CS 5200: Database Management Systems

Project: Parking Management System

Group name: RajaniYPatelD

Group Members:

- Yug Deepak Rajani
- Dhruv Patel

Presentation: Completed on 19 April 2023 2:50 PM

Introduction

The objective of this project is to design and implement a robust database management system for the Parking Management System that will support its day-to-day operations. The database should be designed to store and manage all parking-related data, including customer information, vehicle details, parking space availability, parking history, and billing records. It will be an automated parking management system where users will be able to enter parking lots with or without any prior reservation and use the available spaces of the parking lot.

Database Overview

The parking management system is a software application that efficiently manages parking spaces for vehicles and related data such as customers, staff, reservations, transactions, pricing, and fines. The system keeps track of parking space availability, ensures smooth entry and exit of vehicles, tracks parking history, and manages pricing and transactions.

The system consists of several entities, each of which stores data related to different aspects of the parking management process. For example, the Parking Lot entity stores information about the parking lot such as its name, address, and total capacity. The Vehicle entity stores information about each vehicle parked in the parking lot, such as its registration number, vehicle type, and parking space allotted.

The Parking Event entity stores information about each parking event, including the entry and exit time of the vehicle and the parking space number. The Customer entity stores information about each customer, such as their name, phone number, and address. The Staff entity stores information about staff members, such as their name, role, and address.

The Reservation entity stores information about the reservations made by customers, such as the date and time of the reservation. The Transaction entity stores information about the transactions made by customers, such as the amount paid, mode of payment, and date and time of the transaction. The Pricing entity stores information about the pricing of the parking lot, such as hourly, daily, and monthly rates and the type of vehicle.

The Fine entity stores information about any fines imposed on a vehicle, such as the reason for the fine and the amount charged. Overall, the parking management system provides an efficient and organized approach to managing parking spaces, vehicles, and customers while ensuring a hassle-free experience for all.

Entities:

1. Parking Lot
2. Parking Space
3. Vehicle

4. Parking Event
5. Customer
6. Staff (worker and admin)
7. Reservation
8. Transaction
9. Pricing
10. Fine

Technical Specifications

CLI based project

Parking Management System is a database-centric project that provides a platform for managing parking lots and spaces, vehicle information, customer bookings and transactions, staff and administrative roles. It enables efficient and streamlined operations for parking management through a command-line interface (CLI) for ease of use and quick access to data.

SQL vs. NO SQL storage

SQL databases are typically the preferred choice for parking management systems due to their strong support for complex relationships, transactions, and data consistency. NoSQL databases may be more suitable for applications such as e-commerce which require more flexibility and scalability but are less suitable for applications that require strong data consistency and complex relationships between entities.

Software: MySQL Workbench

Languages:

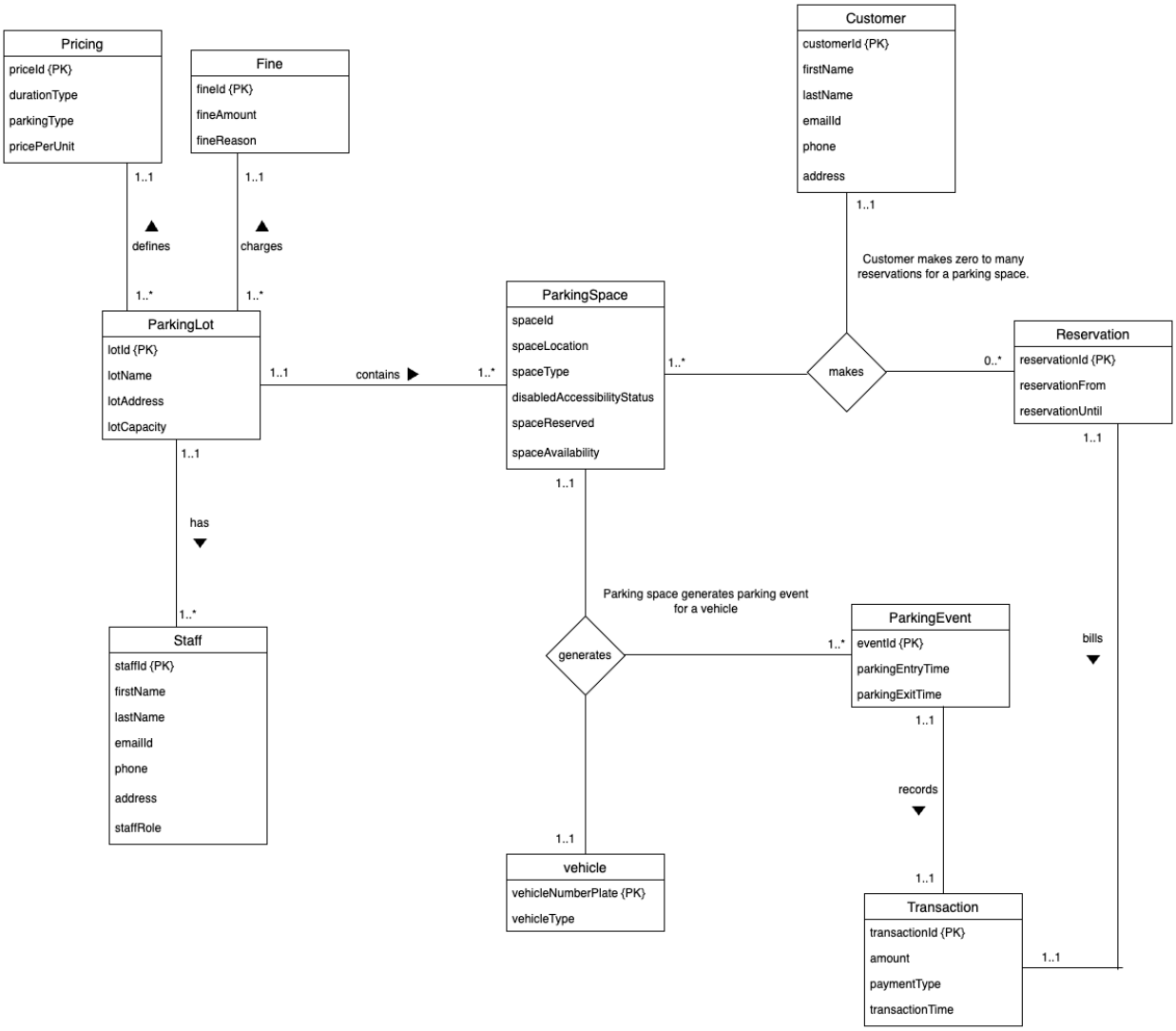
- Python (pymysql for database connectivity)
- SQL for querying

Why does this project interest us?

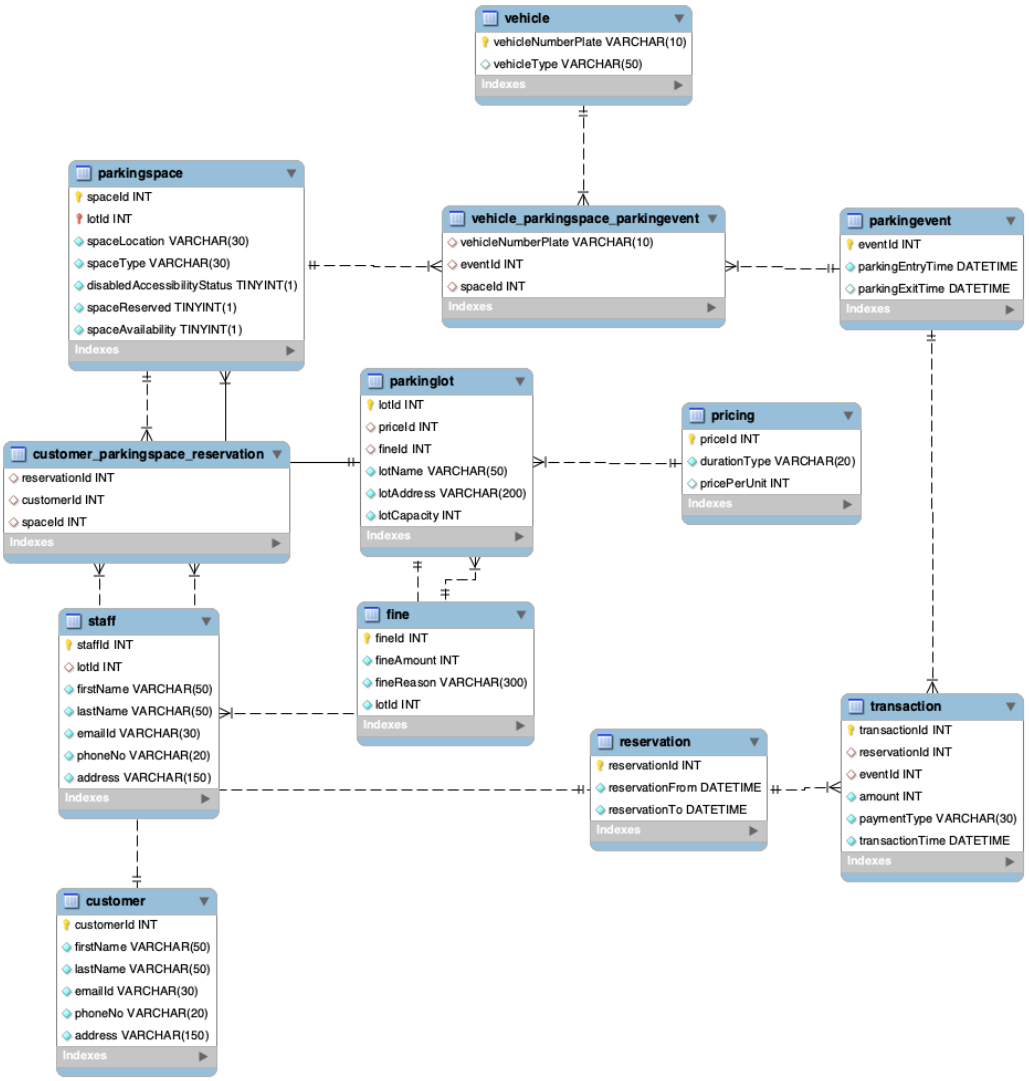
A parking management system can generate valuable insights and analytics that can help parking lot operators optimize their operations, improve customer experience, and increase revenue. And looking at the current scenario of parking problems, an efficient parking management system can help to analyze where and how much space is required. This motivates us to work on this project and develop an efficient system which can reduce the parking problems.

Overall, the parking management system is an interesting database project due to its potential to solve real-world problems related to parking, its complexity, and the valuable insights and analytics it can generate.

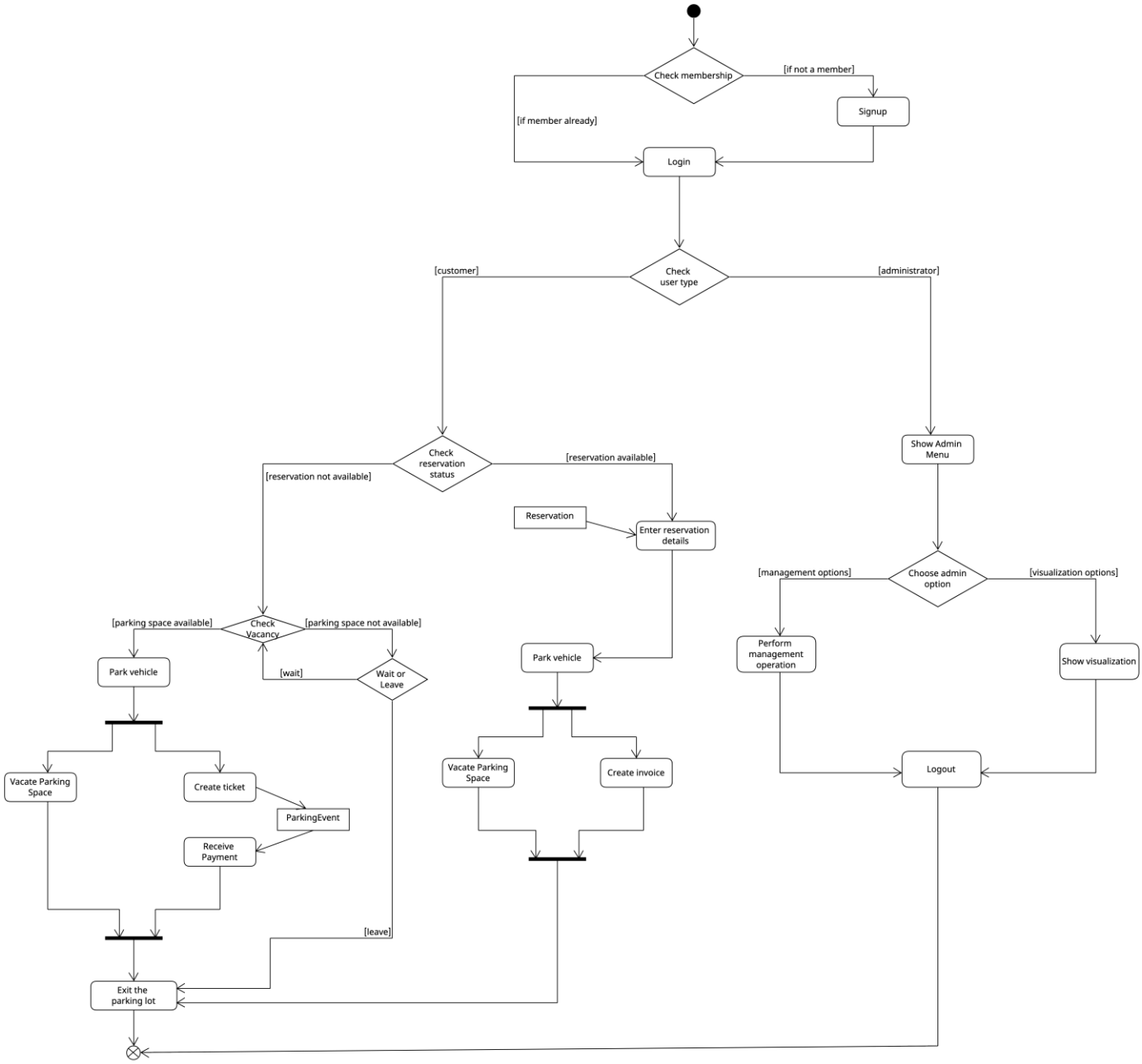
Conceptual Design (UML Diagram)



Logical Design (Reverse Engineered Schema)



Activity Diagram



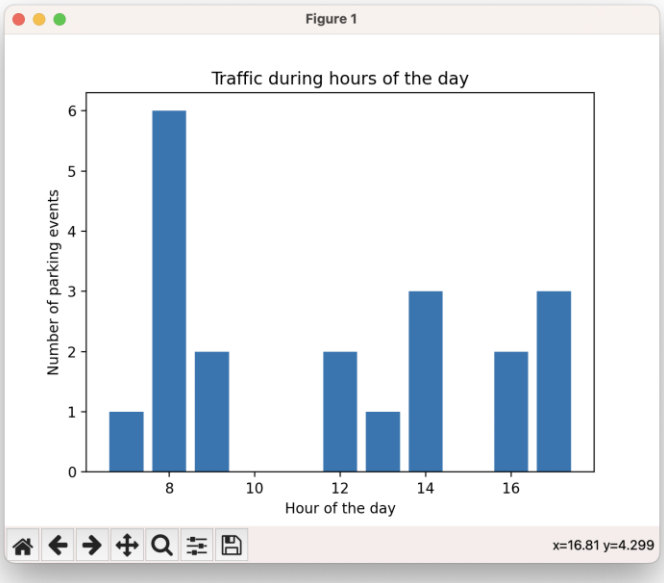
Final user flow of the system (User activity description)

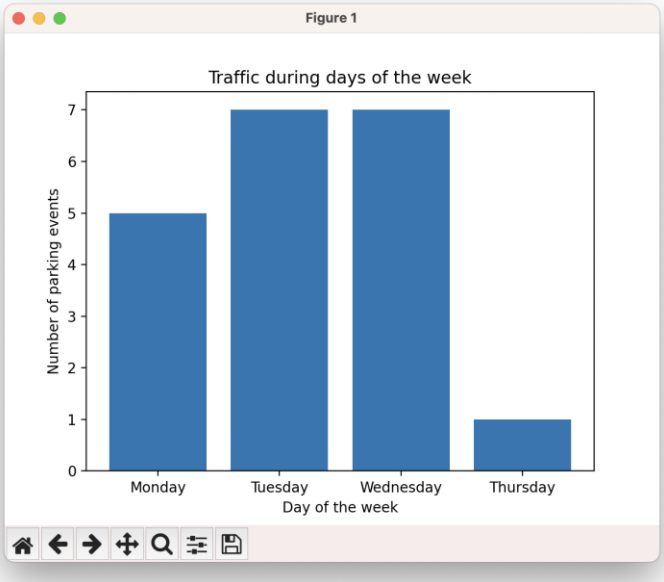
The details of the installation, setting up and running the system locally have been provided in the README.md file.

Here is a user flow of the system:

1. User runs the CLI application.
2. There are two types of users with different roles for the system: **Customer** and **Administrator**.
3. There are two basic options for the end user: Login and Signup
4. CLI prompts the user to enter their email address or phone number to verify their account when logging in.
5. Once the email address is verified, CLI prompts the user with the possible options based on the role of the user.
6. If the user is a customer, then he/she can claim their reserved parking spot or can get an unreserved parking spot (if available).
7. CLI checks for available parking spots and prompts the user to indicate if they require a disability accessible spot.
8. Based on the user's response, CLI finds and reserves a parking spot for the user and provides them with a parking event ID and parking spot number.
9. The user parks their vehicle in the designated parking spot and goes about their business.
10. When the user returns to their vehicle, they enter their parking event ID into the CLI application to indicate that they are leaving.
11. CLI calculates the parking duration and transaction cost based on the parking rates and prompts the user to make the payment.
12. Once the payment is made, the user exits the parking lot, and the parking space is marked as available in the system.
13. If the administrator (staff member) logs in, the relevant options for management of parking lots, parking spaces, and generation of bar charts and summary for getting useful insights from the data are shown.
14. The administrator is allowed to perform the administrative operations.

Here are some of the screenshots of the application generated insights:






```
Summarize earnings: Transactions summary
Yesterday's total: 250
Last week's total: 600
Last month's total: 620
```

Lessons Learned

Technical expertise gained

During the development of this project, we gained a lot of technical expertise. We learned how to use Python to develop software applications and how to connect to a MySQL database to store and manage data. Additionally, we gained experience in using libraries like curses, matplotlib, and pymysql to simplify tasks and make user interactive command line applications. Overall, we feel confident in my abilities to develop software applications using Python and connect them to databases. We learned the importance of designing the conceptual database design for the given problem statement before starting off with the development and the implementation. Moreover, we got to know when using relational databases is advantageous over using NoSQL databases in specific use cases.

Insights, time management insights, data domain insights etc.

In terms of insights, we learned the importance of time management and how it can impact the success of a project. We found that setting realistic deadlines and prioritizing tasks helped me stay on track and complete the project on time. Additionally, we gained valuable data domain insights on the parking management industry, including the challenges faced by parking lot managers and the solutions provided by parking management systems.

Realized or contemplated alternative design / approaches to the project

Throughout the project, we realized that there were several alternative designs and approaches that could have been taken to achieve the same goal. For example, instead of using a MySQL database, we could have used a NoSQL database like MongoDB. Additionally, we could have used a web framework like Django instead of Flask, which may have provided more features out of the box with REST API endpoints. However, given the scope of the project, we feel that the technical decisions that we made throughout were the best fit for the project.

Code not functional

The function of sending OTP (One Time Password) to email or phone number is static for now. The One Time Password to log in for both, administrator and the customer is 012345 for now.

Similarly, there is no way of accepting payments as of now in the functionality of generating receipts for exiting the parking lot.

Both these features have not been implemented because they are out of scope of the project requirements.

Future work

While the parking management system is functional and meets the requirements specified, there are some areas where it could be improved and expanded in the future.

Planned uses of the database

The parking management system can be used to generate detailed reports about the parking lot's usage, including information such as the number of vehicles parked per day, and the parking spots used most. This data can help in better managing the parking lot, optimizing resource utilization, and understanding customer behavior.

Potential areas for added functionality

The system can be integrated with license plate recognition technology to automate the parking space allocation process. This feature will reduce the time taken for vehicle entry and exit and minimize the need for manual intervention in the parking process.

The parking management system can be integrated with sensors installed at each parking space to detect the availability of the parking spot. This feature will enable real-time monitoring of the parking lot's occupancy, making it easier to manage parking space availability.