MINOR-PROJECT BATCH-3

VEHICLE PARKING MANAGEMENT SYSTEM

By: Aayushi Sharma

Project Overview:

This project is a console application made using Python and SQLite. This project contains functionalities to simplify the booking process of vehicle parking and helps in effective management of customer data. Customers can book a parking slot based on their vehicle type by getting the help of an attendant. Staff or attendant acts as an intermediator between customer and the management system.

Key Features:

This project offers below mentioned functionalities and features:

1. User authentication and role-based access as admin or attendant.
2. Interactive interface.
3. Maintaining confidentiality of user credentials through hashing.
4. Logging and monitoring.
5. Input validations using Regex.
6. Exception Handling.
7. Clean code with proper folder structure.
8. Classes following single responsibility principles.
9. Maintained different files for prompts and input statements showing uniformity.

Major Modules of Project:

1. **User Module:**

   User will be the one interacting with the system.

   Users can be 2 types:

   (i) Admin:

   - The very first user to login into the system will be admin.

- This system assumes that there will be only one active admin in the system.
- Admin will be the one responsible for adding other users or employees that will be assigned attendant role.
- This system can have only one active admin.
- Admin will have different functionalities than other staff.
- Admin will be responsible for parking slot details and managing them.
- Admins will also be responsible for managing employee details.

(ii) Attendant:

- Attendant will be responsible for registering customers.
- Customers will be interacting with the attendant to book a parking slot.
- Attendants will have all the rights to manage customer data and booking or vacating parking slots.

**** Important Points:**

- All the users will have a default-password during their first login. This password will be provided by the admin to the user and will be known to the user.
- After entering the default password, the user will have to change their password and this password will be stored as a hash and thus will be unknown to admin, maintaining **confidentiality** of credentials.
- Also, this password will follow all the rules of strong password generation.
- Also, every user will have only 3 login attempts, on failure of which user will be restricted from interaction with the system for some duration.

2. Vehicle Type:

- Admin will be responsible for adding vehicle types and managing the price per hour of the vehicles.
- Based upon this price, the customer will have to pay the amount according to the number of hours spent in parking slot.

3. **Parking Slot:**
    - Admin will be responsible for managing the parking slot details about which slot is for which vehicle type.
    - Attendant will be responsible for assigning and vacating of parking slots to customers.

Tech Stack:

Language: Python

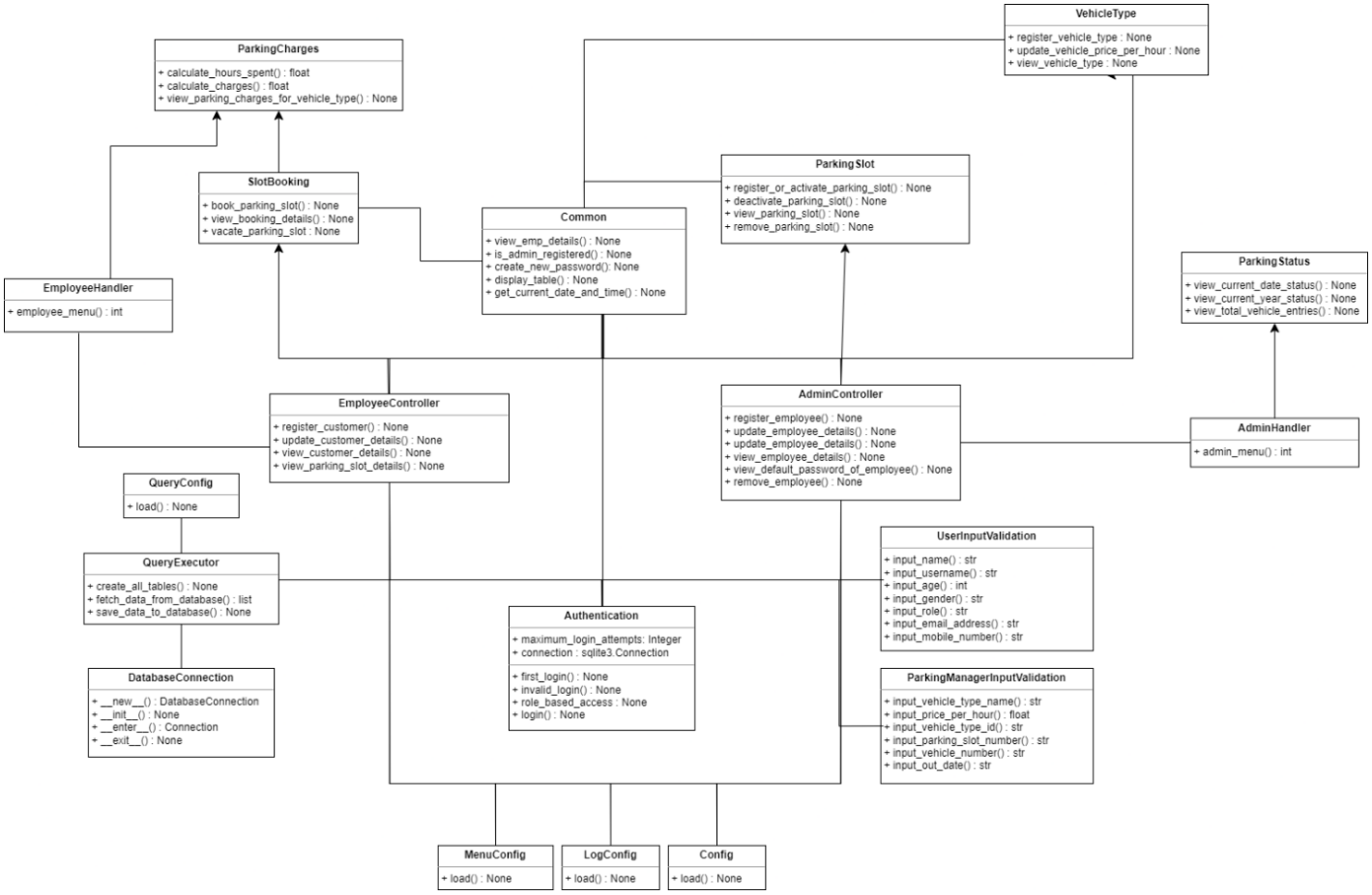
Database: SQLite

Integrated Development Environment: Visual Studio Code

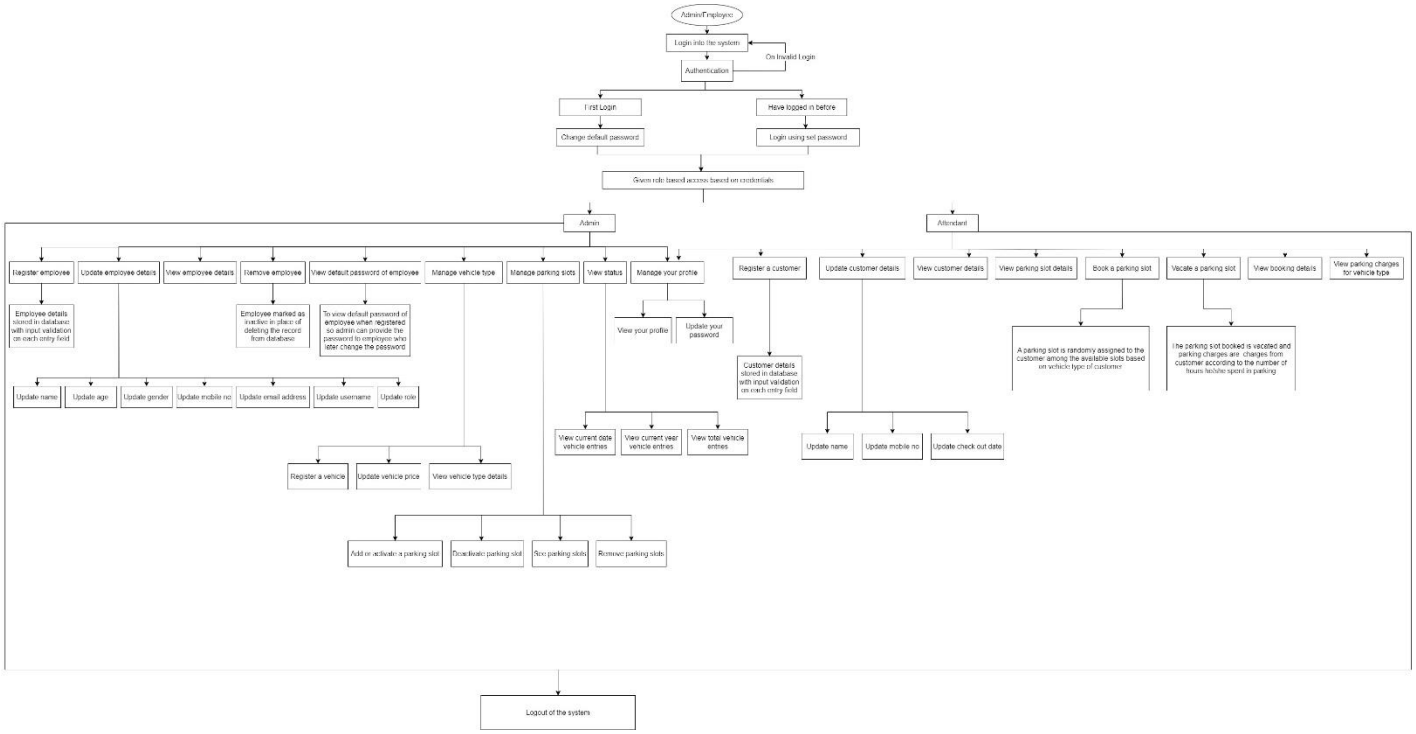
UML Diagrams:

1. Database Schema

2. Class Diagram



3. Flowchart


Admin/Employee
Login into the system
On Invalid Login
Authentication
First Login
Have logged in before
Change default password
Login using set password
Given role based access based on credentials
Admin
Attendant
Register employee
Update employee details
View employee details
Remove employee
View default password of employee
Manage vehicle type
Manage parking slots
View status
Manage your profile
Register a customer
Update customer details
View customer details
View parking slot details
Book a parking slot
Vacate a parking slot
View booking details
View parking charges for vehicle type
Employee details stored in database with input validation on each entry field
Employee marked as inactive in place of deleting the record from database
To view default password of employee when registered so admin can provide the password to employee who later change the password
View your profile
Update your password
Customer details stored in database with input validation on each entry field
A parking slot is randomly assigned to the customer among the available slots based on vehicle type of customer
The parking slot booked is vacated and parking charges are charges from customer according to the number of hours he/she spend in parking
Update name
Update age
Update gender
Update mobile no
Update email address
Update username
Update role
View current date vehicle entries
View current year vehicle entries
View total vehicle entries
Update name
Update mobile no
Update check out date
Register a vehicle
Update vehicle price
View vehicle type details
Add or activate a parking slot
Deactivate parking slot
See parking slots
Remove parking slots
Logout of the system

4. Use Case Diagram