2024W CSD 3464 5 - Project Report

Car Rental Management System

Team Satoshi

Arshdeep Singh - C0913266

Amisha Nakrani - C0903272

Aashika Vachhani - C0903269

Ghazala Anjum - C0905815

Problem Statement

A car rental management system serves several objectives, which are crucial for the efficient operation and success of a car rental business:

1. **Efficiency**: Automating processes such as booking, returns, and invoices streamlines operations, reducing manual errors and saving time.
2. **Fleet Optimization**: The system helps manage the rental fleet by tracking vehicle availability, scheduling maintenance, maximizing vehicle utilization, minimizing downtime, and maximizing revenue.
3. **Customer Experience**: With features like online booking, real-time availability, and self-service options, it enhances the overall customer experience, making it easier for customers to find and book vehicles conveniently.
4. **Financial Management**: It provides tools for monitoring the revenue generated by customers and different cars
5. **Business Growth**: Through reporting and analytics, it offers valuable insights for making informed decisions related to business expansion, marketing strategies, and customer segmentation, thus enabling the rental business to scale operations and adapt to market changes effectively.

The car rental management system has captured our interest because it addresses many real-world challenges companies and customers face. By leveraging technology, we can automate and streamline many day-to-day operations like adding a car, removing a car, adding customers, booking and returning cars, etc. In addition, we have added a maintenance dashboard that will help us manage the fleet. A reporting dashboard has also been built so that managers can monitor the revenue generated. All of these things combined will ensure smooth day-to-day operations and better growth of the company

System design

1. **Architecture**: The system follows a multi-layered architecture, based on the Model-View-Controller (MVC) pattern. It separates concerns into different layers: presentation, business logic, and data access.
2. **Layers**:
    - **Presentation Layer**: This layer handles user interactions and displays information to the user. It includes the menu services and dashboards.
    - **Business Logic Layer**: This layer contains the logic for processing user requests, applying business rules, and coordinating interactions between different components. It includes services like booking, finance, employee, and customer services.
    - **Data Access Layer**: Responsible for interacting with the database. It includes classes like DAOs (Data Access Objects) and services for retrieving and manipulating data from the database.
3. **Modules/Components**:
    - **Menu Services**: These services handle the display of menus to the user and process user input to trigger appropriate actions.
    - **Dashboard Services**: These services provide functionality related to different dashboards, such as the finance dashboard, fleet dashboard, etc.
    - **Data Access Services**: Services responsible for interacting with the database, including retrieving, inserting, updating, and deleting data.
    - **Entity Classes**: Classes representing entities in the system, such as Employee, Customer, Car, Booking, etc.
    - **Populator Classes**: These classes populate entity objects from database query results.
4. **Database Interaction**:
    - The system interacts with a database management system (SQL Server) to store and retrieve data.

- JDBC (Java Database Connectivity) is used for database access from Java code.
- Prepared statements execute parameterized SQL queries and improve performance.

5. **Error Handling**:
    - Error handling mechanisms, such as try-catch blocks, handle exceptions that may occur during database interactions or other operations.
    - Error messages may be logged or displayed to the user to provide feedback on any issues encountered.
6. **User Input Handling**:
    - User input is validated to ensure correctness and prevent unexpected behavior.
    - Scanner objects are commonly used to read user input from the command line.
7. **Security**:
    - Basic security measures are implemented, such as password hashing for storing sensitive information like user passwords.
    - Access control mechanisms are in place to restrict certain functionalities to authorized users (e.g., managers).

Home

Author: Ghazala Anjum

We will first ask the user for their username and password here and allow them to log in.

Implementation for login:

The login details for the customer or an employee will be stored in a database. To make the login process more secure, we will use password hashing.

Hashing: Instead of storing passwords directly, we use a special function to turn them into scrambled codes called hashes. It's like turning "password123" into something like "f5e6d7c8b9a0". Once hashed, it's nearly impossible to figure out the original password.

Salting: We add a unique random string (a salt) to each password before hashing it. This makes it even harder for attackers to crack passwords because even if two users have the same password, their hashed passwords will look different due to the unique salts.

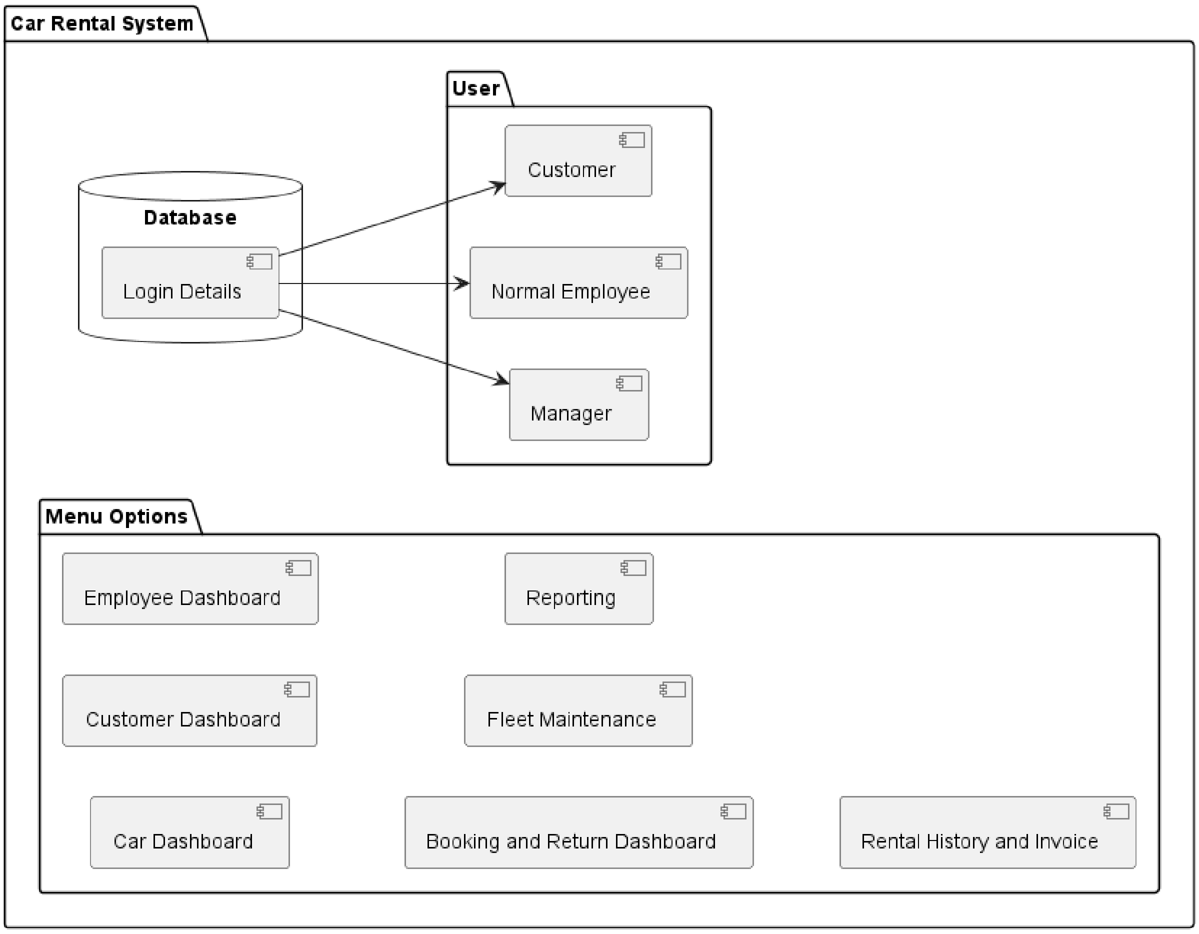
So, when a user logs in, we take their password, add the salt that we stored for them, hash the result, and compare it with the stored hashed password. If they match, the login is successful. This way, even if someone gets into our system, they can't easily figure out the passwords.

Depending on the user type, different menu options will be shown to the customer.

The user types are as follows:

- Customer: When the customer logs in, he will only be able to access the following options
    - Car Dashboard - The customer should only be able to search and view cars. Functionalities like adding, removing, and editing a car should not be available to the customer.
    - Customer Dashboard - The customer should only be able to edit their details here. Other functionalities, like adding other customers, should not be available to the customer.

    - Booking and Return Dashboard - Customer should be able to book and return a car
    - Rental history and Invoice - Customer should be able to view their rental history and should be able to download the invoice for each booking
- Normal Employee: He will be able to do the following
    - Car dashboard - The employee will be able to see all the details here, like Add Car, Remove Car, Edit car info, search for cars, and display cars
    - Customer dashboard - An employee will have full access to add, delete, edit, search, and display customers
    - Employee dashboard - Should only be able to search and display employees and edit their details
    - Booking and Return Dashboard - Employee should be able to book and return a car for the customer
    - Fleet Maintenance - Employee should be able to edit the data here
    - Rental history and invoice - Employee should be able to see the rental history of the customer and download their invoice
- Manager: He will be able to do the following
    - Car dashboard - Complete access to all features
    - Customer dashboard - Complete access to all features
    - Employee dashboard - Complete access to all features
    - Booking and return dashboard - Complete access to all features
    - Fleet Maintenance - Complete access to all features
    - Reporting - Complete access to all features
    - Rental History and Invoice - Complete access to all features


Car Rental System
User
Customer
Normal Employee
Manager
Database
Login Details
Menu Options
Employee Dashboard
Reporting
Customer Dashboard
Fleet Maintenance
Car Dashboard
Booking and Return Dashboard
Rental History and Invoice

Car Dashboard

Author: Aashika Vachhani

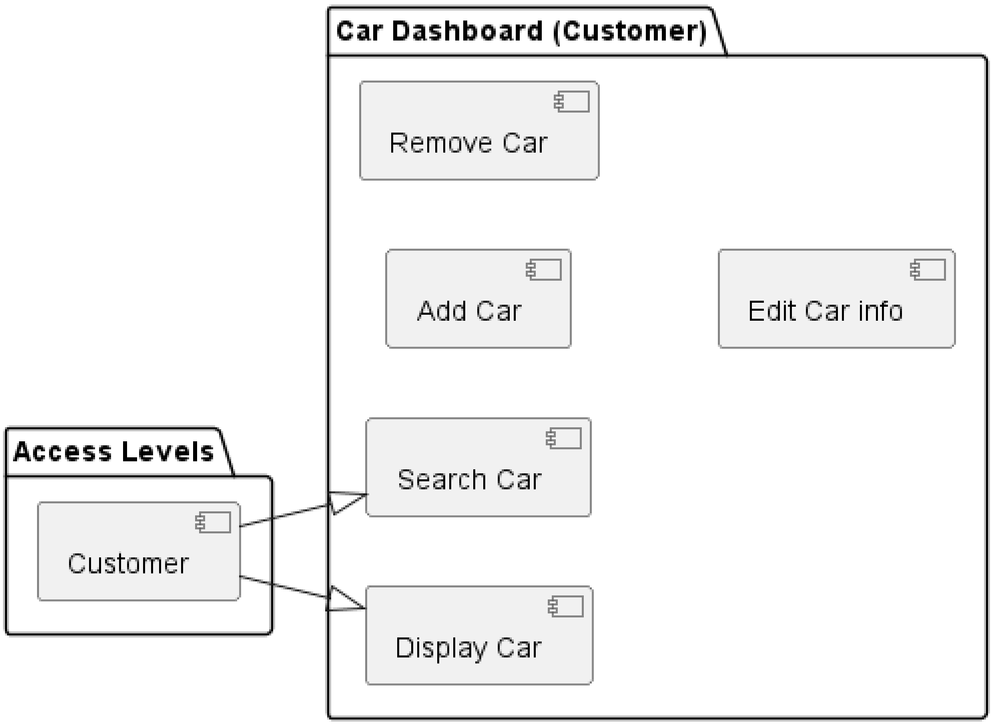
The car dashboard will have the following:

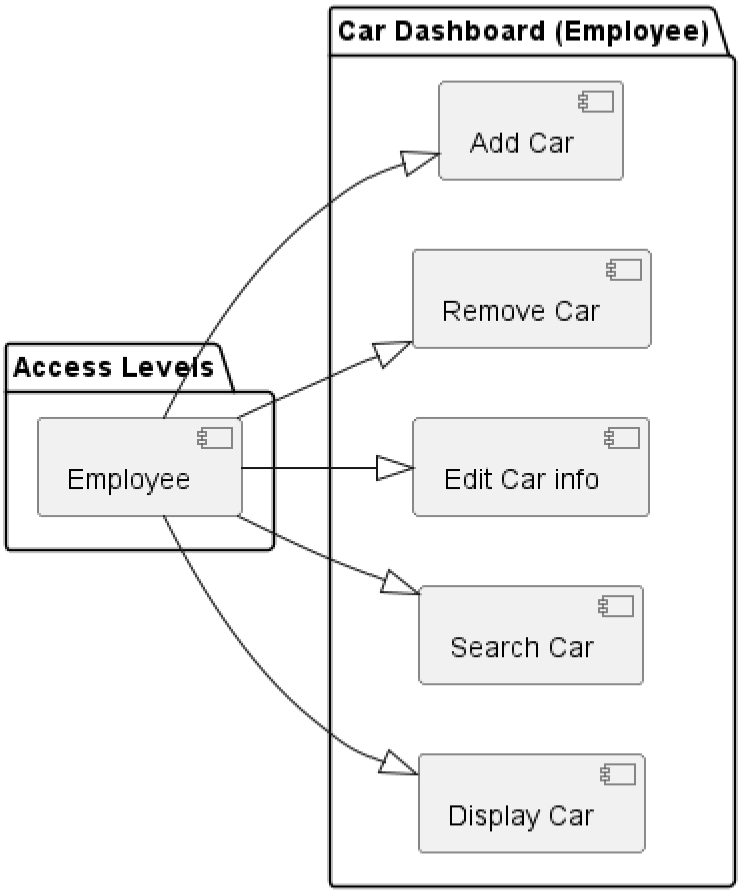
1. Add Car - This feature is available only to employees and managers. Whenever the company buys a new car, any employee or manager should be able to add it to the list of cars that the company owns.
2. Remove Car - This feature is available only to employees and managers. Whenever the company needs to let go of a car, an employee or manager should be able to remove it. However, we must check that the car is not currently rented to a customer. If it is removed, we set the isRemoved attribute to true.
3. Edit Car info - This feature is available only to employees and managers. Various details of a car should be editable.
4. Search Car - This feature is available to everyone. One should be able to search a car using its ID, model, and other features.
5. Display Car - This feature is available to everyone. This shows the list of all the cars the company owns.

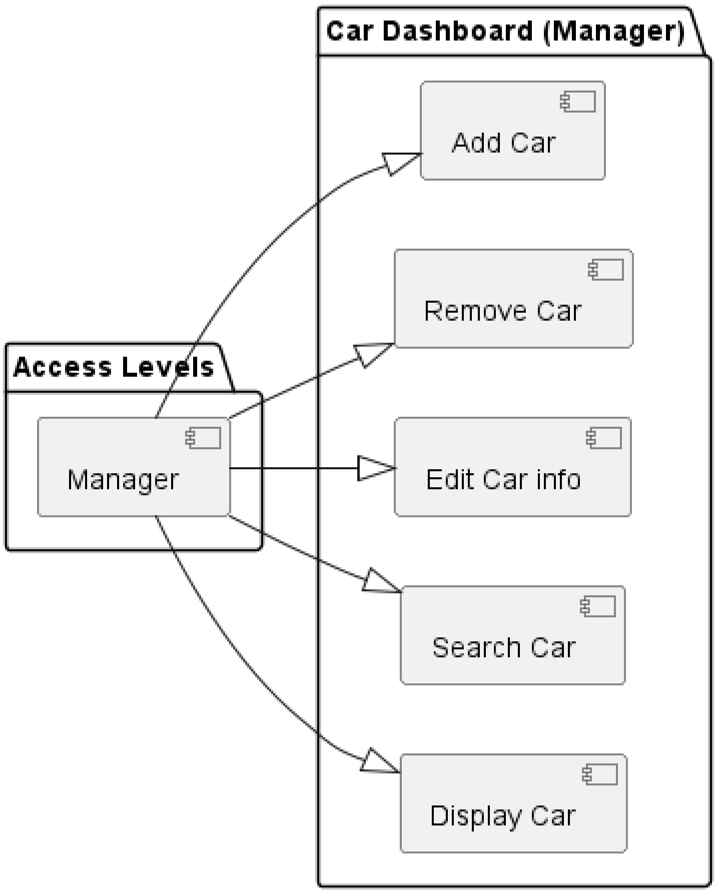
Car Attributes:

Attribute	**Mandatory(Y/N)**	**Data type**	**Description**
carId	Y	string	Stores the car id
brand	Y	string	Stores the brand
model	Y	string	Stores the model of the car
color	Y	string	Stores the color of the car
noOfPassengers	Y	~~string~~ integer	Stores the number of people that can be accommodated in the car
fuelType	Y	enum	can be either Petrol or Diesel
pruchaseDate	Y	Date	date when the car was purchased by the company

transmissionType	Y	enum	can either be automatic or manual
isAvailable	Y	boolean	If true, the car is available for rent
carPrice	Y	double	The price of the car
rentalPrice	Y	double	Cost of renting the car per hour
isRemoved	Y	boolean	The default value is false. If it is set to true, the car has been removed




Car Dashboard (Employee)
Add Car
Remove Car
Edit Car info
Search Car
Display Car
Access Levels
Employee


Car Dashboard (Manager)
Add Car
Remove Car
Edit Car info
Search Car
Display Car
Access Levels
Manager

Customer Data Dashboard

Author: Amisha Nakrani

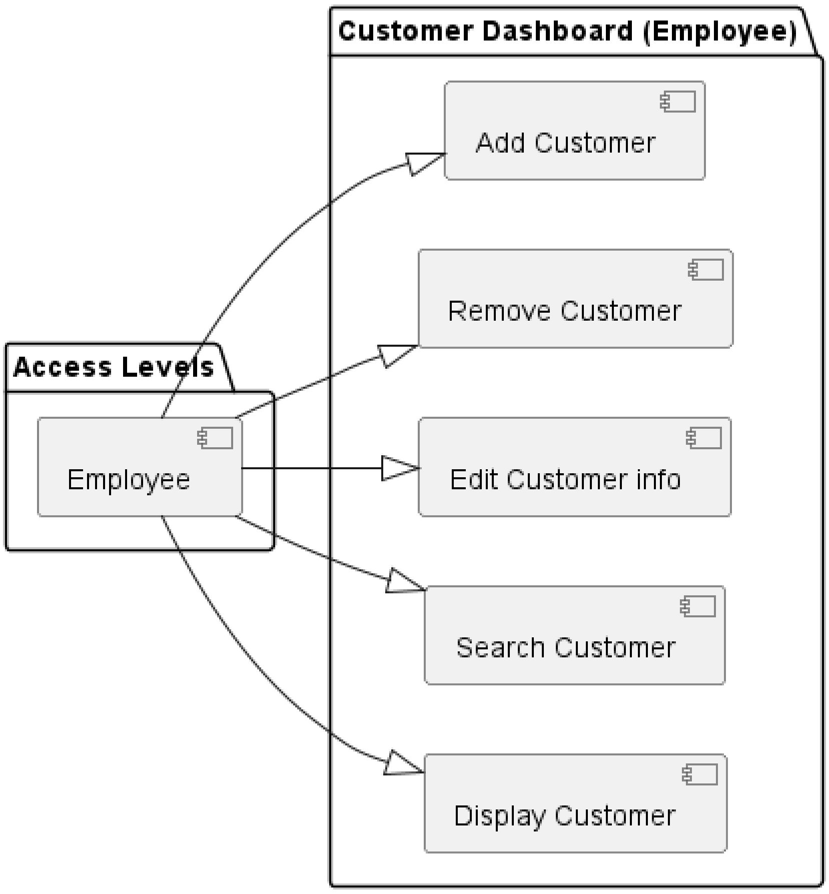
The customer dashboard will have the following features

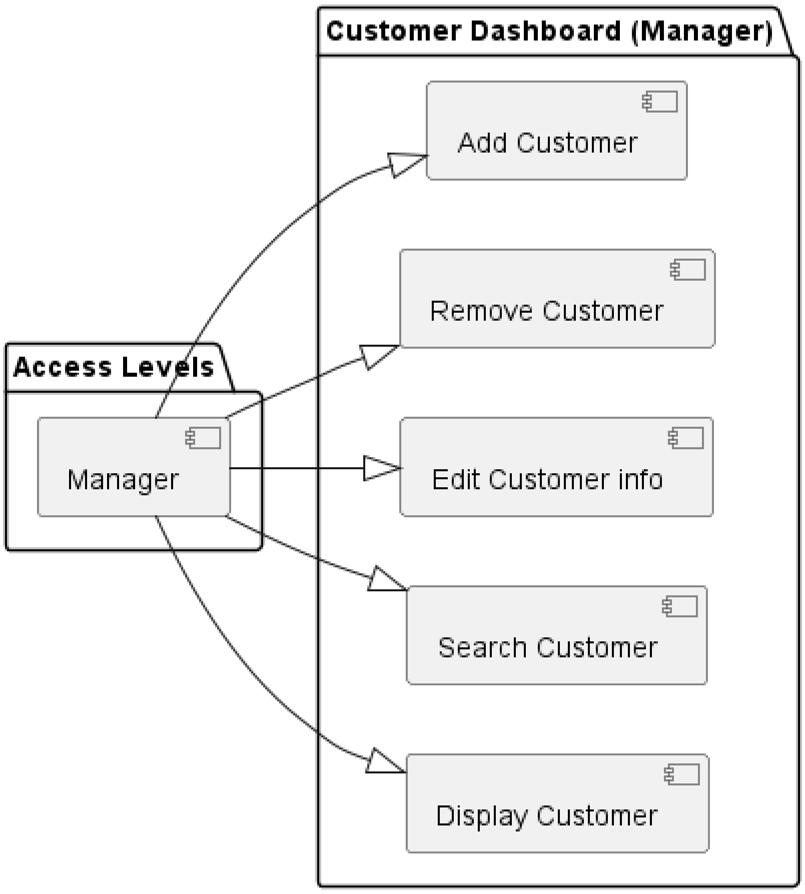
1. Add Customer- This feature is available only to employees and managers. Whenever the company adds a new customer, any employee or manager should be able to add it to the list of customers that the company serves.
2. Remove Customer- This feature is available only to employees and managers. Whenever the company needs to let go of a customer, any employee or manager should be able to remove him. However, we must check that the customer has no car in his possession. We will just set the isRemoved flag to true when the customer is removed
3. Edit Customer info - This feature is available to everyone. However, a customer should be able to edit only their information. Employees and managers can edit information for all customers.
4. Search customer- This feature is available only to employees and managers. One should be able to search a customer's ID, phone number, email, etc.
5. Display customer- This feature is available only to employees and managers. This shows the list of all the customers in a company.

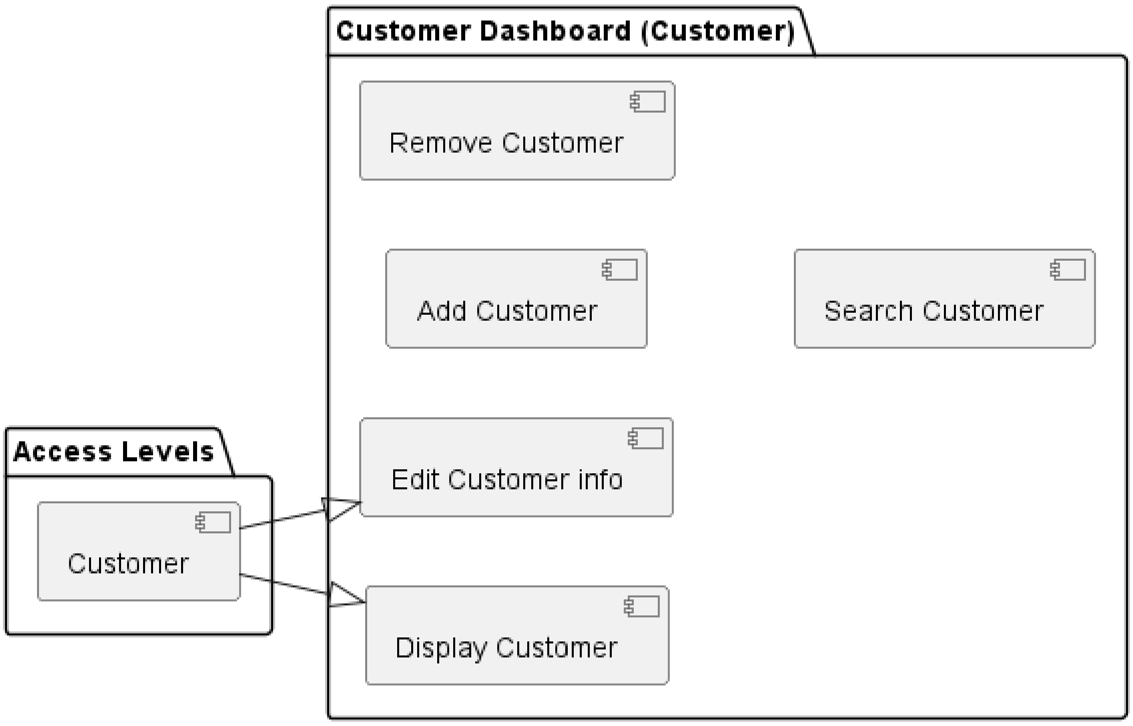
Each new customer for the company must be created by an employee and added to the company’s database. The attributes of a customer are as follows:

Attribute	Mandatory	Data type	Description
customerId	Y	string	The unique ID for the customer
firstName	Y	string	First name of the customer
lastName	Y	string	Last name of the customer
address	Y	string	Address for the customer
email	Y	string	email ID for the customer
contactNo	Y	long	The contact number of the customer
isRemoved	Y	boolean	The default value is false. This will be set to true when the customer

			is no longer with the company




Customer Dashboard (Manager)
Add Customer
Remove Customer
Access Levels
Manager
Edit Customer info
Search Customer
Display Customer


Customer Dashboard (Customer)
Remove Customer
Add Customer
Search Customer
Access Levels
Edit Customer info
Customer
Display Customer

Employee Data Dashboard

Author: Arshdeep Singh

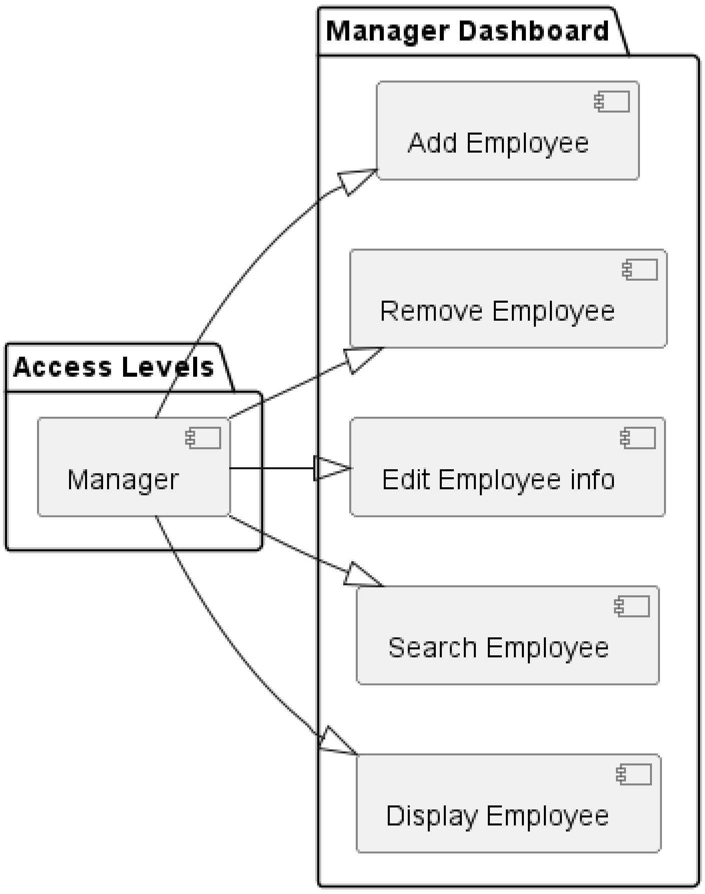
The employee dashboard will have the following features

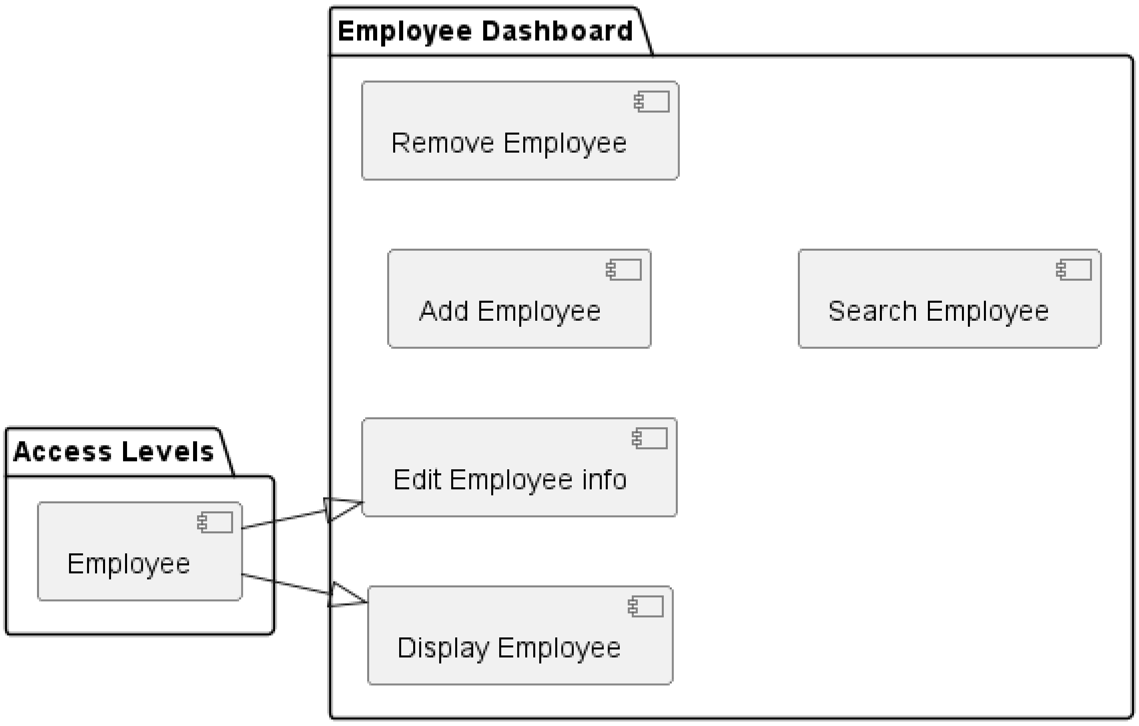
1. Add Employee- This feature is available only to managers. Whenever the company adds a new employee, the said employee needs to be added to the company’s database
2. Remove Employee- This feature is available only to managers. The relieved employees will need to be removed. The isRemoved attribute is set to true for this.
3. Edit Employee info - This feature is available to managers. However, an employee should be able to edit only their information.
4. Search Employee- This feature is available to all employees. One should be able to search an employee ID, phone number, email, etc.
5. Display Employee- This feature is available to all employees and managers. This shows the list of all the employees in a company.

Each employee for the company must have the following attributes:

Attribute	Mandatory(Y/N)	Data type	Description
employeeId	Y	string	Unique id for an employee
firstName	Y	string	string to store the first name
lastName	Y	string	string to store the last name
address	Y	string	Address of the employee
email	Y	string	Email ID for the employee
phoneNumber	Y	string	Phone number of the employee
dateOfBirth	Y	Date	Date of birth of the emplloyee
hireDate	Y	Date	Date at which the employee was hired
isRemoved	Y	boolean	The default value is false. This will be set to true when the employee is no longer with the company

isManager	Y	boolean	The default value is false. If an employee is a manager this value will be set to true.




Employee Dashboard
Remove Employee
Add Employee
Search Employee
Edit Employee info
Display Employee
Access Levels
Employee

Rental History and Invoice

Author: Arshdeep Singh

Rental history will include the following details:

Attrbiute	Mandatory(Y/N)	Data type	Description
bookingId	Y	string	A unique ID for the booking
car	Y	Car	The car that has been booked
customer	Y	Customer	Customer who has booked the car
employee	Y	Employee	Employee who has helped in the booking
bookingDate	Y	Date	The date when the car was booked
returnDate	N	Date	The date when the car was returned. If the car is still in the customer’s possession, the value will be NULL
cancellationDate	N	Date	In case the customer cancels the booking, the cancellation date will be filled

We will have a different view for customers and employees of the company. Customers will only be able to see their own rental history, while employees and managers will be able to see the rental history for all the customers. The same restrictions will apply to invoices as well. Whenever an invoice is required, an invoice will be generated containing all the relevant details of the booking.

Booking and Return Dashboard

Author: Amisha Nakrani

This dashboard is available to everyone: customers, employees, and managers.

1. Booking a car: Whenever a booking is made, a new entry will be added to the rental history for the customer with the customer, employee, and booking date.
2. Returning a car: The return date will be added to the corresponding booking entry when the car needs to be returned.
3. Cancelling a car: If the customer cancels the booking, the cancellationDate in the rental history will be filled.

There a different validations in place so that we can’t book removed cars. Also the invoice will only be calculated for cars that have been returned.

Fleet Maintenance

Author: Ghazala Anjum

This dashboard is used by the employees and managers of the company to monitor and maintain the condition of the cars at the company.

These attributes will be added to the car to monitor its condition.

Attribute	Mandatory(Y/N)	Data type	Description
lastOilChange	N	Date	This will record the last oil change of the car
nextOilChange	N	Date	This will record the next oil change of the car
lastInspectionDate	N	Date	The will record the last inspection date of the car
nextInspectionDate	N	Date	This will record the next inspection date of the car
carStatus	Y	enum	Functional and OutOfService will be the two enum values. If the car is not working properly, we can mark it as out of service
lastWashingDate	N	Date	This will record the last time the car was washed
nextWashingDate	N	Date	This will record the next time the car is due for washing

This dashboard can be used to see the following:

1. Cars due for oil change: We can see cars that are due for an oil change in the next week
2. Cars due for washing: We can see cars that are due for washing in the next week
3. Cars due for inspection: We can see cars that are due for inspection in the next week

4. Out-of-service cars: Employees and managers can see out-of-service cars so that their maintenance can be prioritized

Finance Report

Author: Aashika Vachhani

This view is available only to managers. Managers can use this to monitor the company's finances and understand which car makes and models generate more revenue. The finance report will include the following:

1. Revenue within a specific time period: This will allow the managers to enter a start date and end date, and managers will be able to see the revenue generated for a particular time period
2. Revenue generated by a particular car: This will allow managers to see how much revenue has been generated by a particular car
3. Revenue generated by a particular customer: This will allow managers to see how much revenue has been generated by a particular customer at the company

Outstanding issues

1. Implementation of PDF for Invoices:
We have yet to implement a feature to generate PDF invoices for our customers. This functionality will enhance the professionalism of our invoicing system and provide customers with a standardized and easily shareable document format.

2. Customer Communication via Email:
Currently, no automated email notifications keep customers informed about their bookings and invoices. Integrating automated email notifications will ensure timely updates for customers, improving their experience and engagement with our services.

3. Utilizing JavaFX for Enhanced GUI:
We have the opportunity to enhance our system's graphical user interface (GUI) by leveraging JavaFX technology. JavaFX offers a more modern and user-friendly interface design, which can improve usability and customer satisfaction with our application.

System evaluation

1. **Functionality**:
    - **Goal**: To provide a comprehensive rental management system with features like booking cars, returning cars, managing employees and customers, and generating finance reports.
    - **Result**: The system successfully implements all required functionalities, allowing users to book, return, and cancel cars, manage employees and customers, and generate finance reports
2. **Ease of Use**:
    - **Goal**: Design an intuitive user interface that is easy to navigate and understand.
    - **Result**: The system features user-friendly menus and prompts that guide users through different tasks. Input validation mechanisms ensure that users provide correct inputs, reducing the chance of errors.
3. **Reliability**:
    - **Goal**: Build a reliable system that operates smoothly without frequent failures.
    - **Result**: The system undergoes rigorous testing to identify and address potential issues. Error handling mechanisms are in place to gracefully handle exceptions and prevent crashes.
4. **Security**:
    - **Goal**: Implement robust security measures to protect sensitive data and prevent unauthorized access.
    - **Result**: User authentication mechanisms are enforced to control access to different system functionalities. Passwords are stored securely using encryption or hashing techniques.
5. **Maintainability**:
    - **Goal**: Create a system that is easy to maintain and update over time.

- **Result**: The codebase follows best practices and coding standards, making it easy for developers to understand and modify. Documentation is provided to aid in system maintenance and troubleshooting.

Learnings

From this project, we've learned several valuable lessons:

1. **Domain Knowledge**: We gained a deeper understanding of the car rental industry, including the various processes involved in managing fleets, booking cars, handling customer interactions, and generating reports.
2. **Database Management**: We enhanced our skills in database management, including schema design, query optimization, and data manipulation. Working with SQL databases helped us better understand data modeling and relational database concepts.
3. **Error Handling and Validation**: We gained insights into error handling strategies and input validation techniques to ensure the robustness and reliability of the system. Handling exceptions gracefully and providing meaningful error messages were essential aspects.
4. **Collaboration and Communication**: Working on a project with multiple people taught us the importance of effective communication and collaboration. Clear and regular communication among team members ensured smooth progress and alignment with project goals.
5. **Continuous Improvement**: The iterative nature of software development highlighted the importance of continuous improvement. Embracing feedback, identifying areas for enhancement, and iterating on solutions were critical for delivering a high-quality product.

Overall, this project provided us with hands-on experience in various aspects of software development and reinforced the importance of problem-solving, teamwork, and continuous learning in achieving project success.