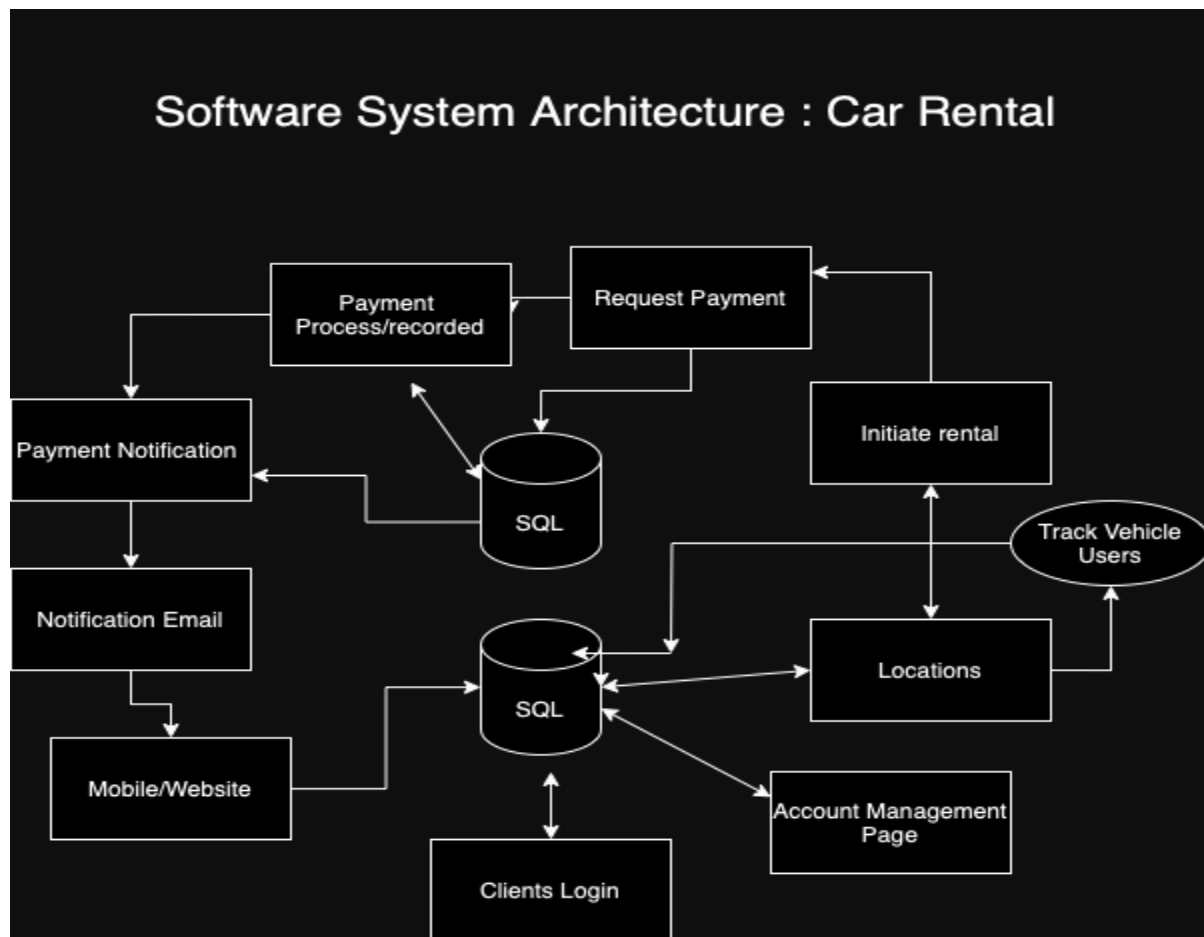


Software Design 2.0

Team Members: Jesus Serna, Matthew Sprague

Software Architecture:

Payment information must be securely stored and maintained separately from other system data to reduce the risk of privacy breaches. Additionally, the system should securely manage the distribution and signing of rental agreement contracts, as they contain customer signatures and sensitive information.



- Database Management:
 - The system utilizes two databases: one for payment-related information and another for other necessary data such as rental information, locations, user accounts, etc.
 - The payment database manages payment-related processes, including request payment and payment confirmation, ensuring secure and efficient transaction handling.
 - The second database manages all other aspects of the system, including user account management, rental tracking, location data, etc., providing a comprehensive and organized storage solution for the system's operational data.
- Mobile Application and Website Access:
 - Users can access the car rental system through either a mobile application or a website. Both versions of the system offer similar functionalities and require user authentication to access.
- Location Services:
 - Users are prompted to enable location/GPS services, allowing them to view rental locations near their current location and receive directions if needed.
- Rental Process Options:
 - Users have various options within the rental process, including making a rental, viewing rental history, and looking up rental locations.
- Distribution and Signing of Rental Agreements:
 - The system facilitates the distribution and signing of rental agreement contracts, ensuring they are kept under secure conditions to maintain customer confidentiality.
- Employee Access:
 - Employees of the company have access to the system, allowing them to review customer rental statuses and contracts, check the availability of cars for rental, and update the status of cars as needed, such as for maintenance.
- User Authentication:
 - Users need to create an account to use the system, providing valid information such as email and password. A verification code is sent to their email to finalize the account creation process, ensuring the validity of user accounts.

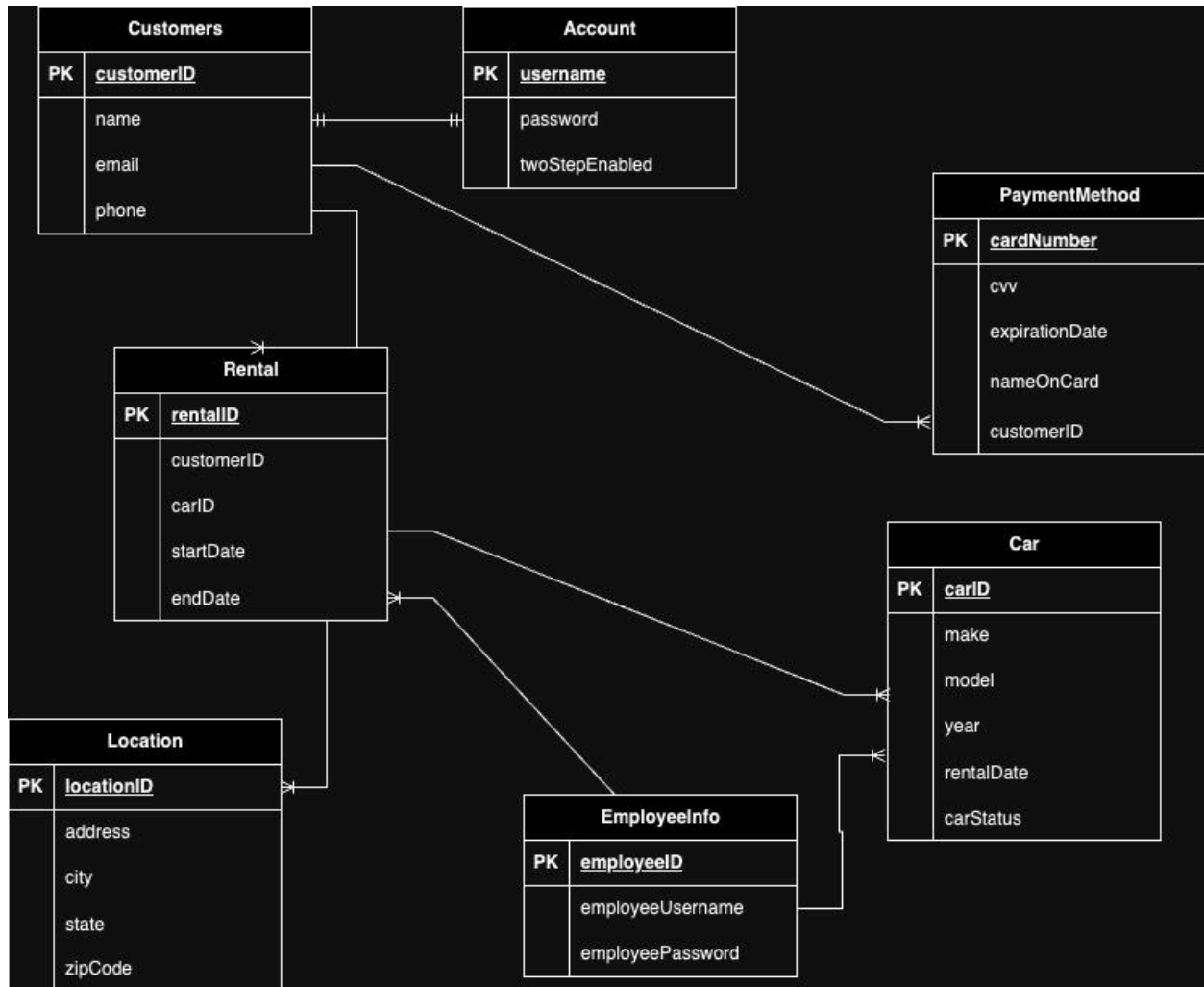
- Two-Step Authentication:
 - - While not enabled by default, users have the option to enable two-step authentication through the app's settings. This additional layer of security enhances account protection by linking with an authenticator app.
- Payment Method Linking:
 - - Users can link a payment method to their account for purchases. Payment information is stored securely and kept separate from other user data to minimize the risk of privacy breaches.

By incorporating these components into the system architecture and database design, the car rental system can effectively meet the needs of both users and employees while ensuring secure and efficient data management processes. In this modify diagram there is a major difference compared to the first implementation, the difference is that in this new version there is new database added that will be use to only store all payment information and anything related to it such as card info, payment type, hash transaction id, payment confirmation notification. This will add security features to the software system and will result in scalability meaning that it becomes easier to scale and optimize performance specifically for the demand of payment processing and will not affect the functionally of the other software functionalities.

Data Management Strategy:

For our software system we decided to go with a SQL relational data management strategy. Data will be managed on a single database.

Entity Diagram for our data:



Tables:

- Customers: stores the personal information about a customer
- Account: stores the login information of a customer
- PaymentMethod: stores the payment information/method that the customer has linked
- Rental: contains the information about a customer's rental
- Car: holds the information about the fleet of cars
- Location: stores information for the physical location of the rental cars
- EmployeeInfo: login information for employees

Relationships:

- One to One
 - Customers and account have a one to one relationship as each customer creates their own personal account
- One to Many
 - Customer to Rental is one to many as customers may be allowed to have multiple rentals but having only one is most common
 - Customer to PaymentMethod, as each customer can have multiple payment methods, such as credit cards, paypal, venmo, etc.
 - Car to Rental is one to many since cars can be rented multiple times by multiple different customers
 - EmployeeInfo to Rental and Car are both one to many since employees can lookup information for every customer's rental and for every car in the fleet
 - Rental to Location is one to one since a rental can be made at multiple locations and since they do not have to be picked up or dropped off specifically at the same location where the rental was made.

An SQL data management system was the system of choice for us over NoSQL due to its relational design, simplicity, and the fact that NoSQL does not guarantee Atomicity, Consistency, Isolation, and Durability which are crucial for a data management system that deals with a high volume of financial transactions. We chose to use a single database because this allows us to manage our data relatively simply and makes our transaction data flow more efficiently.

Utilizing a NoSQL database would also be a great way to manage the data in our car rental system as it definitely offers the scalability that SQL lacks. The horizontal scalability of a NoSQL database could help lead to faster data flow and processing. This would be beneficial especially for when cars are constantly being added to the fleet. However, a NoSQL database does provide the security and integrity that an SQL database does, and since our system consists of loads of transactions being processed, security is essential.