT, IN p_AdminID INT, IN p_UserID INT, IN p_StartTime DATETIME, IN p_EndTime DATETIME ) BEGIN DECLARE spotStatus VARCHAR(45);

-- Start transaction for concurrency control
START TRANSACTION;
```

```
-- Lock the specific parking spot row to prevent other users from
reserving it simultaneously
SELECT Status INTO spotStatus
FROM smartparkingcity.ParkingSpot
WHERE SpotID = p_SpotID
  AND ParkingLot_LotID = p_LotID
  AND ParkingLot_Admin_AdminID = p_AdminID
FOR UPDATE;

-- Check if the spot is currently available
IF spotStatus = 'Available' THEN
    -- Insert the reservation into the Reservation table
    INSERT INTO smartparkingcity.Reservation (
        Arrived, StartTime, EndTime,
        User_idUser, User_Admin_AdminID,
        ParkingSpot_SpotID, ParkingSpot_ParkingLot_LotID,
ParkingSpot_ParkingLot_Admin_AdminID
    )
    VALUES (
        'No', p_StartTime, p_EndTime,
        p_UserID, p_AdminID,
        p_SpotID, p_LotID, p_AdminID
    );

    -- Commit the transaction if the reservation is successful
    COMMIT;
ELSE
    -- Rollback the transaction if the spot is not available
    ROLLBACK;
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Parking spot is not available.';
END IF;


END$$

DELIMITER ;
```

-- Populate Admin Table INSERT INTO smartparkingcity.Admin (AdminID, Name, Email, Password) VALUES (1, 'John Doe', 'admin1@parkingsystem.com', 'password123'), (2, 'Jane

Smith', 'admin2@parkingsystem.com', 'password456'), (3, 'Samuel Wilson', 'admin3@parkingsystem.com', 'password789'), (4, 'Emily Johnson', 'admin4@parkingsystem.com', 'password321');

-- Populate User Table INSERT INTO smartparkingcity.User (idUser, Debt, LicensePlate, Name, Password, PaymentMethod, Admin_AdminID, Email) VALUES (1, 0, 'ABC123', 'Alice Johnson', 'alicepass', 'Credit Card', 1, 'alice@example.com'), (2, 20, 'XYZ789', 'Bob Brown', 'bobpass', 'Cash', 1, 'bob@example.com'), (3, 10, 'LMN456', 'ReserveSpotCharlie Davis', 'charliepass', 'Debit Card', 2, 'charlie@example.com'), (4, 0, 'JKL123', 'Diana Prince', 'dianapass', 'Credit Card', 2, 'diana@example.com'), (5, 15, 'NOP456', 'Bruce Wayne', 'brucepass', 'Mobile Payment', 3, 'bruce@example.com'), (6, 30, 'QRS789', 'Clark Kent', 'clarkpass', 'Credit Card', 4, 'clark@example.com'), (7, 5, 'TUV001', 'Tony Stark', 'tonypass', 'Cash', 3, 'tony@example.com');

-- Populate ParkingLot Table INSERT INTO smartparkingcity.ParkingLot (LotID, Directions, Location, Capacity, Admin_AdminID, Revenue) VALUES (1, 'Turn left after main gate', 'Downtown Garage', 100, 1, 5000), (2, 'Next to the mall entrance', 'Mall Parking', 150, 2, 7500), (3, 'Near the north gate', 'City Center Parking', 120, 3, 3000), (4, 'Behind the supermarket', 'Supermart Lot', 80, 4, 2000), (5, 'Next to the stadium', 'Stadium Parking', 200, 1, 10000);

-- Populate ParkingSpot Table INSERT INTO smartparkingcity.ParkingSpot (SpotID, Type, Status, Price, ParkingLot_LotID) VALUES (1, 'Regular', 'Available', 10, 1), (2, 'EV', 'Reserved', 15, 1), (3, 'Disabled', 'Available', 12, 1), (4, 'Regular', 'Available', 10, 2), (5, 'EV', 'Occupied', 20, 2), (6, 'Regular', 'Available', 8, 3), (7, 'Disabled', 'Occupied', 10, 3), (8, 'EV', 'Reserved', 15, 3), (9, 'Regular', 'Available', 10, 4), (10, 'Regular', 'Occupied', 12, 4), (11, 'Disabled', 'Available', 10, 5), (12, 'EV', 'Available', 20, 5), (13, 'Regular', 'Available', 10, 5), (14, 'Regular', 'Occupied', 10, 2);

INSERT INTO smartparkingcity.Reservation (ReservationID, Arrived, StartTime, EndTime, User_idUser, ParkingSpot_SpotID) VALUES (1, 'No', '2024-12-26 09:00:00', '2024-12-26 12:00:00', 1, 2), (2, 'Yes', '2024-12-26 08:00:00', '2024-12-26 10:00:00', 2, 5), (3, 'No', '2024-12-26 14:00:00', '2024-12-26 16:00:00', 3, 14), (4, 'Yes', '2024-12-25 18:00:00', '2024-12-25 20:00:00', 4, 7), (5, 'Yes', '2024-12-26 10:00:00', '2024-12-26 13:00:00', 5, 6), (6, 'No', '2024-12-26 15:00:00', '2024-12-26 17:00:00', 6, 8);

UPDATE smartparkingcity.ParkingLot SET Directions = 'Move 20m to the left,and 5m right' WHERE LotID = 1;

UPDATE smartparkingcity.ParkingLot SET Directions = 'Go straight for 30m, then turn left for 15m' WHERE LotID = 2;

UPDATE smartparkingcity.ParkingLot SET Directions = 'Head north for 50m, then turn right for 20m' WHERE LotID = 3;

UPDATE smartparkingcity.ParkingLot SET Directions = 'Go straight for 40m, then left after 10m' WHERE LotID = 4;

UPDATE smartparkingcity.ParkingLot SET Directions = 'Drive 100m east, then turn left for 30m' WHERE LotID = 5;

-- SHOW INDEX FROM smartparkingcity.User; -- DROP INDEX idx_user_signin ON smartparkingcity.User; -- DROP INDEX idx_user_email ON smartparkingcity.User;

-- CREATE INDEX idx_user_email ON smartparkingcity.User (Email); -- Add a unique index for Email in the User table CREATE UNIQUE INDEX idx_user_email ON smartparkingcity.User (Email)

-- Index for the Admin table CREATE INDEX idx_admin_signin ON smartparkingcity.admin (Email, Password);

-- Index for the ParkingLot table CREATE INDEX idx_parking_lot_id ON smartparkingcity.ParkingLot (LotID); CREATE INDEX fk_ParkingLot_Admin1_idx ON smartparkingcity.ParkingLot (Admin_AdminID);

-- Index for the ParkingSpot table CREATE INDEX idx_parking_spot_id ON smartparkingcity.ParkingSpot (SpotID); CREATE INDEX fk_ParkingSpot_ParkingLot1_idx ON smartparkingcity.ParkingSpot (ParkingLot_LotID);

-- Index for the User table CREATE INDEX idx_user_id ON smartparkingcity.user (idUser); CREATE INDEX idx_user_email ON smartparkingcity.user (Email); CREATE INDEX idx_user_signin ON smartparkingcity.user (Email, Password); CREATE INDEX fk_User_Admin_idx ON smartparkingcity.user (Admin_AdminID);

-- Index for the Reservation table CREATE INDEX fk_Reservation_User1_idx ON smartparkingcity.reservation (User_idUser); CREATE INDEX fk_Reservation_ParkingSpot1_idx ON smartparkingcity.reservation (ParkingSpot_SpotID);

-- Check indexes on the admin table SHOW INDEX FROM smartparkingcity.admin;

-- Check indexes on the parkingtable table SHOW INDEX FROM smartparkingcity.ParkingLot;

-- Check indexes on the parkingowner table SHOW INDEX FROM smartparkingcity.ParkingSpot;

-- Check indexes on the user table SHOW INDEX FROM smartparkingcity.user

Triggers, Stored Procedures, and Constraints

- **Triggers:**
  - **UpdateParkingSpotStatus**: This trigger updates the status of a parking spot from `Available` to `Reserved` or back to `Available` when a reservation is made or canceled. It operates after a reservation is inserted.
- **Stored Procedures:**

  User Management:

  - InsertNewUser: Adds a new user to the database.
  - UserSignIn: Verifies user login by comparing Email and Password.
  - AdminSignIn: Verifies administrator login credentials.
  - RemoveUserByID: Deletes a user based on UserID and AdminID.

  Parking Spot Reservations:

  - ReserveParkingSpot: Handles the reservation of parking spots by a user, ensuring the spot is available before inserting the reservation. Uses a transaction to lock the spot during reservation to prevent concurrent reservations of the same spot.
  - ReserveSpot: A legacy procedure for reserving a spot.
  - UpdateParkingSpotStatus: Updates the status of parking spots after a reservation is made.

- **Concurrency Control:**

- Transaction Management: In the ReserveParkingSpot procedure, a transaction is initiated using START TRANSACTION to ensure that a parking spot is reserved without conflicts. The FOR UPDATE statement locks the specific parking spot row while the reservation is being processed. If the spot is available, the reservation is inserted; if not, the transaction is rolled back, preventing double booking.
- Isolation: The procedure uses a read lock (FOR UPDATE) to prevent other transactions from modifying the parking spot's status during the reservation process.

Constraints to Ensure Data Integrity

- **Foreign Key Constraints**:
  - Foreign keys are used between tables (e.g., `User` to `Admin`, `ParkingSpot` to `ParkingLot`) to ensure referential integrity.
  - When an admin or parking lot is deleted, corresponding records in the related tables are also deleted or updated to prevent orphaned data.
- **Unique Constraints**:
  - The `Email` field in both the `User` and `Admin` tables is unique, preventing users from registering multiple accounts with the same email.
- **Check Constraints**: Ensure valid data entries like the `Status` of parking spots and the valid range of reservation times.

7. Events

- **UpdateSpotStatusAndDebt**: An event that runs every minute, checking expired reservations and updating the status of parking spots to `Available` if the reservation has ended. It also applies a penalty to users who did not arrive for their reservations (`Arrived = 'No'`).

3. Performance Optimization

Indexing

- **Indexes** are used across the `User`, `Admin`, `ParkingLot`, and `ParkingSpot` tables to speed up common queries like logins, lookups, and status checks.
- Unique indexes on `Email` in the `User` table ensure no two users share the same email address.
- Composite indexes are created for the `User` and `Admin` tables on `Email` and `Password` to speed up authentication.

Query Optimization

- Avoids full table scans by utilizing indexed queries for searching reservations and user authentication.

Database Tuning

- `AUTO_INCREMENT` ensures unique primary keys.
- Optimized triggers and stored procedures to minimize execution time.

Additional Optimizations

- Regularly scheduled events for:
    - Resetting expired reservations.
    - Applying penalties for no-shows.

4. Simulation:

```
Time 0: A - A1 is now available
Time 0: A - A2 is now available
Time 0: B - B1 is now occupied
Time 0: B - B2 is now available
Time 2: B - B1 is now available
Time 2: B - B2 is now available
Time 5: A - A1 is now occupied
Time 5: A - A2 is now occupied
Time 5: B - B1 is now available
Time 5: B - B2 is now occupied
Time 7: A - A1 is now occupied
Time 9: B - B2 is now occupied
Time 10: A - A2 is now occupied
Time 10: B - B1 is now occupied
Time 10: A - A1 is now available
Time 13: B - B2 is now available
Time 13: A - A2 is now available
Time 13: B - B1 is now available
Time 13: A - A1 is now available
Time 16: A - A2 is now occupied
Time 18: B - B2 is now available
Time 18: B - B1 is now available
Time 18: A - A1 is now available
Time 19: A - A2 is now available
Time 20: B - B1 is now available
Time 20: A - A1 is now occupied
Time 21: B - B2 is now available
Time 23: A - A2 is now available
Time 23: B - B1 is now available
Time 24: A - A1 is now occupied
Time 26: B - B2 is now available
Time 26: B - B1 is now occupied
Time 28: A - A2 is now occupied
Time 28: B - B2 is now occupied
Time 29: A - A1 is now occupied
```

spotid	parking_lot_lotid	price	status	type
1	1	10	Available	Regular
2	1	15	Reserved	EV
3	1	12	Available	Disabled
4	2	10	Available	Regular
5	2	20	Occupied	EV
6	3	8	Available	Regular
7	3	10	Occupied	Disabled
8	3	15	Reserved	EV
9	4	10	Available	Regular
10	4	12	Occupied	Regular
11	5	10	Available	Disabled
12	5	20	Available	EV
13	5	10	Available	Regular
14	2	10	Occupied	Regular
15	3	18	Available	EV
16	3	15	Occupied	Regular
17	2	12	Reserved	Disabled
18	1	10	Available	Regular
19	4	20	Occupied	EV
20	5	25	Available	Premium
NULL	NULL	NULL	NULL	NULL

Time 0: 1 - Spot 1 is now occupied

Updated 1 - 1 to occupied in database.

Time 0: 1 - Spot 2 is now occupied

Updated 1 - 2 to occupied in database.

Time 0: 1 - Spot 3 is now occupied

Updated 1 - 3 to occupied in database.

Time 0: 1 - Spot 18 is now occupied

Updated 1 - 18 to occupied in database.

Time 0: 2 - Spot 4 is now occupied

Updated 2 - 4 to occupied in database.

Time 0: 2 - Spot 5 is now available

Updated 2 - 5 to available in database.

Time 0: 2 - Spot 14 is now available

Updated 2 - 14 to available in database.

Time 0: 2 - Spot 17 is now available

Updated 2 - 17 to available in database.

Time 0: 3 - Spot 6 is now available

Updated 3 - 6 to available in database.

Time 0: 3 - Spot 7 is now occupied

Updated 3 - 7 to occupied in database.

Time 0: 3 - Spot 8 is now occupied

Updated 3 - 8 to occupied in database.

Time 0: 3 - Spot 15 is now available

Updated 3 - 15 to available in database.

Time 0: 3 - Spot 16 is now occupied

Updated 3 - 16 to occupied in database.

Time 0: 4 - Spot 9 is now available

Updated 4 - 9 to available in database.

Time 0: 4 - Spot 10 is now available

Updated 4 - 10 to available in database.

Time 0: 4 - Spot 19 is now occupied

Updated 4 - 19 to occupied in database.

Time 0: 5 - Spot 11 is now available

Updated 5 - 11 to available in database.

Time 0: 5 - Spot 12 is now occupied

Updated 5 - 12 to occupied in database.

Time 0: 5 - Spot 13 is now available

Updated 5 - 13 to available in database.

Time 0: 5 - Spot 20 is now occupied

Updated 5 - 20 to occupied in database.

Time 2: 1 - Spot 1 is now occupied

Updated 1 - 1 to occupied in database.

Time 2: 2 - Spot 4 is now occupied

Updated 2 - 4 to occupied in database.

Time 3: 1 - Spot 3 is now occupied

Updated 1 - 3 to occupied in database.

Time 3: 1 - Spot 18 is now available

Updated 1 - 18 to available in database.

Time 3: 2 - Spot 14 is now available

Updated 2 - 14 to available in database.

Time 3: 3 - Spot 6 is now occupied

Updated 3 - 6 to occupied in database.

Time 3: 3 - Spot 15 is now available

Updated 3 - 15 to available in database.

Time 3: 3 - Spot 16 is now available

Updated 3 - 16 to available in database.

Time 3: 4 - Spot 19 is now occupied

Updated 4 - 19 to occupied in database.

Time 3: 5 - Spot 11 is now occupied

Updated 5 - 11 to occupied in database.

Time 3: 5 - Spot 13 is now occupied

Updated 5 - 13 to occupied in database.

Time 4: 1 - Spot 2 is now occupied

Updated 1 - 2 to occupied in database.

Time 4: 2 - Spot 5 is now occupied

Updated 2 - 5 to occupied in database.

Time 4: 3 - Spot 7 is now available

Updated 3 - 7 to available in database.

Time 4: 3 - Spot 8 is now occupied

Updated 3 - 8 to occupied in database.

Time 4: 4 - Spot 9 is now occupied

Updated 4 - 9 to occupied in database.

Time 5: 2 - Spot 17 is now occupied

Updated 2 - 17 to occupied in database.

Time 5: 4 - Spot 10 is now occupied

Updated 4 - 10 to occupied in database.

Time 5: 5 - Spot 12 is now occupied

Updated 5 - 12 to occupied in database.

Time 5: 5 - Spot 20 is now available

Updated 5 - 20 to available in database.

Time 5: 2 - Spot 4 is now occupied

Updated 2 - 4 to occupied in database.

Time 5: 1 - Spot 18 is now available

Updated 1 - 18 to available in database.

Time 5: 2 - Spot 14 is now occupied

Updated 2 - 14 to occupied in database.

Time 6: 1 - Spot 1 is now occupied

Updated 1 - 1 to occupied in database.

Time 6: 3 - Spot 6 is now occupied

Updated 3 - 6 to occupied in database.

Time 6: 5 - Spot 13 is now occupied

Updated 5 - 13 to occupied in database.

Time 6: 4 - Spot 9 is now occupied

Updated 4 - 9 to occupied in database.

Time 7: 1 - Spot 3 is now available

Updated 1 - 3 to available in database.

Time 7: 3 - Spot 15 is now occupied

Updated 3 - 15 to occupied in database.

Time 7: 4 - Spot 19 is now occupied

Updated 4 - 19 to occupied in database.

Time 7: 4 - Spot 10 is now occupied

Updated 4 - 10 to occupied in database.

Time 7: 5 - Spot 12 is now occupied

Updated 5 - 12 to occupied in database.

Time 7: 5 - Spot 20 is now available

Updated 5 - 20 to available in database.

Time 8: 3 - Spot 16 is now available

Updated 3 - 16 to available in database.

Time 8: 5 - Spot 11 is now occupied

Updated 5 - 11 to occupied in database.

Time 8: 3 - Spot 7 is now occupied

Updated 3 - 7 to occupied in database.

Time 8: 1 - Spot 1 is now available

Updated 1 - 1 to available in database.

Time 9: 1 - Spot 2 is now occupied

Updated 1 - 2 to occupied in database.

Time 9: 2 - Spot 5 is now available

Updated 2 - 5 to available in database.

Time 9: 3 - Spot 8 is now occupied

Updated 3 - 8 to occupied in database.

Time 9: 2 - Spot 17 is now occupied

Updated 2 - 17 to occupied in database.

Time 9: 2 - Spot 4 is now available

Updated 2 - 4 to available in database.

Time 9: 1 - Spot 3 is now available

Updated 1 - 3 to available in database.

Time 9: 5 - Spot 12 is now occupied

Updated 5 - 12 to occupied in database.

Time 10: 1 - Spot 18 is now available

Updated 1 - 18 to available in database.

Time 10: 2 - Spot 14 is now occupied

Updated 2 - 14 to occupied in database.

Time 10: 5 - Spot 13 is now available

Updated 5 - 13 to available in database.

Time 10: 3 - Spot 15 is now occupied

Updated 3 - 15 to occupied in database.

Time 11: 3 - Spot 6 is now occupied

Updated 3 - 6 to occupied in database.

Time 11: 4 - Spot 9 is now available

Updated 4 - 9 to available in database.

Time 11: 4 - Spot 19 is now occupied

Updated 4 - 19 to occupied in database.

Time 11: 4 - Spot 10 is now available

Updated 4 - 10 to available in database.

Time 11: 5 - Spot 11 is now available

Updated 5 - 11 to available in database.

Time 11: 1 - Spot 1 is now occupied

Updated 1 - 1 to occupied in database.

Time 11: 1 - Spot 2 is now available

Updated 1 - 2 to available in database.

Time 11: 2 - Spot 4 is now occupied

Updated 2 - 4 to occupied in database.

Time 12: 5 - Spot 20 is now available

Updated 5 - 20 to available in database.

Time 12: 3 - Spot 16 is now available

Updated 3 - 16 to available in database.

Time 12: 3 - Spot 7 is now available

Updated 3 - 7 to available in database.

Time 12: 2 - Spot 17 is now available

Updated 2 - 17 to available in database.

Time 12: 3 - Spot 15 is now available

Updated 3 - 15 to available in database.

Time 13: 2 - Spot 5 is now available

Updated 2 - 5 to available in database.

Time 13: 1 - Spot 3 is now available

Updated 1 - 3 to available in database.

Time 13: 5 - Spot 12 is now available

Updated 5 - 12 to available in database.

Time 13: 5 - Spot 13 is now occupied

Updated 5 - 13 to occupied in database.

Time 13: 1 - Spot 2 is now available

Updated 1 - 2 to available in database.

Time 14: 3 - Spot 8 is now available

Updated 3 - 8 to available in database.

Time 14: 1 - Spot 18 is now available

Updated 1 - 18 to available in database.

Time 14: 2 - Spot 14 is now available

Updated 2 - 14 to available in database.

Time 14: 4 - Spot 19 is now occupied

Updated 4 - 19 to occupied in database.

Time 14: 3 - Spot 16 is now available

Updated 3 - 16 to available in database.

Time 14: 3 - Spot 15 is now occupied

Updated 3 - 15 to occupied in database.

Time 15: 4 - Spot 10 is now available

Updated 4 - 10 to available in database.

Time 15: 5 - Spot 11 is now occupied

Updated 5 - 11 to occupied in database.

Time 15: 3 - Spot 7 is now occupied

Updated 3 - 7 to occupied in database.

Time 16: 3 - Spot 6 is now occupied

Updated 3 - 6 to occupied in database.

Time 16: 4 - Spot 9 is now occupied

Updated 4 - 9 to occupied in database.

Time 16: 1 - Spot 1 is now available

Updated 1 - 1 to available in database.

Time 16: 2 - Spot 4 is now available

Updated 2 - 4 to available in database.

Time 16: 1 - Spot 3 is now available

Updated 1 - 3 to available in database.

Time 16: 1 - Spot 2 is now occupied

Updated 1 - 2 to occupied in database.

Time 16: 3 - Spot 15 is now available

Updated 3 - 15 to available in database.

Time 17: 5 - Spot 20 is now occupied

Updated 5 - 20 to occupied in database.

Time 17: 2 - Spot 17 is now occupied

Updated 2 - 17 to occupied in database.

Time 17: 2 - Spot 5 is now available

Updated 2 - 5 to available in database.

Time 17: 5 - Spot 12 is now available

Updated 5 - 12 to available in database.

Time 17: 3 - Spot 8 is now occupied

Updated 3 - 8 to occupied in database.

Time 17: 2 - Spot 14 is now occupied

Updated 2 - 14 to occupied in database.

Time 17: 3 - Spot 16 is now available

Updated 3 - 16 to available in database.

Time 17: 5 - Spot 11 is now occupied

Updated 5 - 11 to occupied in database.

Time 17: 3 - Spot 7 is now occupied

Updated 3 - 7 to occupied in database.

Time 18: 5 - Spot 13 is now occupied

Updated 5 - 13 to occupied in database.

Time 18: 4 - Spot 19 is now available

Updated 4 - 19 to available in database.

Time 18: 1 - Spot 3 is now occupied

Updated 1 - 3 to occupied in database.

Time 19: 1 - Spot 18 is now occupied

Updated 1 - 18 to occupied in database.

Time 19: 4 - Spot 10 is now occupied

Updated 4 - 10 to occupied in database.

Time 19: 1 - Spot 2 is now available

Updated 1 - 2 to available in database.

Time 19: 3 - Spot 15 is now available

Updated 3 - 15 to available in database.

Time 19: 2 - Spot 17 is now occupied

Updated 2 - 17 to occupied in database.

Time 19: 2 - Spot 14 is now occupied

Updated 2 - 14 to occupied in database.

Time 20: 4 - Spot 9 is now occupied

Updated 4 - 9 to occupied in database.

Time 20: 1 - Spot 1 is now available

Updated 1 - 1 to available in database.

Time 20: 2 - Spot 4 is now available

Updated 2 - 4 to available in database.

Time 20: 4 - Spot 19 is now occupied

Updated 4 - 19 to occupied in database.

Time 21: 3 - Spot 6 is now occupied

Updated 3 - 6 to occupied in database.

Time 21: 5 - Spot 20 is now occupied

Updated 5 - 20 to occupied in database.

Time 21: 2 - Spot 5 is now occupied

Updated 2 - 5 to occupied in database.

Time 21: 3 - Spot 8 is now occupied

Updated 3 - 8 to occupied in database.

Time 21: 3 - Spot 16 is now available

Updated 3 - 16 to available in database.

Time 21: 3 - Spot 7 is now available

Updated 3 - 7 to available in database.

Time 21: 1 - Spot 2 is now available

Updated 1 - 2 to available in database.

Time 22: 5 - Spot 12 is now occupied

Updated 5 - 12 to occupied in database.

Time 22: 5 - Spot 11 is now available

Updated 5 - 11 to available in database.

Time 22: 1 - Spot 3 is now available

Updated 1 - 3 to available in database.

Time 22: 2 - Spot 14 is now available

Updated 2 - 14 to available in database.

Time 22: 4 - Spot 9 is now available

Updated 4 - 9 to available in database.

Time 22: 1 - Spot 1 is now available

Updated 1 - 1 to available in database.

Time 22: 4 - Spot 19 is now available

Updated 4 - 19 to available in database.

Time 23: 5 - Spot 13 is now available

Updated 5 - 13 to available in database.

Time 23: 4 - Spot 10 is now occupied

Updated 4 - 10 to occupied in database.

Time 23: 3 - Spot 8 is now occupied

Updated 3 - 8 to occupied in database.

Time 24: 1 - Spot 18 is now occupied

Updated 1 - 18 to occupied in database.

Time 24: 3 - Spot 15 is now available

Updated 3 - 15 to available in database.

Time 24: 2 - Spot 17 is now occupied

Updated 2 - 17 to occupied in database.

Time 24: 2 - Spot 4 is now available

Updated 2 - 4 to available in database.

Time 24: 5 - Spot 20 is now occupied

Updated 5 - 20 to occupied in database.

Time 24: 2 - Spot 5 is now available

Updated 2 - 5 to available in database.

Time 24: 5 - Spot 12 is now available

Updated 5 - 12 to available in database.

Time 24: 4 - Spot 9 is now occupied

Updated 4 - 9 to occupied in database.

Time 25: 3 - Spot 16 is now occupied

Updated 3 - 16 to occupied in database.

Time 25: 1 - Spot 2 is now available

Updated 1 - 2 to available in database.

Time 25: 2 - Spot 14 is now occupied

Updated 2 - 14 to occupied in database.

Time 25: 4 - Spot 19 is now available

Updated 4 - 19 to available in database.

Time 26: 3 - Spot 6 is now available

Updated 3 - 6 to available in database.

Time 26: 3 - Spot 7 is now occupied

Updated 3 - 7 to occupied in database.

Time 26: 1 - Spot 3 is now occupied

Updated 1 - 3 to occupied in database.

Time 26: 2 - Spot 4 is now occupied

Updated 2 - 4 to occupied in database.

Time 26: 5 - Spot 20 is now occupied

Updated 5 - 20 to occupied in database.

Time 26: 2 - Spot 5 is now occupied

Updated 2 - 5 to occupied in database.

Time 27: 5 - Spot 11 is now available

Updated 5 - 11 to available in database.

Time 27: 1 - Spot 1 is now occupied

Updated 1 - 1 to occupied in database.

Time 27: 5 - Spot 13 is now available

Updated 5 - 13 to available in database.

Time 27: 3 - Spot 8 is now occupied

Updated 3 - 8 to occupied in database.

Time 27: 3 - Spot 16 is now available

Updated 3 - 16 to available in database.

Time 27: 2 - Spot 14 is now occupied

Updated 2 - 14 to occupied in database.

Time 28: 4 - Spot 10 is now available

Updated 4 - 10 to available in database.

Time 28: 1 - Spot 18 is now available

Updated 1 - 18 to available in database.

Time 28: 3 - Spot 15 is now occupied

Updated 3 - 15 to occupied in database.

Time 28: 5 - Spot 12 is now available

Updated 5 - 12 to available in database.

Time 28: 3 - Spot 6 is now occupied

Updated 3 - 6 to occupied in database.

Time 29: 2 - Spot 17 is now occupied

Updated 2 - 17 to occupied in database.

Time 29: 4 - Spot 9 is now available

Updated 4 - 9 to available in database.

Time 29: 1 - Spot 2 is now available

Updated 1 - 2 to available in database.

Time 29: 4 - Spot 19 is now occupied

Updated 4 - 19 to occupied in database.

Time 29: 2 - Spot 4 is now occupied

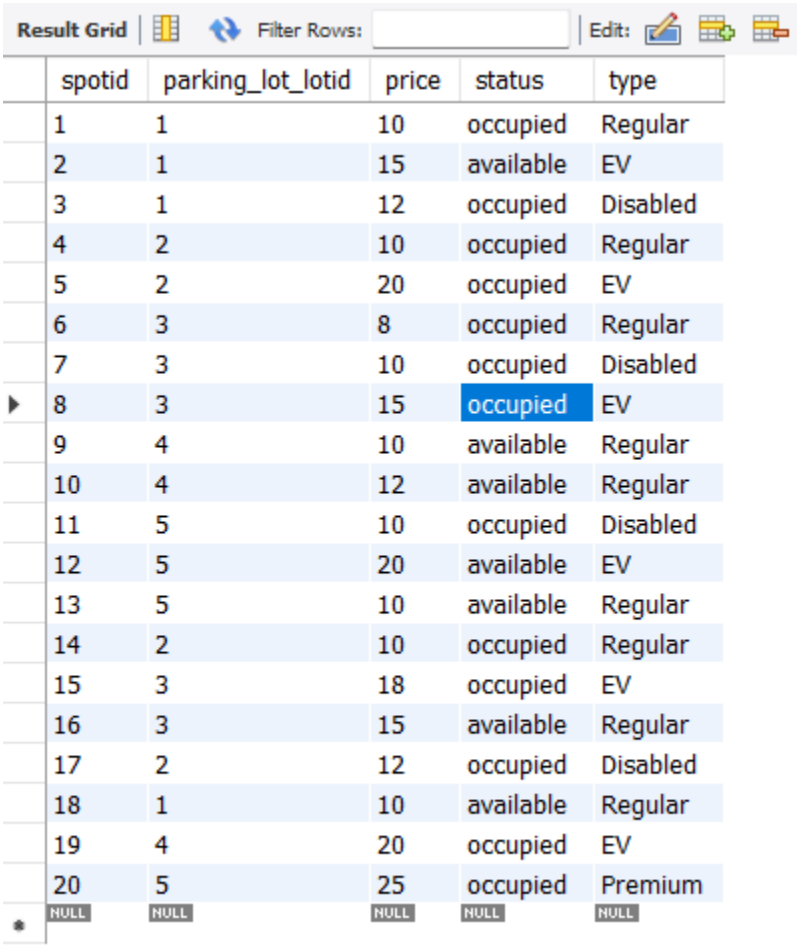
Updated 2 - 4 to occupied in database.

Time 29: 5 - Spot 11 is now occupied

Updated 5 - 11 to occupied in database.

Time 29: 3 - Spot 16 is now available

Updated 3 - 16 to available in database.



Result Grid | Filter Rows: | Edit:

spotid	parking_lot_lotid	price	status	type
1	1	10	occupied	Regular
2	1	15	available	EV
3	1	12	occupied	Disabled
4	2	10	occupied	Regular
5	2	20	occupied	EV
6	3	8	occupied	Regular
7	3	10	occupied	Disabled
8	3	15	occupied	EV
9	4	10	available	Regular
10	4	12	available	Regular
11	5	10	occupied	Disabled
12	5	20	available	EV
13	5	10	available	Regular
14	2	10	occupied	Regular
15	3	18	occupied	EV
16	3	15	available	Regular
17	2	12	occupied	Disabled
18	1	10	available	Regular
19	4	20	occupied	EV
20	5	25	occupied	Premium
NULL	NULL	NULL	NULL	NULL

6. Reporting (Jasper):

Admin Dashborad

Admin

Smart Parking System

Saturday 28 December

Spot ID	Occupancy Count	Total Revenue	Violation Count
1	2	20	0
5	1	30	1
4	1	25	0
3	1	10	0
2	1	10	1
10	1	30	0

Parking Spot Info

Parking Spot

Smart Parking System

Saturday 28 December

Spot ID	Total Revenue
5	20
19	20
15	18
2	15
8	15
16	15
17	12
7	10
14	10
18	10
6	8

Top Users

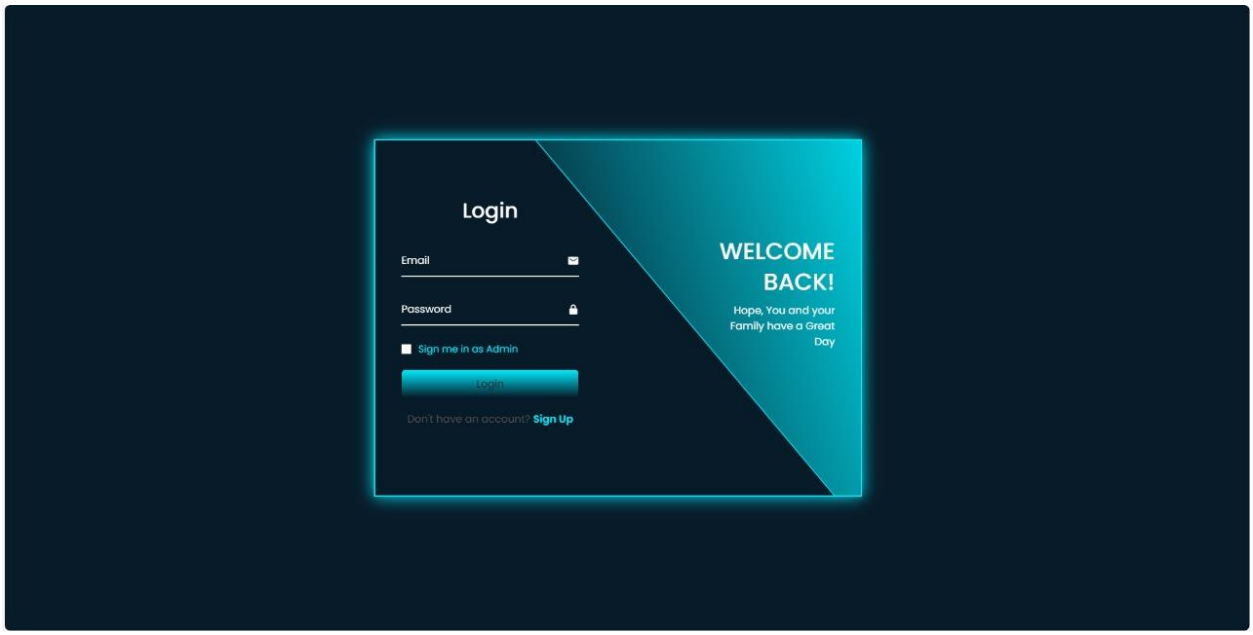
Admin

Smart Parking System

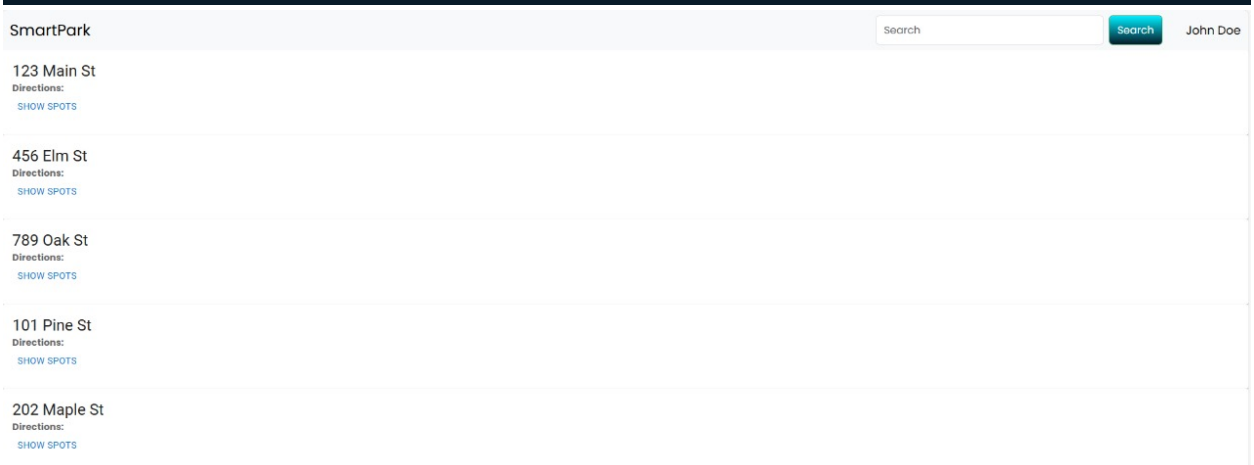
Saturday 28 December

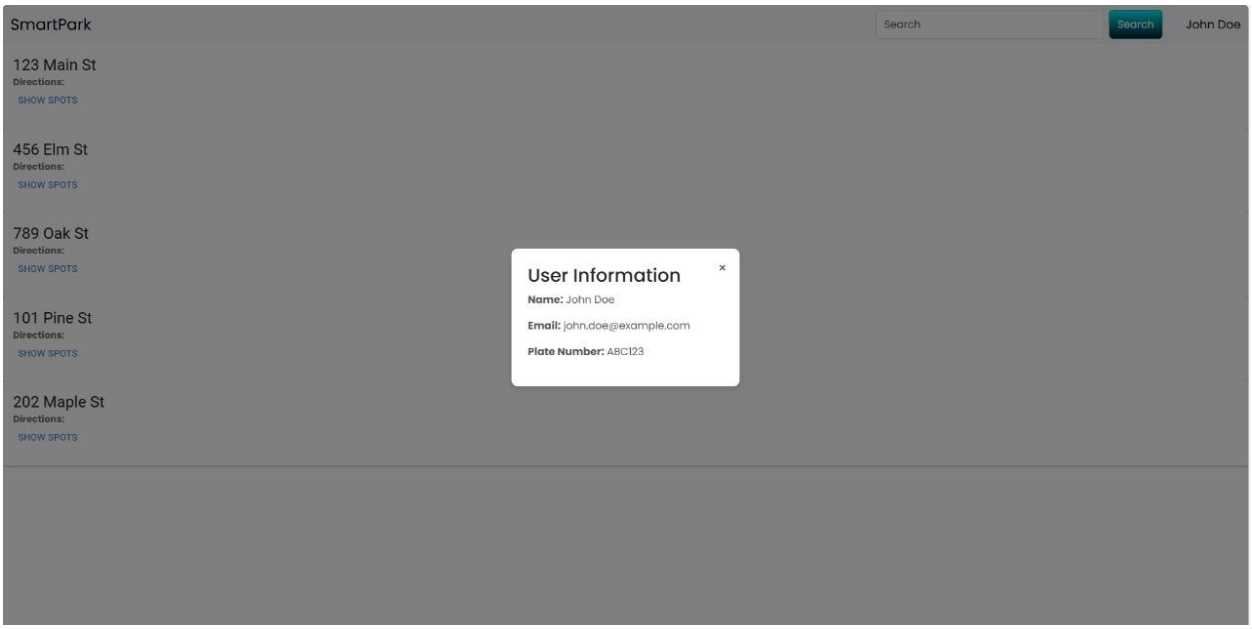
User ID	Reservation Count
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1
11	1

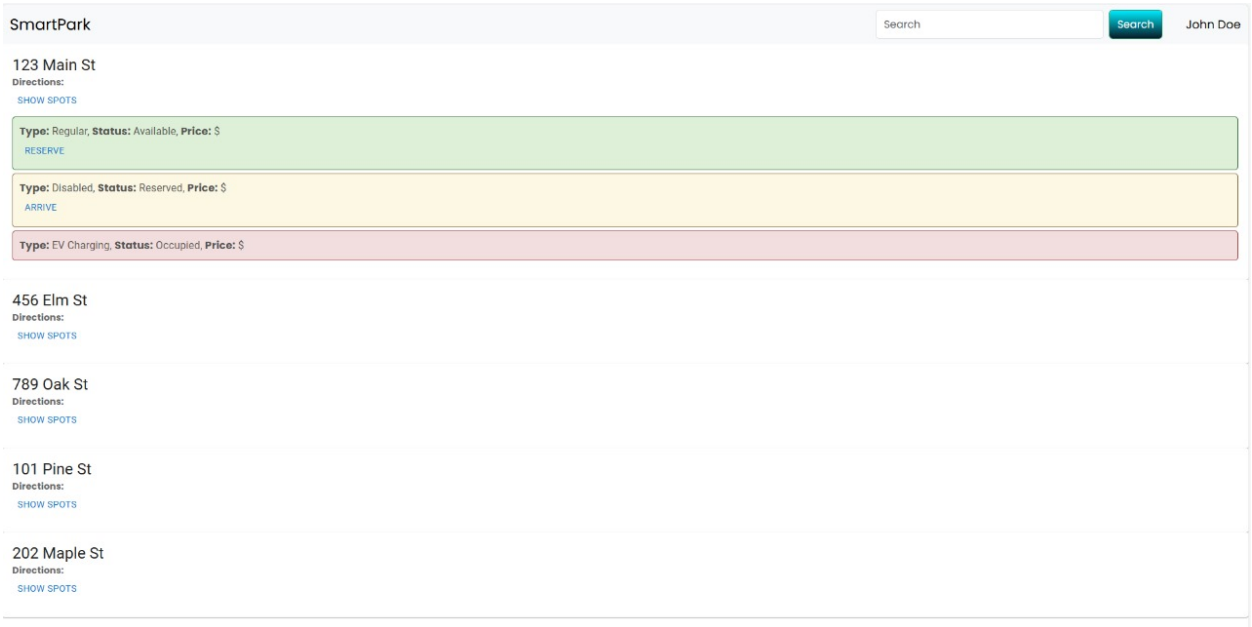
7. Screenshots and Functionality:

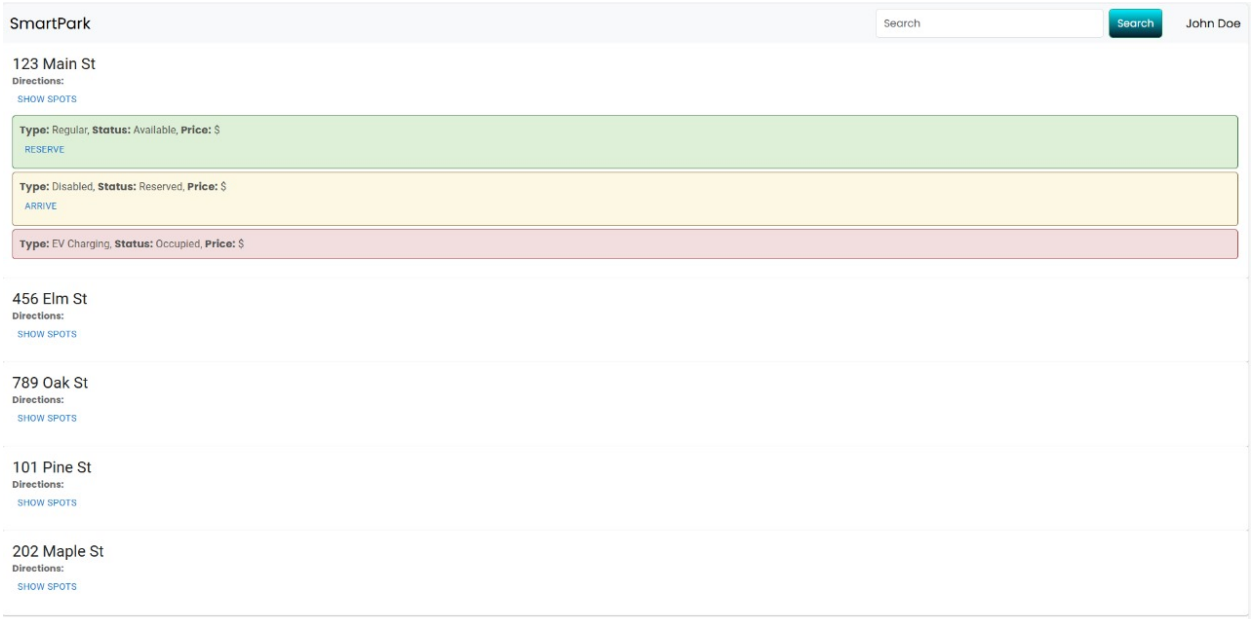

Login
Email
Password
Sign me in as Admin
Login
Don't have an account? Sign Up
WELCOME BACK!
Hope, You and your Family have a Great Day

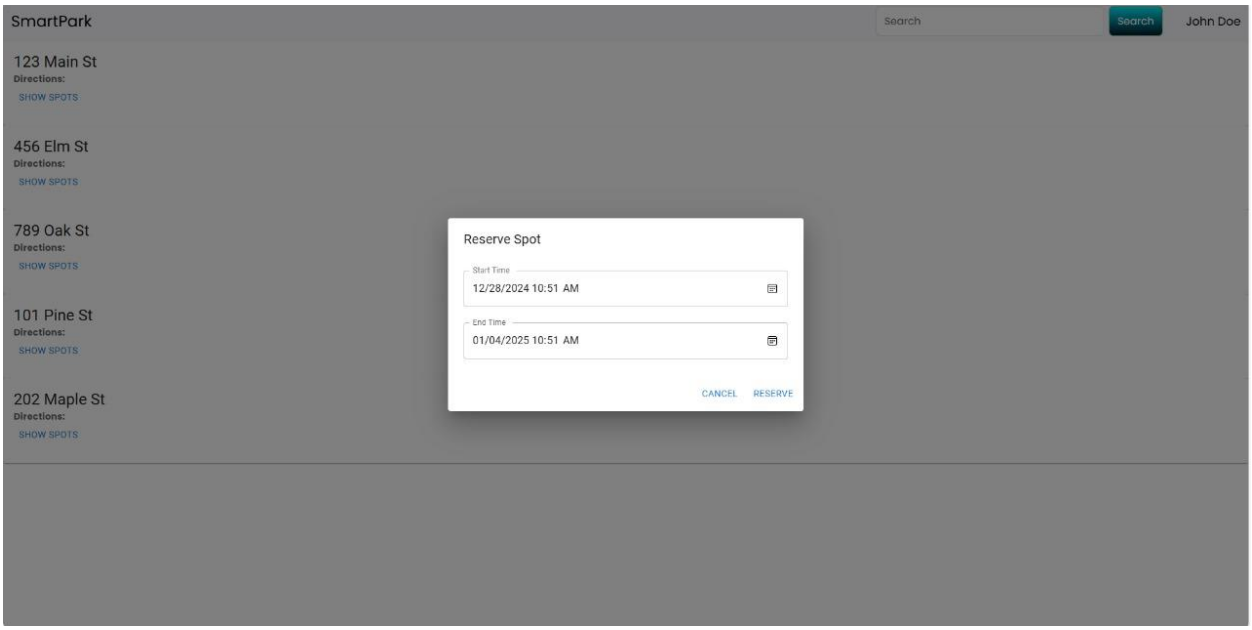

Sign Up
Username
Email
Password
Plate Number
Register
Already have an account? Login
WELCOME!
Hope, You and your Family have a Great Day

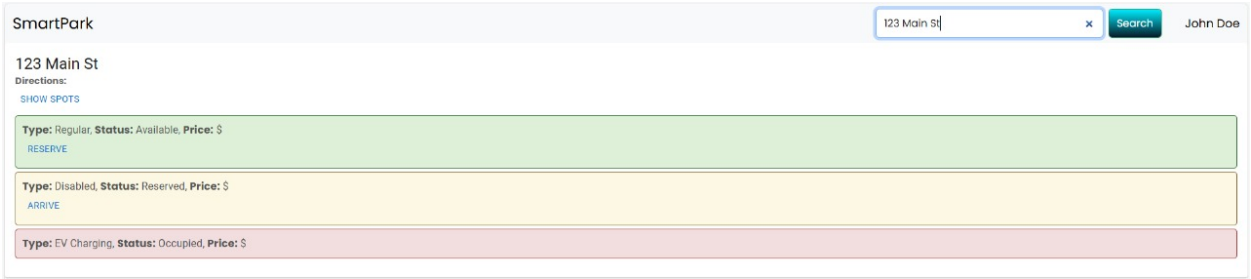

SmartPark
Search
Search
John Doe
123 Main St
Directions:
SHOW SPOTS
456 Elm St
Directions:
SHOW SPOTS
789 Oak St
Directions:
SHOW SPOTS
101 Pine St
Directions:
SHOW SPOTS
202 Maple St
Directions:
SHOW SPOTS

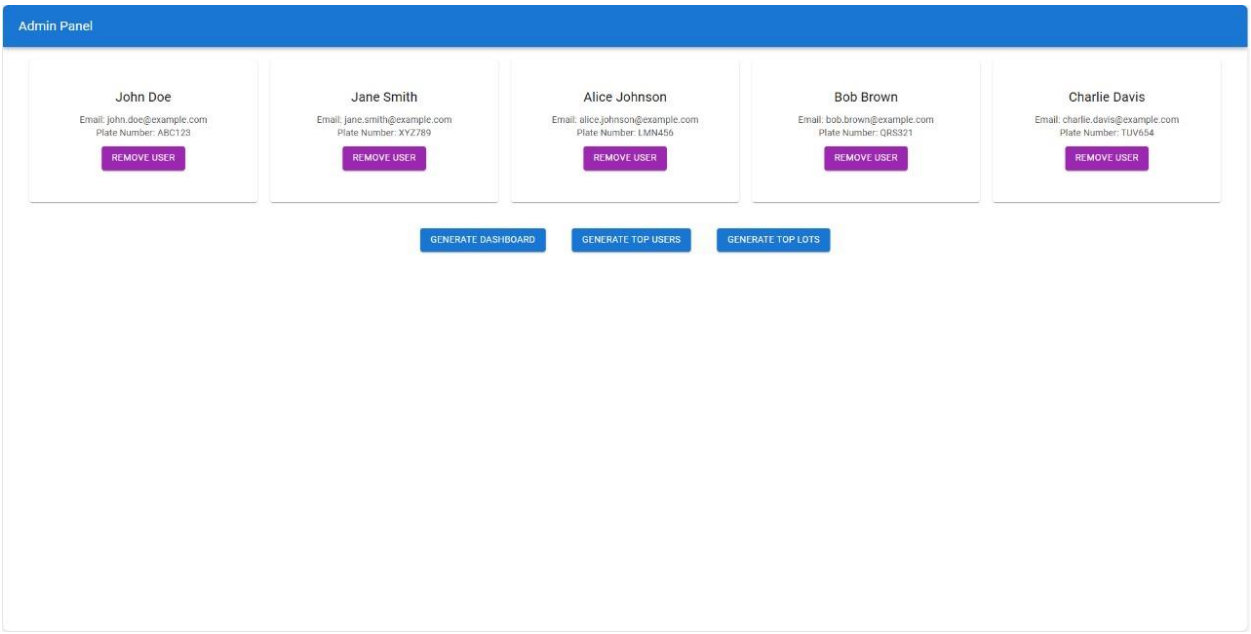

SmartPark
Search
Search
John Doe
123 Main St
Directions:
SHOW SPOTS
456 Elm St
Directions:
SHOW SPOTS
789 Oak St
Directions:
SHOW SPOTS
101 Pine St
Directions:
SHOW SPOTS
202 Maple St
Directions:
SHOW SPOTS
User Information
Name: John Doe
Email: john.doe@example.com
Plate Number: ABC123

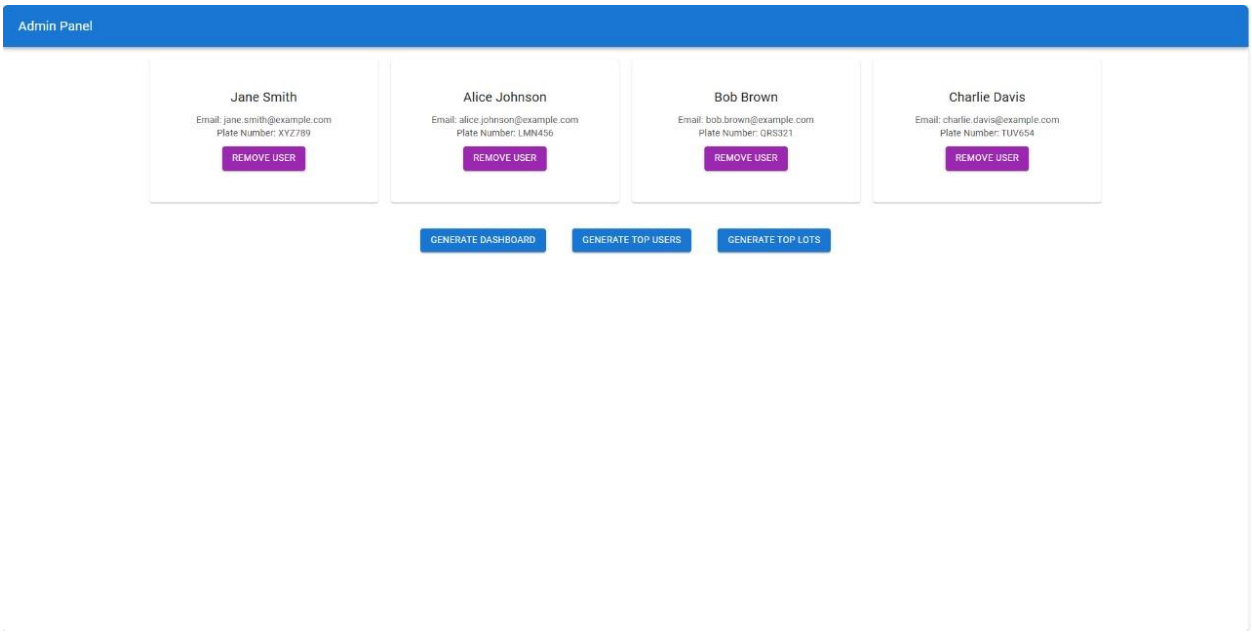

SmartPark
Search
Search
John Doe
123 Main St
Directions:
SHOW SPOTS
Type: Regular, Status: Available, Price: $
RESERVE
Type: Disabled, Status: Reserved, Price: $
ARRIVE
Type: EV Charging, Status: Occupied, Price: $
456 Elm St
Directions:
SHOW SPOTS
789 Oak St
Directions:
SHOW SPOTS
101 Pine St
Directions:
SHOW SPOTS
202 Maple St
Directions:
SHOW SPOTS


SmartPark
Search
Search
John Doe
123 Main St
Directions:
SHOW SPOTS
Type: Regular, Status: Available, Price: $
RESERVE
Type: Disabled, Status: Reserved, Price: $
ARRIVE
Type: EV Charging, Status: Occupied, Price: $
456 Elm St
Directions:
SHOW SPOTS
789 Oak St
Directions:
SHOW SPOTS
101 Pine St
Directions:
SHOW SPOTS
202 Maple St
Directions:
SHOW SPOTS


SmartPark
Search
Search
John Doe
123 Main St
Directions:
SHOW SPOTS
456 Elm St
Directions:
SHOW SPOTS
789 Oak St
Directions:
SHOW SPOTS
101 Pine St
Directions:
SHOW SPOTS
202 Maple St
Directions:
SHOW SPOTS
Reserve Spot
Start Time
12/28/2024 10:51 AM
End Time
01/04/2025 10:51 AM
CANCEL
RESERVE


SmartPark
123 Main St
Search
John Doe
123 Main St
Directions:
SHOW SPOTS
Type: Regular, Status: Available, Price: $
RESERVE
Type: Disabled, Status: Reserved, Price: $
ARRIVE
Type: EV Charging, Status: Occupied, Price: $


Admin Panel
John Doe
Email: john.doe@example.com
Plate Number: ABC123
REMOVE USER
Jane Smith
Email: jane.smith@example.com
Plate Number: XYZ789
REMOVE USER
Alice Johnson
Email: alice.johnson@example.com
Plate Number: LMN456
REMOVE USER
Bob Brown
Email: bob.brown@example.com
Plate Number: QRS321
REMOVE USER
Charlie Davis
Email: charlie.davis@example.com
Plate Number: TUV654
REMOVE USER
GENERATE DASHBOARD
GENERATE TOP USERS
GENERATE TOP LOTS



8. Conclusion:

The Smart City Parking Management System provides an efficient, scalable solution for managing parking lot reservations in urban environments. The database schema captures essential information related to users, parking spots, reservations, and administrative operations, enabling seamless integration with various system functionalities.

Key Findings:

- The system effectively uses a relational database with clear relationships among entities like users, parking spots, parking lots, and reservations.
- Key functionalities include user and admin sign-ins, parking spot reservations, and automatic updates to parking spot statuses based on reservation timings.
- The system includes crucial administrative controls, such as viewing lot profiles, managing users, and handling debt for missed reservations.

Innovative Features Added:

- **Reservation Management**: Real-time tracking and status updates for parking spots, ensuring users are informed of availability and reservations are handled correctly.
- **Trigger-Based Updates**: The use of triggers ensures the parking spot status updates automatically when a reservation starts or ends, reducing manual intervention.

- **Debt Handling**: Automatic debt assignment for users who fail to arrive for a reservation, ensuring fairness and accountability in the system.
- **Concurrency Control**: The reservation system includes a transactional approach for spot reservations, preventing double bookings and ensuring data integrity.

In conclusion, this system is designed to offer a user-friendly interface, secure data handling, and efficient parking lot management, with innovative features that ensure its practicality and effectiveness in a smart city environment.