**Team 23**

**Parking Management System**

**Abstract:**

The Parking Management System is an application designed to manage the records of incoming and outgoing vehicles in a parking facility. This system tracks the entry and exit of cars, maintains a listing of vehicles within the parking lot, and calculates the cost per vehicle based on duration of stay. Implemented in Java, the system leverages the language's versatility and compatibility with various databases, including MySQL. This enables seamless integration with Database Management Systems, ensuring the secure and efficient storage of parking-related data. Its user-friendly interface and adaptable design make parking hassle-free for both administrators and users, enhancing convenience and efficiency.

**Introduction:**

Inefficient management of parking facilities poses challenges in tracking vehicle entry and exit, as well as monitoring parking lot occupancy. The lack of an efficient parking management system exacerbates these issues, resulting in errors, inconvenience in spot reservations, and suboptimal utilization of parking space.

The Parking Management System addresses these shortcomings by offering a sophisticated application designed to streamline the management of vehicle records within parking facilities. Its primary objective is to efficiently track vehicle entry and exit, monitor parking lot occupancy, and calculate parking fees based on the duration of vehicle stays. Key components of the system include the Parking Lot entity, which represents individual parking lots and stores details such as location, capacity, and current occupancy. The Entry/Exit entity records the entry and exit times of vehicles, while the Fee entity manages parking fees, including total charges and entry/exit timestamps. Additionally, the Occupancy entity tracks the number of vehicles in the parking lot over time. To facilitate seamless payment processing, the Payment entity stores payment information, including the payment amount and timestamp.

Overall, the Parking Management System offers a comprehensive solution for efficiently managing parking facilities, enhancing convenience for both administrators and users while optimizing parking space utilization and revenue generation.

**Functional Requirements:**

Parking Lot: Represents individual parking lots with details about their capacity and current occupancy. It uses location to indicate where the parking lot is situated, capacity to specify the maximum number of vehicles the lot can accommodate, and current occupancy to track the number of vehicles currently parked.

Entry/Exit: Records the entry and exit times of vehicles. It utilizes timestamps to capture the date and time when a vehicle enters or exits, along with an entry or exit flag to indicate the type of record.

Fee: Stores information about parking fees, including the total fee charged for parking and timestamps for when the vehicle entered and exited.

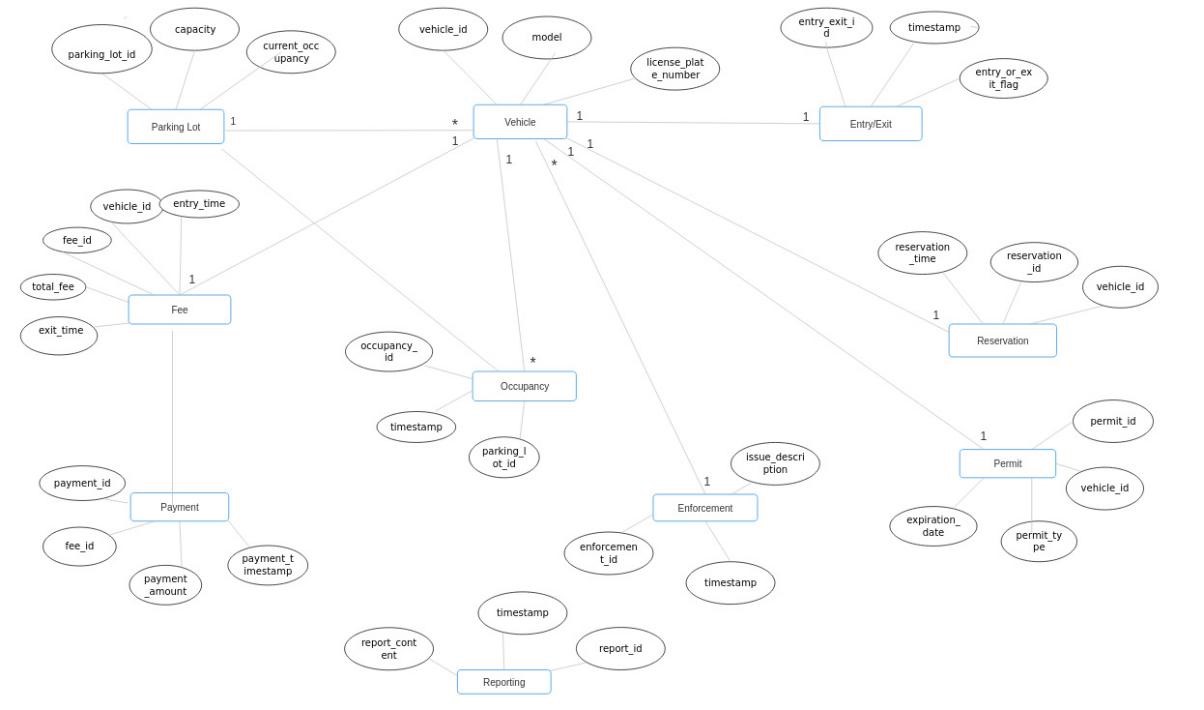
Reservations: Allows users to reserve parking spaces for specified durations. It records the reservation time when the reservation was made and the reserved until time, indicating the expiration time for the reservation.

Occupancy: Tracks the number of vehicles in the parking lot over time. It uses timestamps to record the time when the occupancy was measured and an occupancy count to specify the number of vehicles parked at that time.

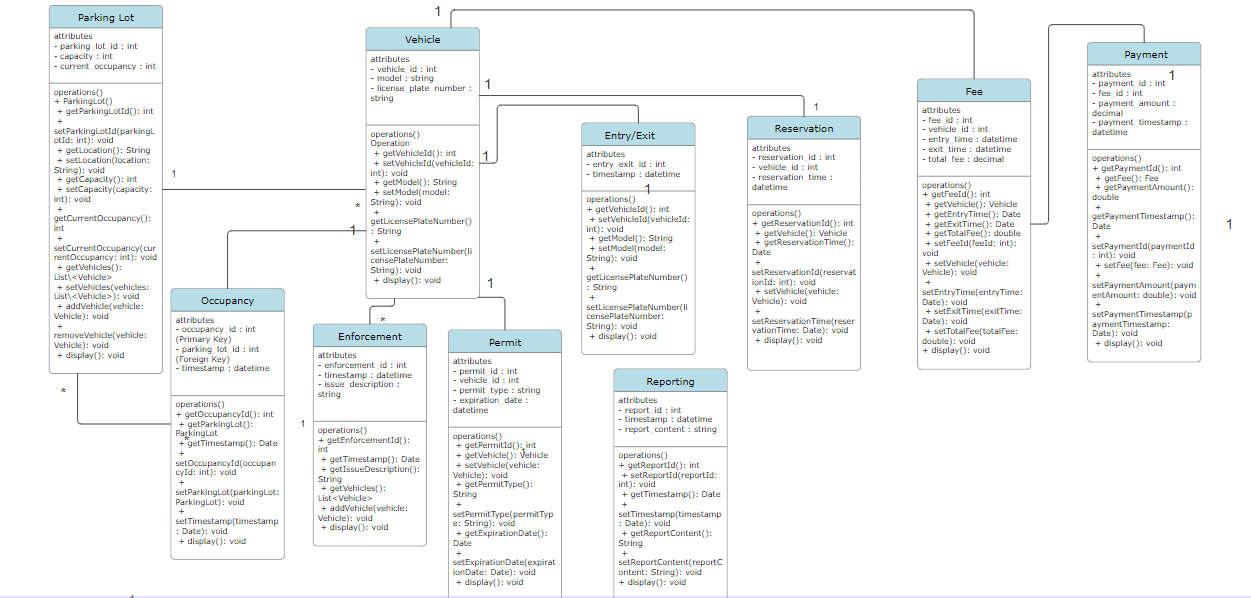
Payment: Stores payment information for parking fees, including the payment amount and timestamp. It ensures accurate recording of the amount paid for parking and the time when the payment was made.

Permit Type: Specifies the type of permit, such as a monthly pass or handicapped permit. It includes an expiration date to indicate when the permit expires, providing users with information about the validity of their permits.

**ER Diagram:**



**UML Diagram:**



**Queries to create Database:**

create database pm\_system;

use pm\_system;

CREATE TABLE parking\_lot (

parking\_lot\_id INT(20) PRIMARY KEY,

capacity INT(20),

current\_occupancy VARCHAR(20));

CREATE TABLE vehicle (

vehicle\_id VARCHAR(20) PRIMARY KEY,

model VARCHAR(50),

license\_plate\_number VARCHAR(20));

CREATE TABLE entry\_exit (

entry\_exit\_id VARCHAR(20) PRIMARY KEY,

timestamps TIMESTAMP,

entry\_or\_exit\_flag VARCHAR(20));

CREATE TABLE fee (

fee\_id VARCHAR(50) PRIMARY KEY,

vehicle\_id VARCHAR(50),

entry\_time VARCHAR(50),

exit\_time VARCHAR(50),

total\_fee VARCHAR(50),

FOREIGN KEY (vehicle\_id) REFERENCES vehicle(vehicle\_id));

create table occupancy(

occupancy\_id varchar(50) primary key,

timestamps timestamp,

parking\_lot\_id int(50),

foreign key(parking\_lot\_id) references parking\_lot(parking\_lot\_id));

create table reservation(

reservation\_time time,

reservation\_id varchar(50) primary key,

vehicle\_id varchar(50),

foreign key(vehicle\_id) references vehicle(vehicle\_id));

CREATE TABLE payment (

payment\_id VARCHAR(50) PRIMARY KEY,

fee\_id VARCHAR(50),

payment\_amount INT(50),

payment\_timestamp TIMESTAMP,

FOREIGN KEY (fee\_id) REFERENCES fee(fee\_id));

CREATE TABLE reporting (

report\_content VARCHAR(50),

timestamps TIMESTAMP,

report\_id VARCHAR(50) PRIMARY KEY);

CREATE TABLE enforcement (

enforcement\_id VARCHAR(50) PRIMARY KEY,

timestamps TIMESTAMP,

issue\_description VARCHAR(50)

);

CREATE TABLE permit (

permit\_id VARCHAR(50) PRIMARY KEY,

vehicle\_id VARCHAR(50),

permit\_type VARCHAR(50),

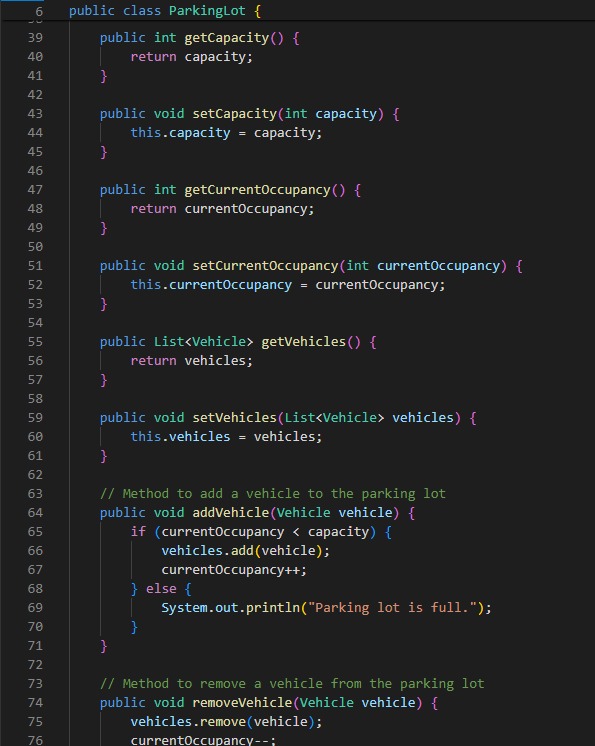
expiration\_date DATE,

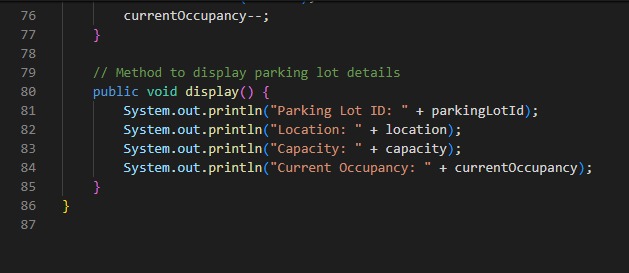
FOREIGN KEY(vehicle\_id) REFERENCES vehicle(vehicle\_id)

);

**Java Code:**

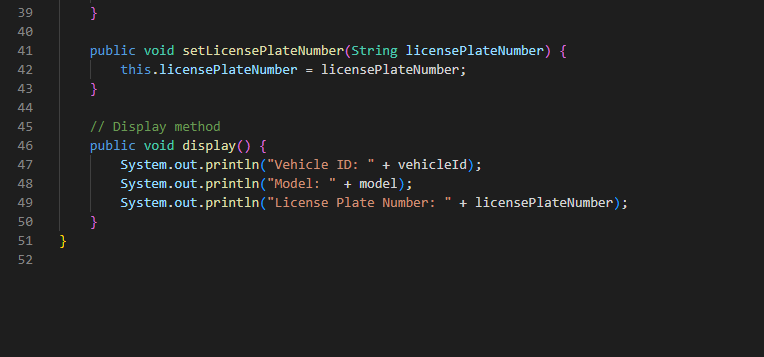
ParkingLot.java



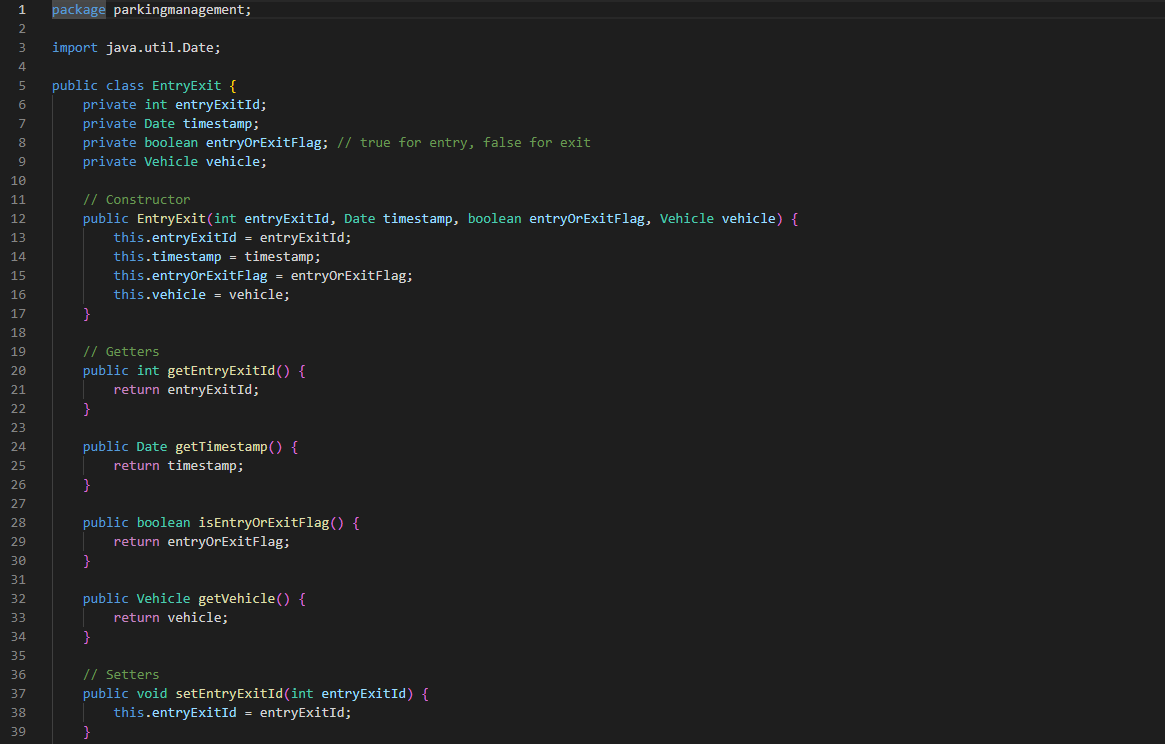


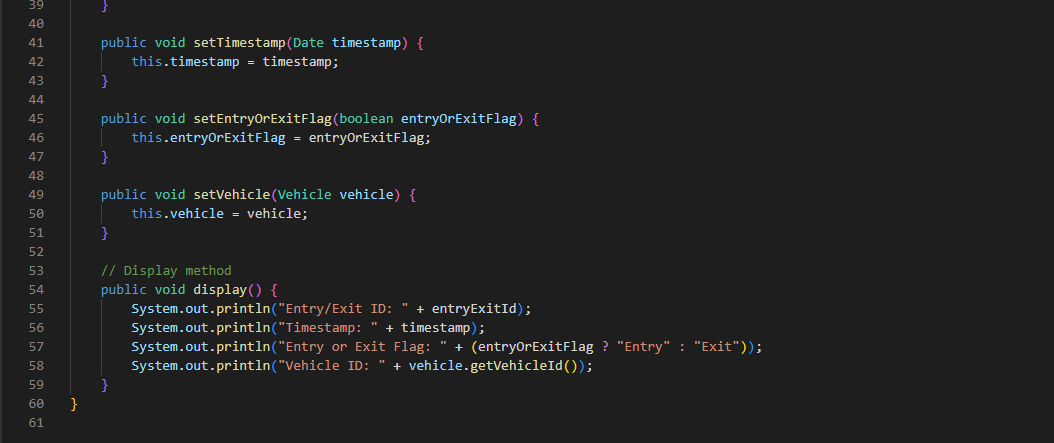
Vehicle.java



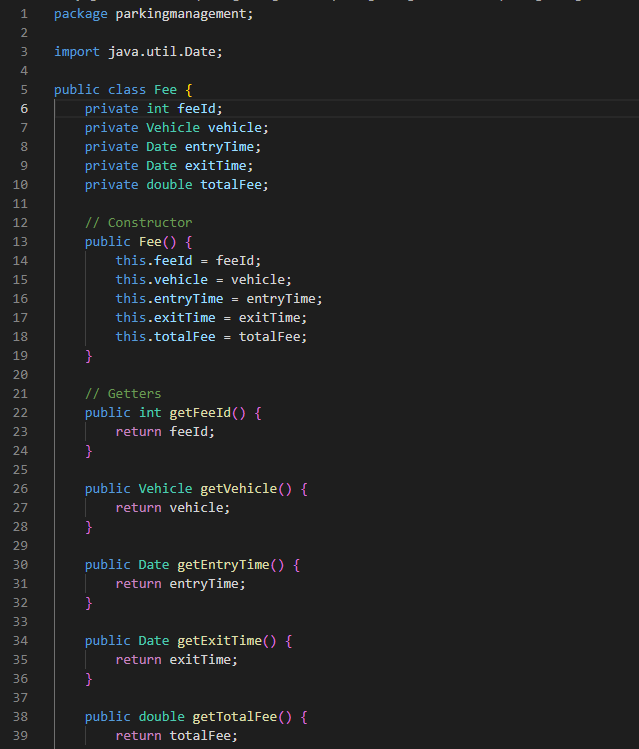


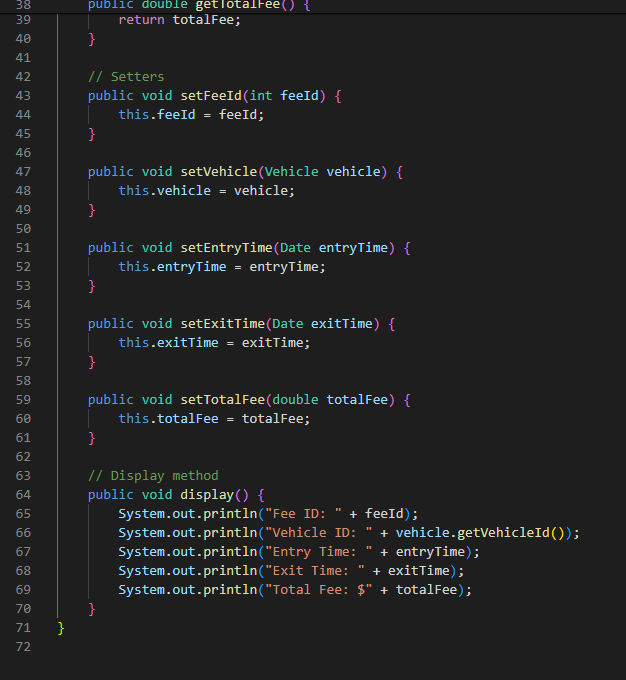
EntryExit.java



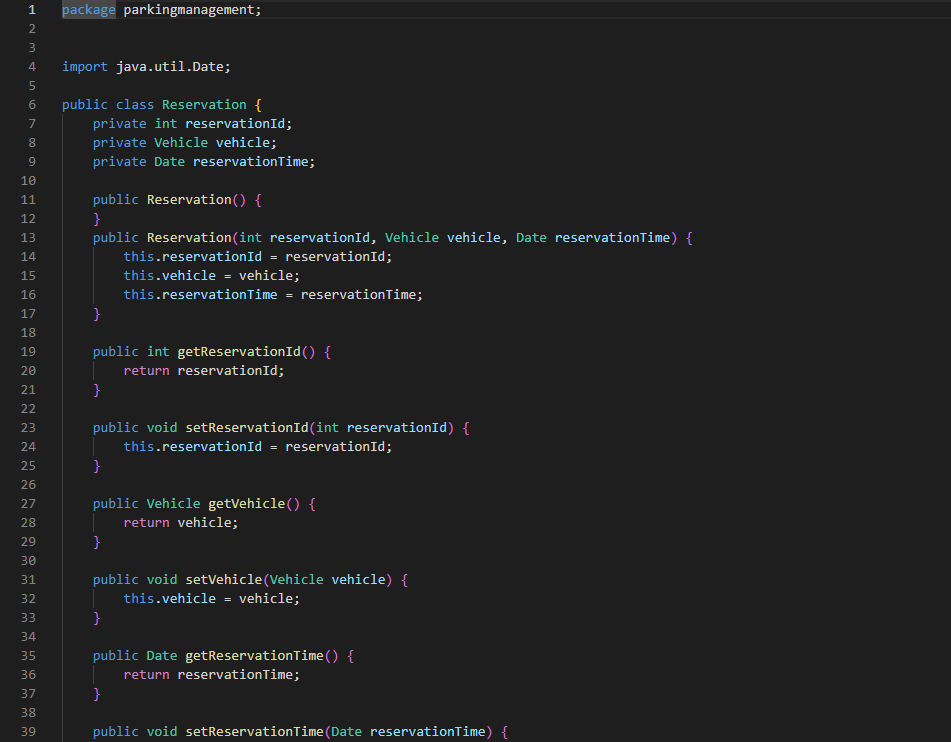


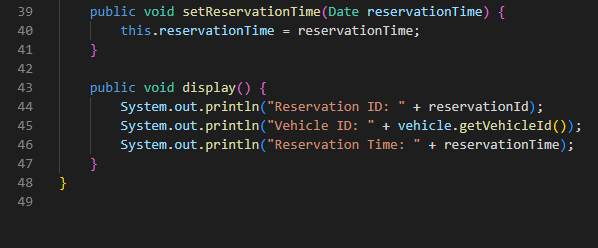
Fee.java



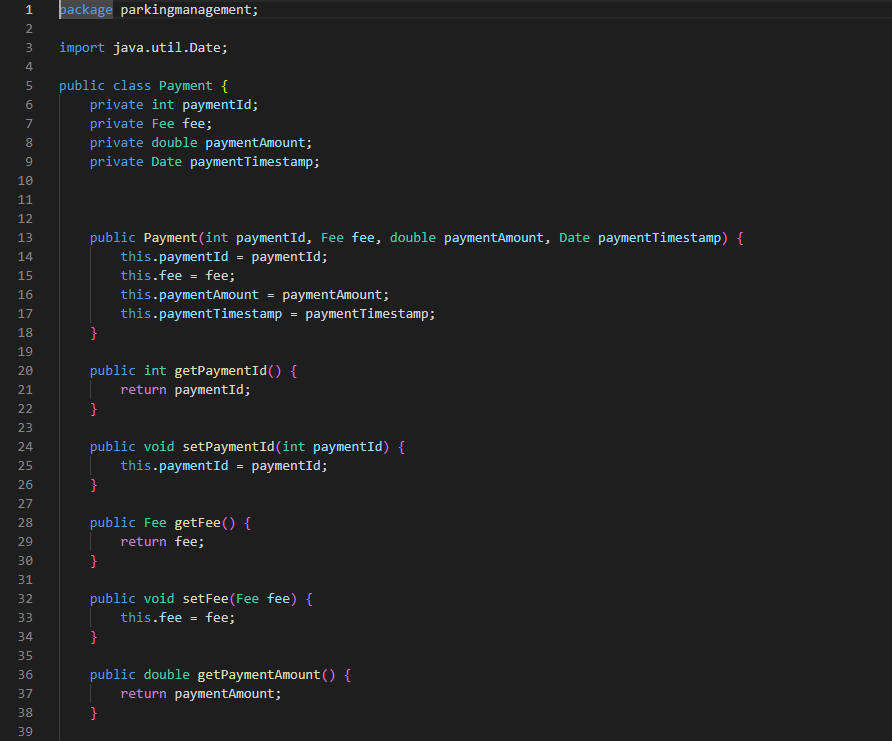


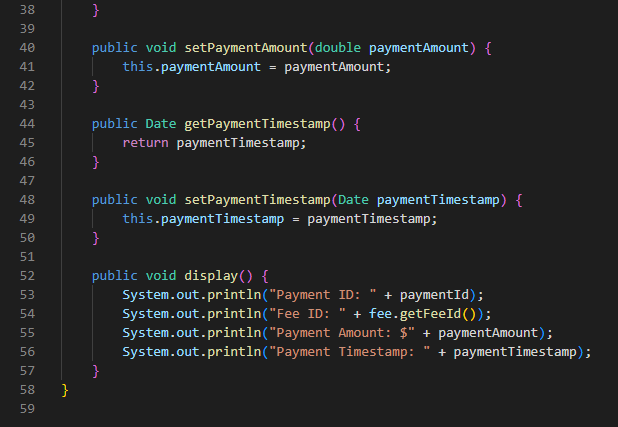
Reservation.java



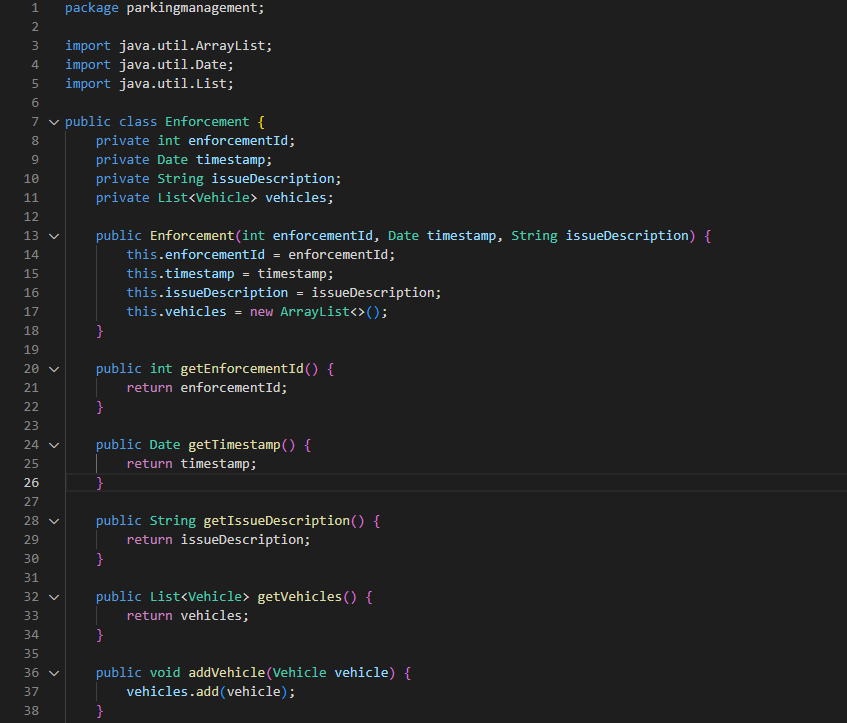


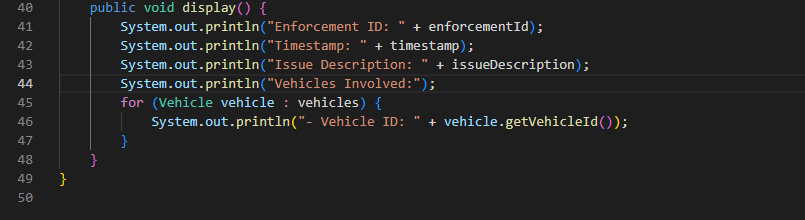
Payment.java



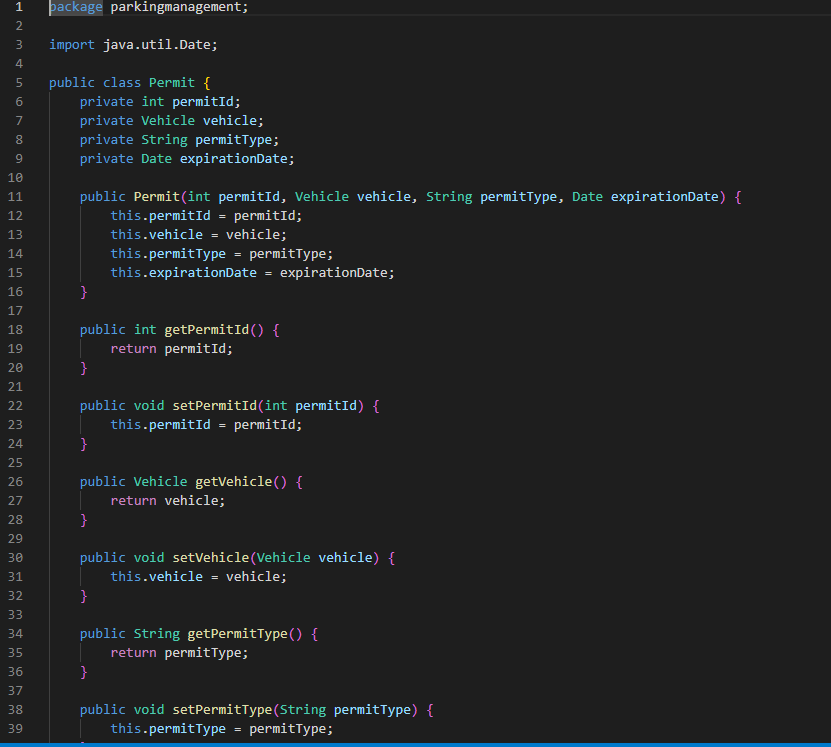


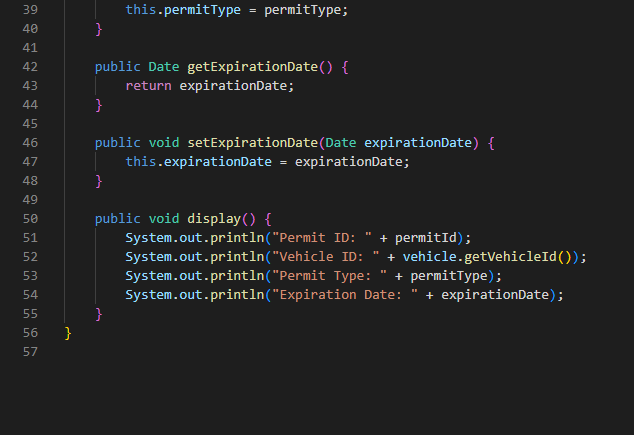
Enforcement.java



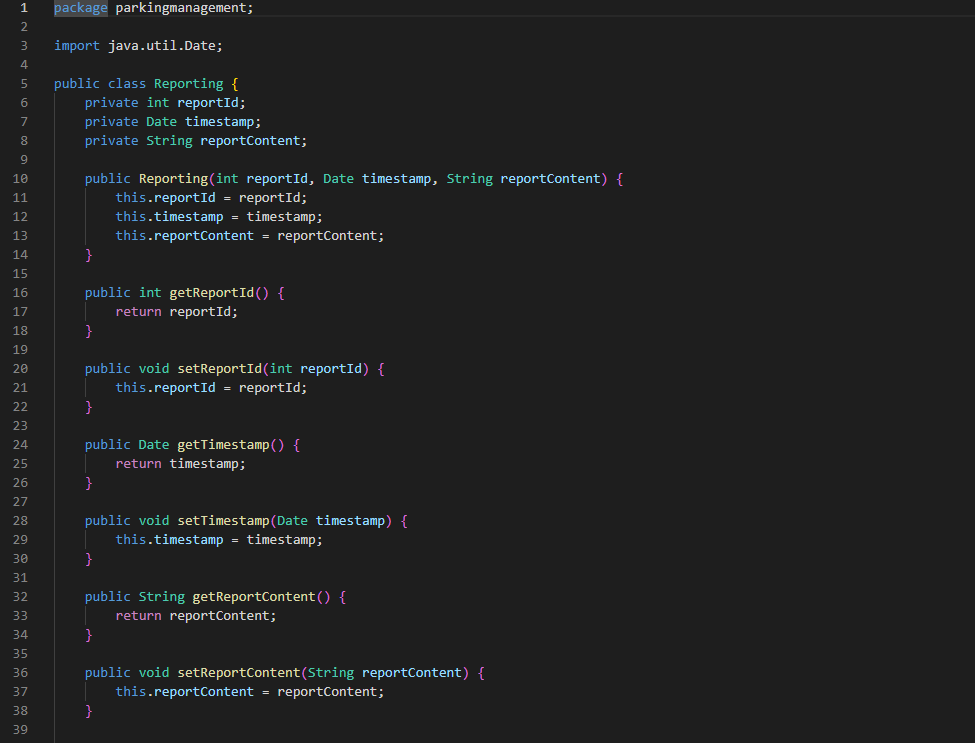


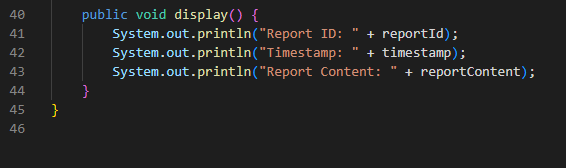
Permit.java





Reporting.java





**Challenges Faced in the Project:**

1. Integration: Combining different system modules like entry/exit recording, fee calculation, and occupancy tracking seamlessly can be complex.

2. Scalability: Ensuring the system can handle a growing number of vehicles and parking lots without performance issues is challenging.

3. Real-Time Processing: Processing entry/exit data and fee calculations in real-time while maintaining accuracy poses technical challenges.

4. Maintenance: Ensuring ongoing system maintenance and updates to address issues and enhance functionality is a continuous challenge.