CraZy CarZz

By Team 19:
Allen Jang U36320493
Vincent Nguyen U36440571
John Bestavros U80097673
Alexander Epifantsev U71426550

Professor Ahmed Ibrahim

April 29, 2022

Problem Definition

The business needs a mobile application where customers will be able to search for and reserve cars for hire. The specific functionality of the system:

- Online verification of a driving license
- Search for car models, rental prices, car type, location
- View current rental time remaining, extend car rent time,
- Rent multiple cars at once, manage all rented cars
- Cancel a rent early, change rental time, return a car early
- Pick a return location in advance, keep it reserved until car returned
- Log a regular rent schedule for commuters (if possible)
- Contact the company employee/operator regarding any issues(Car accident, etc.)

Project Objective

The objective of this project is for a car rental business owner to have a software with functionality described in the problem definition, by April 29th.

Stakeholder identification

Car Rental Owners

- Interest:
    - Minimize errors while renting cars
    - Easy to use system to keep track of information regarding car rental(s)
    - Save time and have potential to expand business to other locations
- Impact:
    - Rental Cost Savings; Less low-cost employees since their responsibilities are covered by the application's functions; Less tracking expenses.

Car Rental Employees:

- Interest:
    - Minimize errors while renting cars
    - Easy to use system to keep track of information regarding car rental(s)
- Impact:
    - Time saving

Car Rental Customers: College students, Tourists, People at the early stages of their career who cannot afford buying a car yet, People who work on the opposite side of the city

- Interest:
    - Convenient way for people to rent a car online within just a couple minutes, rather than going to a physical rental agency.
    - Convenient and everyday access to car usage without direct payment.
- Impact:
    - Time saving

Success/Acceptance Criteria

Meeting expectations would be if software is developed correctly. Exceeding expectation of stakeholder(s) would be significant cost savings for car rental owners and time savings for employees and customers.

Use case diagram

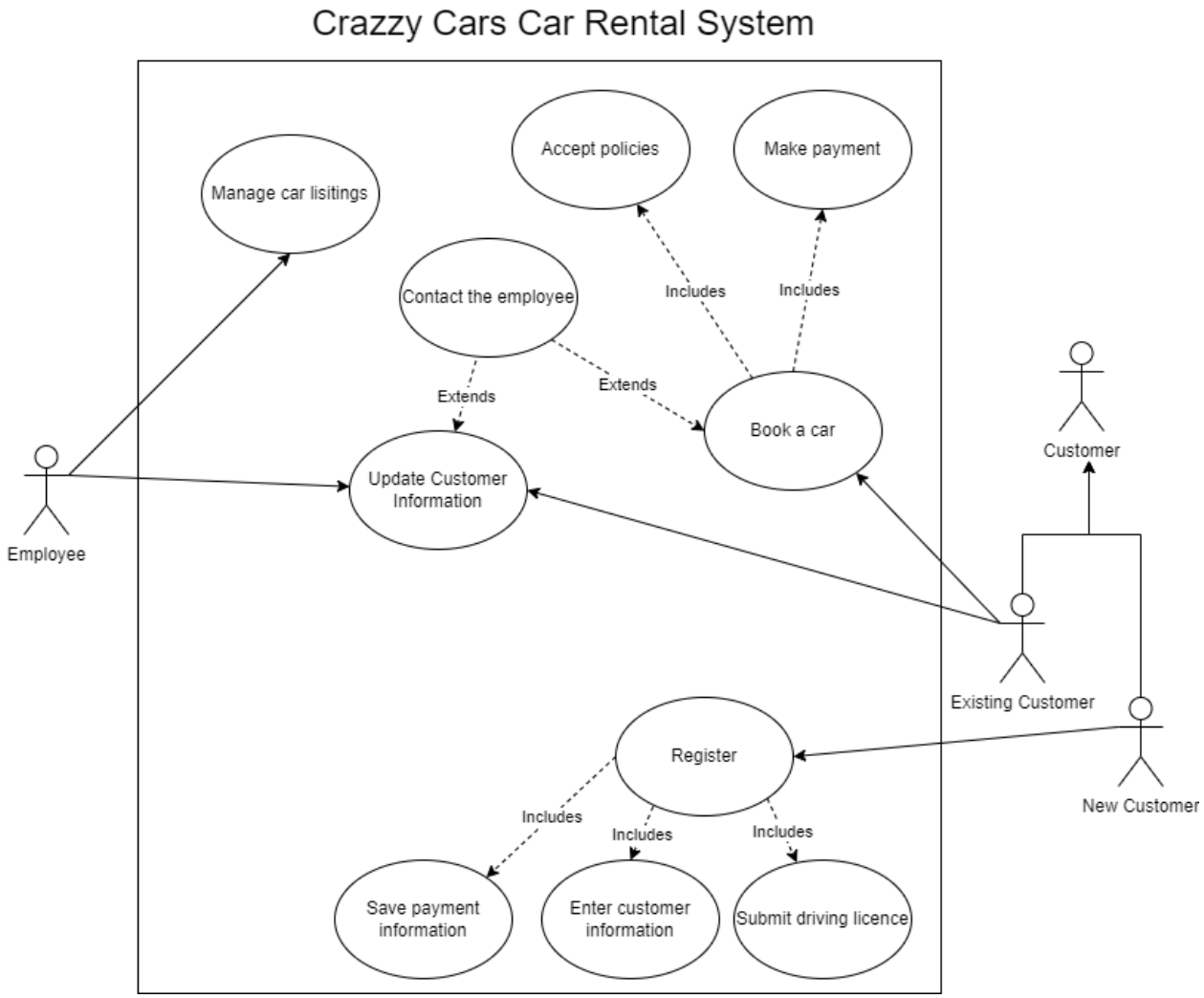

Figure 1: Use case diagram for CraZy CarZz

Figure 1, which is listed above is a graphical representation of a user's possible interactions within the system of CraZy CarZz. Within *Figure 1*, we can see the various use cases and the different users that will be interacting within the system, shown by the actors: Employee and Customer with the sub-actors which consist of Existing Customer(s) and New Customer(s).

Use case descriptions

Use Case Name: Book a car
Actors: Existing customer
Related Use Cases: Includes: Make a payment, Accept policies Extends: Contact the employee
Description: When a customer wants to book a car, the customer verifies payment and accepts liability policies and selects the car they want in application, within the car listings.
Stakeholders: Business owner: they want customer book car Employee: they depend on business that rent car Customer: they want car to go place(s)
Triggering event: Customer wants a car to drive but doesn't wanna buy!
Steps performed: 1. Go on app 2. Click “Log In” buttons 3. Go to listings 4. Pick an available car 5. Click “Book” button 6. “Read terms and conditions” and accept them 7. Verify payment
Preconditions: Customer doesn’t have a car
Postconditions: Customer has a car

Figure 2: Use case description for booking a car

Figure 2 is the description for the use case in which a user books a car using the CraZy CarZz application. Here *Figure 2* lists the related actors, related use cases, stakeholders, triggering events, steps performed as well as the pre and post conditions.

Use Case Name: Register
Actors: New customer
Related Use Cases: Includes Save payment information, Enter customer information, Submit driving license
Description: New customer creates account for system
Stakeholders: Business owner: they want customer book car Employee: they depend on business that rent car Customer: they want car to go place(s)
Triggering event: Customer has a desire to rent a car, but does not have an account yet to use the app
Steps performed: 1. Open app/website 2. Click “Register” button 3. Fill in customer information 4. Save payment method 5. Submit image of driver’s license / fill in driver’s license information
Preconditions: User is not registered, Has the application on their smart device
Postconditions: User is now registered, and able to book cars

Figure 3: Use case description for registering on application

Similarly, *Figure 3* also shows the use case description for the use case where a user is registering for an account for the CraZy CarZz application. Here *Figure 2* lists the related actors, related use cases, stakeholders, triggering events, steps performed as well as the pre and post conditions.

System Architecture: Client-Server

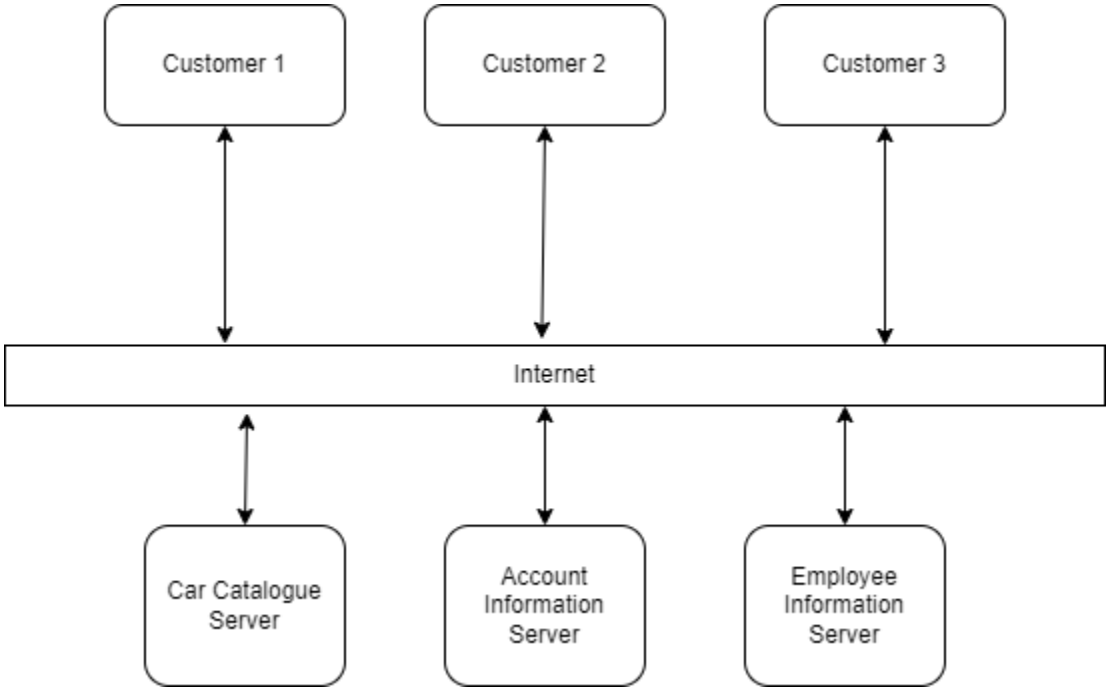

Figure 4: System Architecture for CraZy CarZz

CraZy CarZz's system architecture is a Client-Server architecture. CraZy CarZz has the following services held on separate servers such as the Car Catalog, Account Information, as well as Employee Information. Clients are able to access these servers and make use of them from any location while in the search for renting a car

Detailed Class Diagram

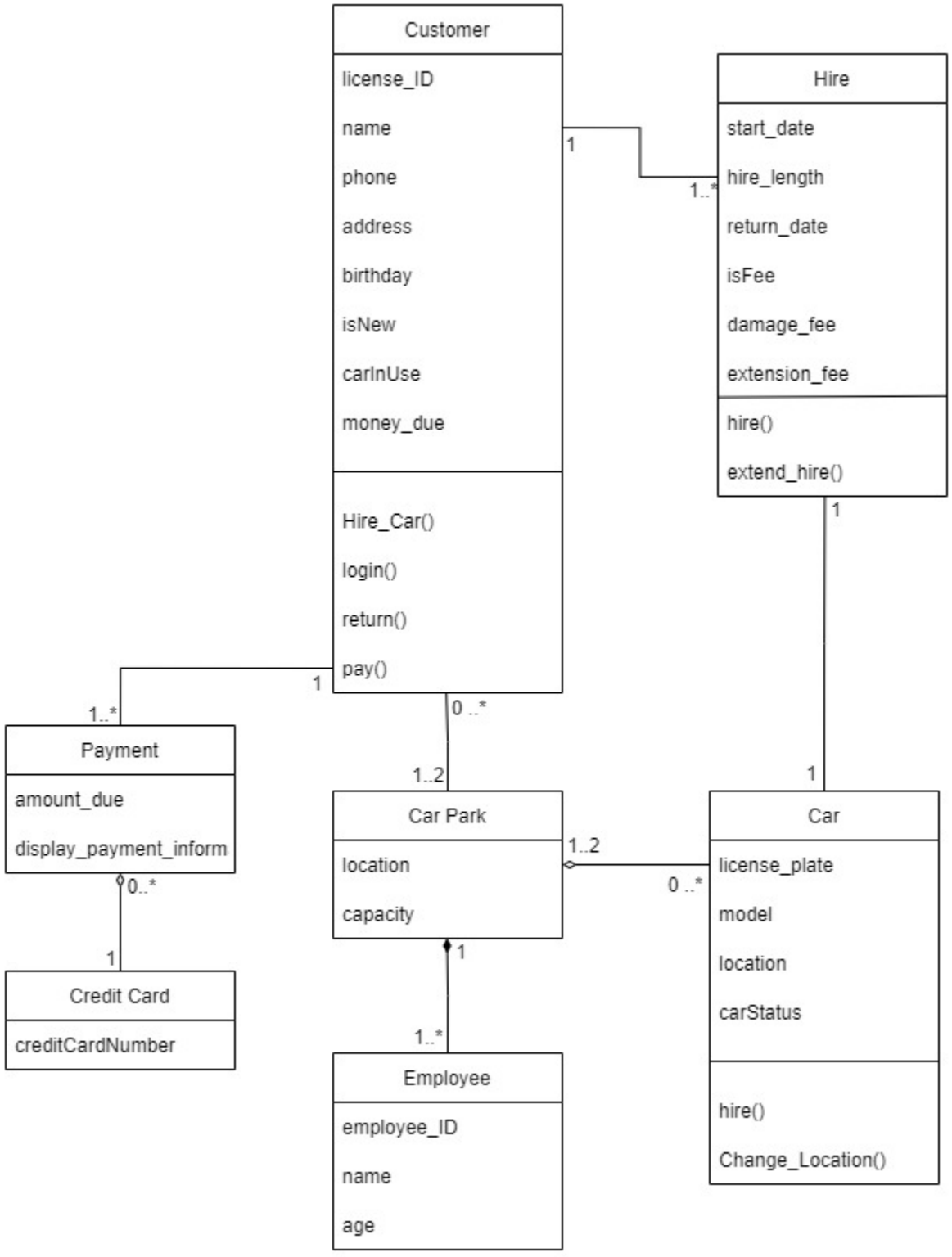

Figure 5: Detailed Class Diagram

Figure 5 is the detailed class diagram for our system. Which we used to help form our codebase.

State Machine Diagram

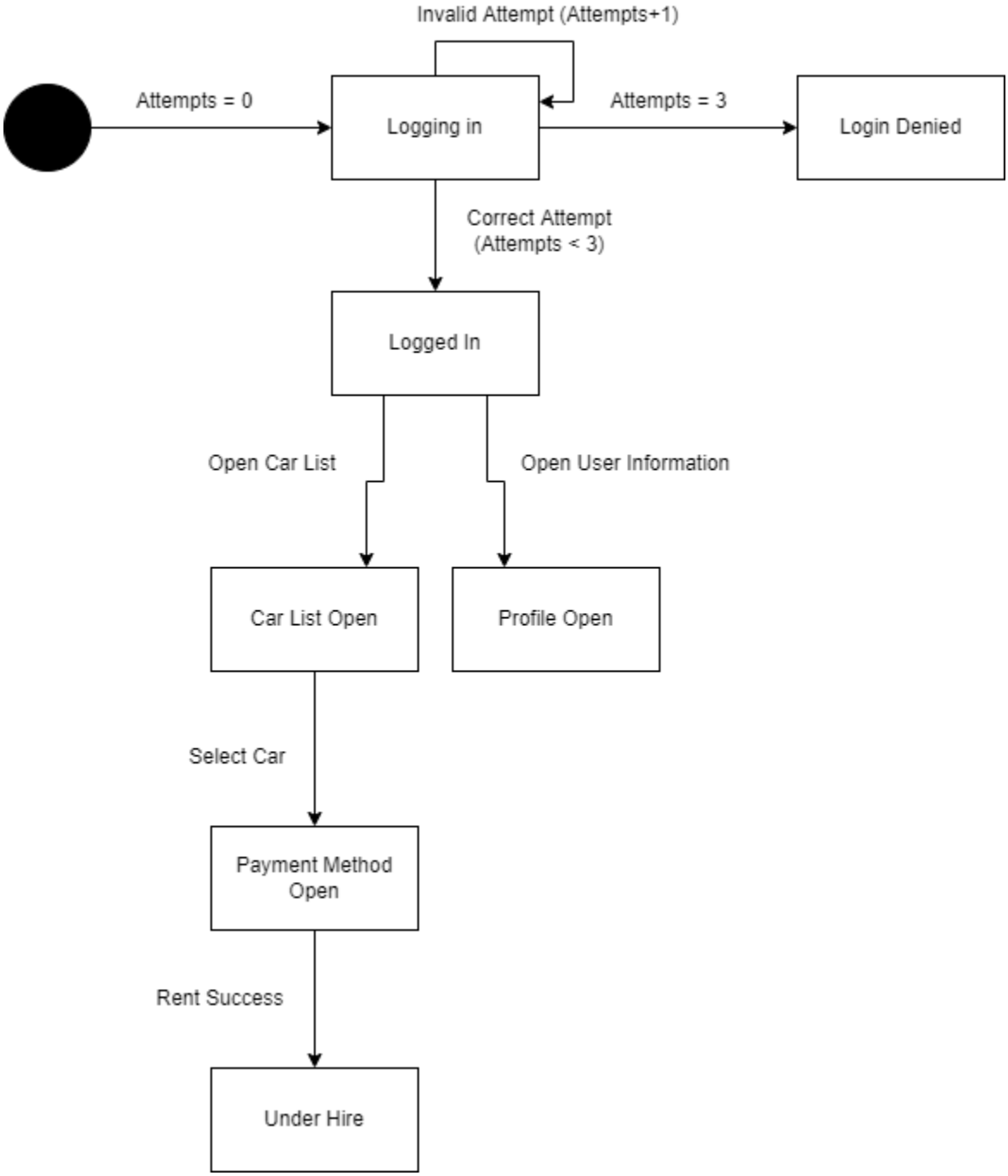

Figure 6: State machine diagram for CraZy CarZz application.

Figure 6 shows the different states that the CraZy CarZz application undergoes when a user is navigating through the system.

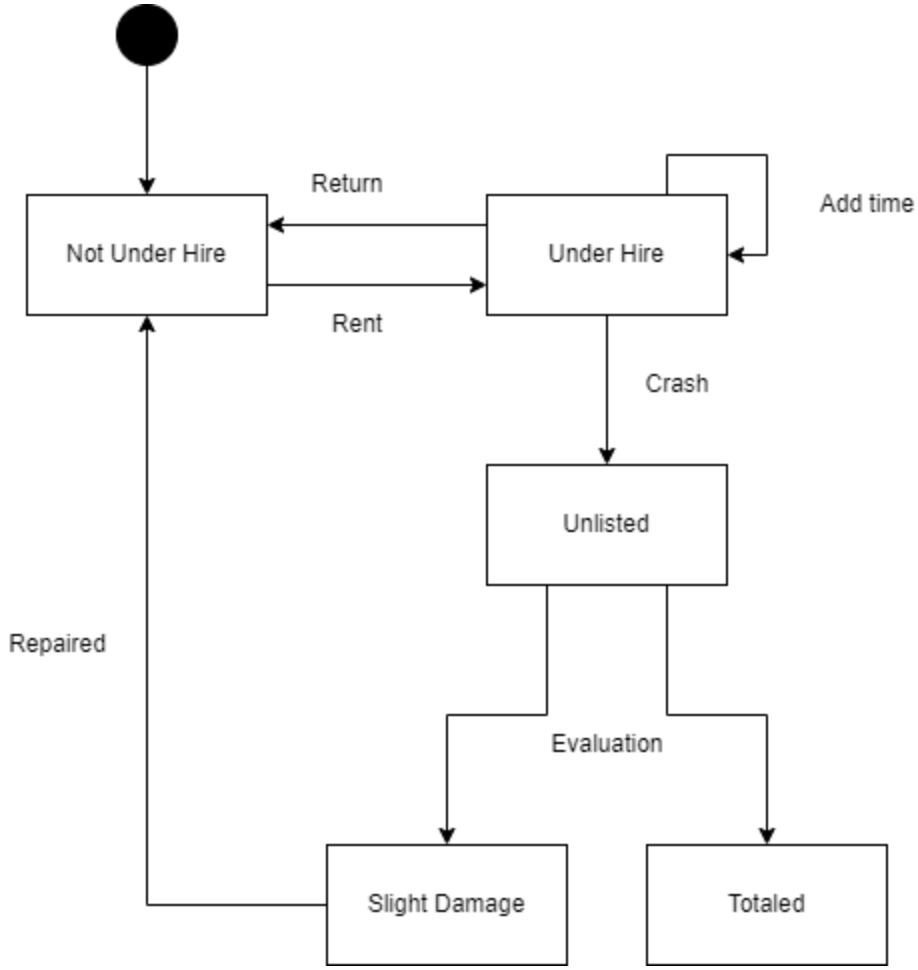

Figure 7: State machine diagram for car(s) rented with CraZy CarZz application

Figure 7 shows the different states that a car that is posted on CraZy CarZz undergoes once a customer selects that car and proceeds to book it.

ER Diagram

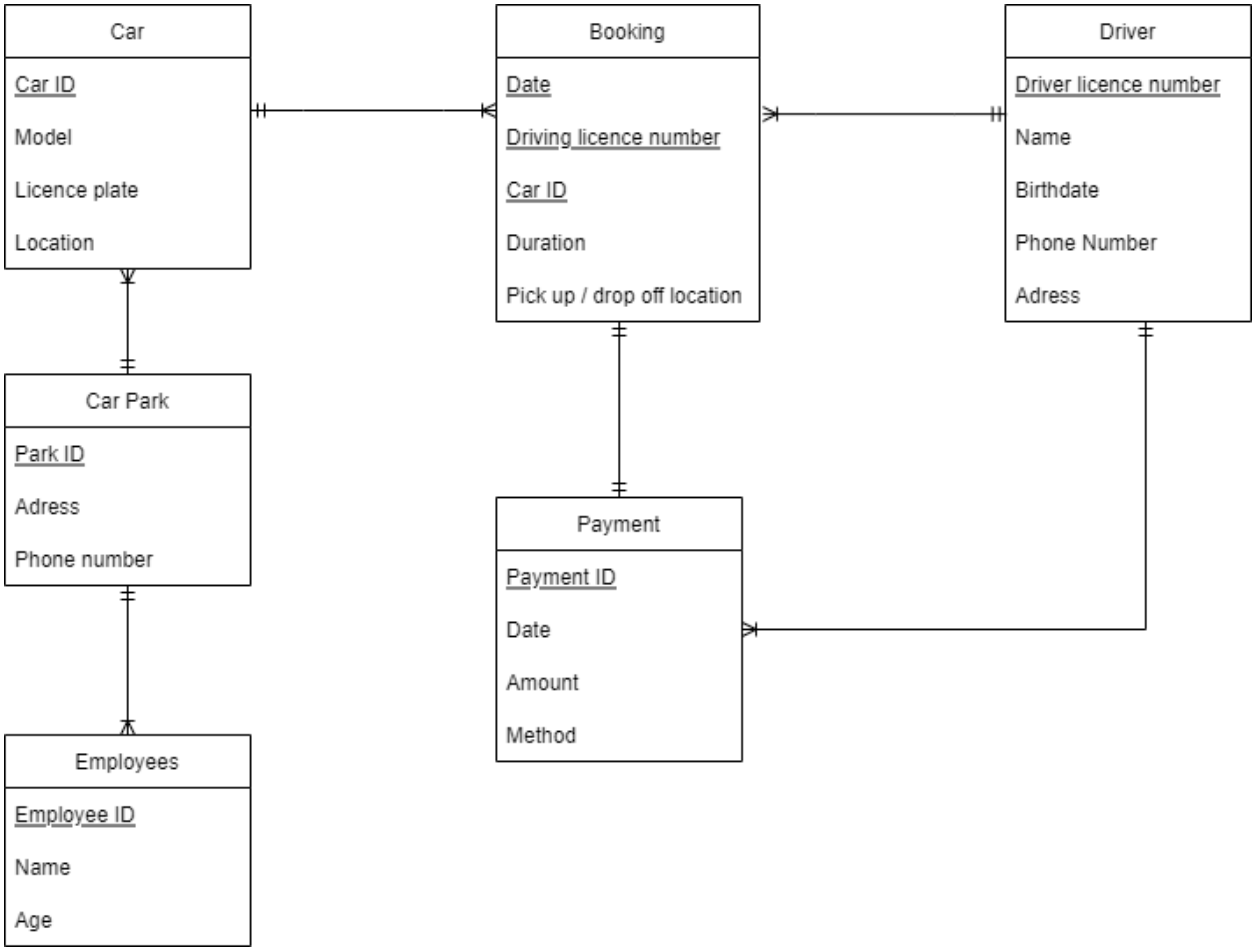

Figure 8: ER diagram for CraZy CarZz

Figure 8 is the ER diagram for CraZy CarZz which shows the different tables that will be used in the CraZy CarZz system. The items underlined within each table are the primary keys for that corresponding table. The ER diagram also shows the relationship between each connected table(s).

Github link: https://github.com/agjang/CraZyCarZz

Conclusion:
Some of the lessons we learned throughout the semester working on this project was the importance of communication, peer-editing/programming as well as seeking help not only from each other but from the professor and TAs as well. We learned that with strong communication, we are able to work together to solve problems that one of us may be facing and how to allocate resources to helping that one person overcome that roadblock. Furthermore we were able to communicate effectively about our week to week plan on preparing the material required for the Project Report, Presentation, as well as the implementation of our system.

References

ZipCar (https://www.zipcar.com/)
Turo (https://turo.com/us/en/car-rental/united-states)

Appendix:
Project WBS

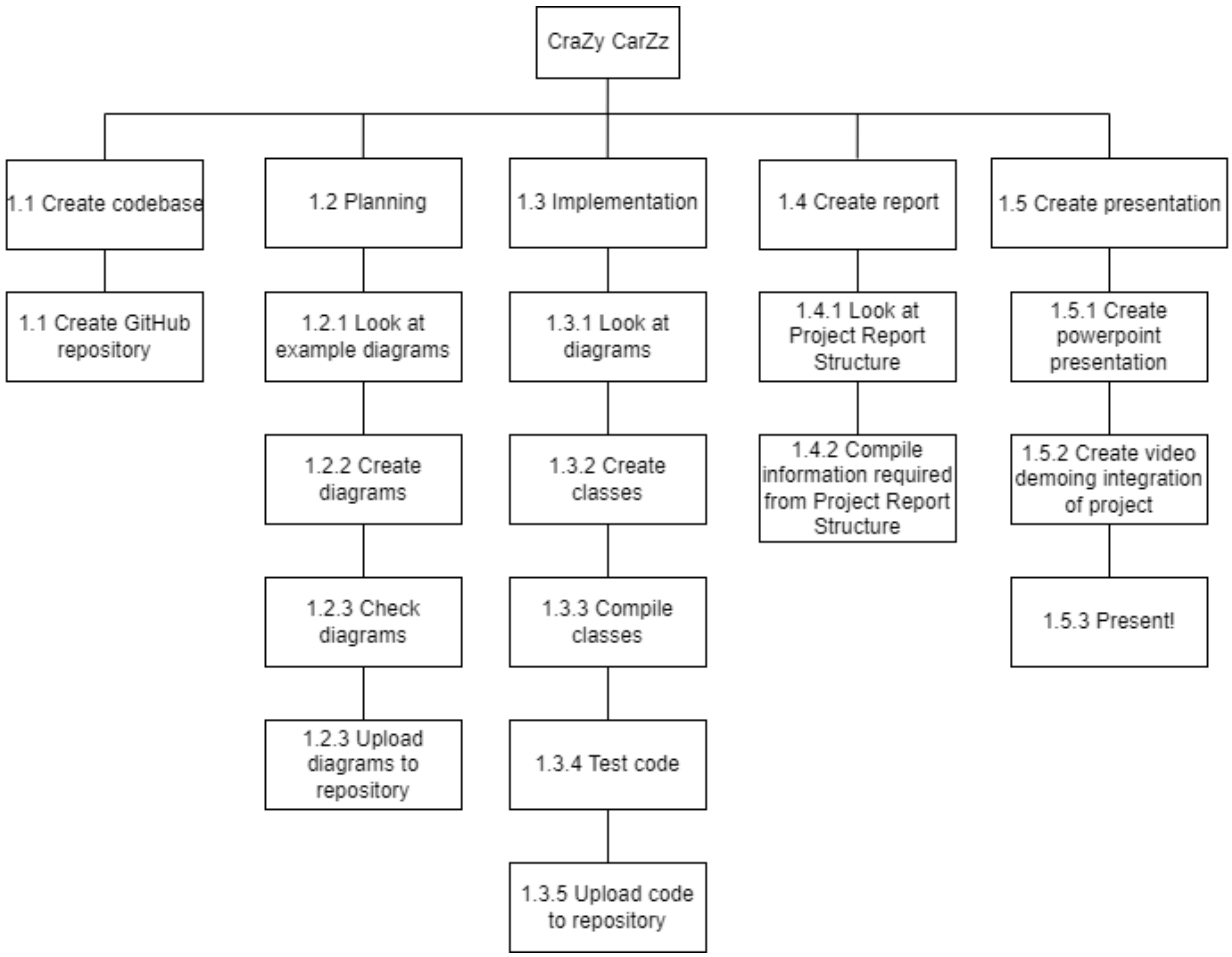

Figure 9: CraZy CarZz Work Breakdown Structure (WBS)

The above figure is a work-breakdown structure of CraZy CarZz. Our WBS is broken down into 5 smaller components: Create codebase, Planning, Implementation, Create report and Create presentation. These components are then broken down into smaller parts that help our team manage our workload and priorities.

Task Assignment Matrix

Task Assigned Matrix

Task	Owner	Support
Register	New Customer	
Submit driving license	New Customer	
Enter personal information	New Customer	
Save payment information	New Customer	
Update Customer information	Existing Customer	Employee
Manage Car Listings	Employee	
Book a car	Existing Customer	Employee
Accept Policies	Existing Customer	
Contact Employee	Existing Customer	Employee
Make Payment	Existing Customer	

Figure 10: Task Assignment Matrix for CraZy CarZz

The above figure shows the participation of various owners in completing the tasks within the CraZy CarZz system. It also shows the support that each owner may/may not have during the process of completing their corresponding task.

Sample of commits

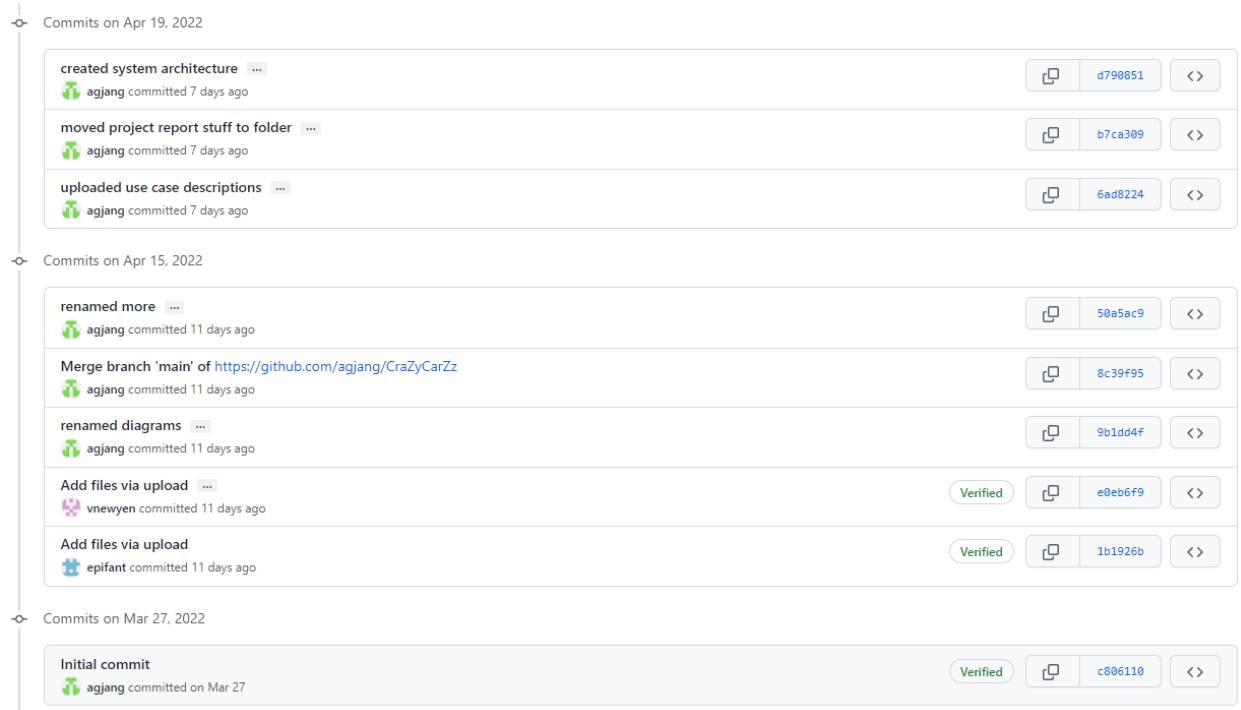

Figure 11: Sample commit for CraZy CarZz

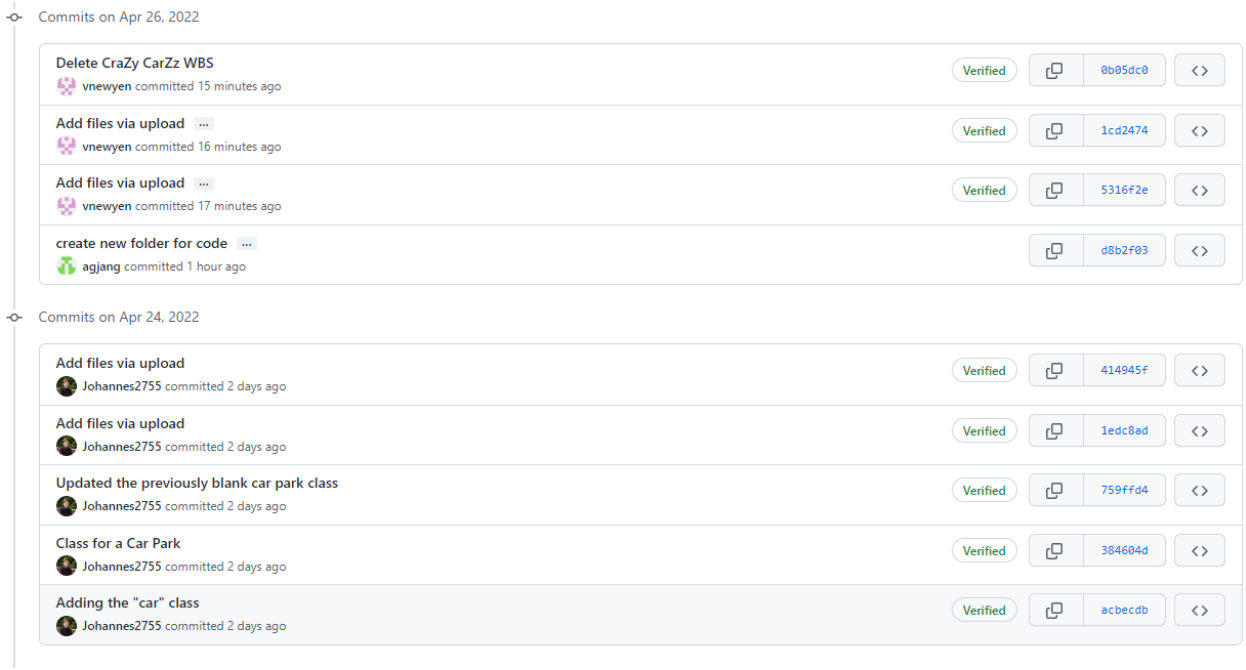

Figure 12: Another sample commit for CraZy CarZz