Car Rental System – Software Design
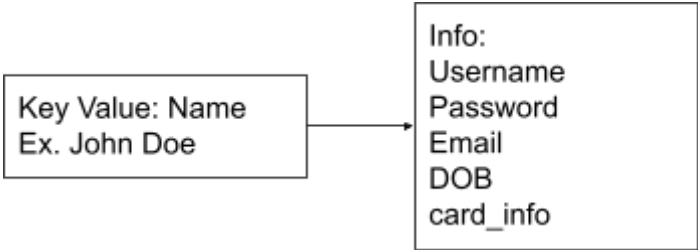
By: Winnie Thong, Khoi Tran, Chloe Kershner
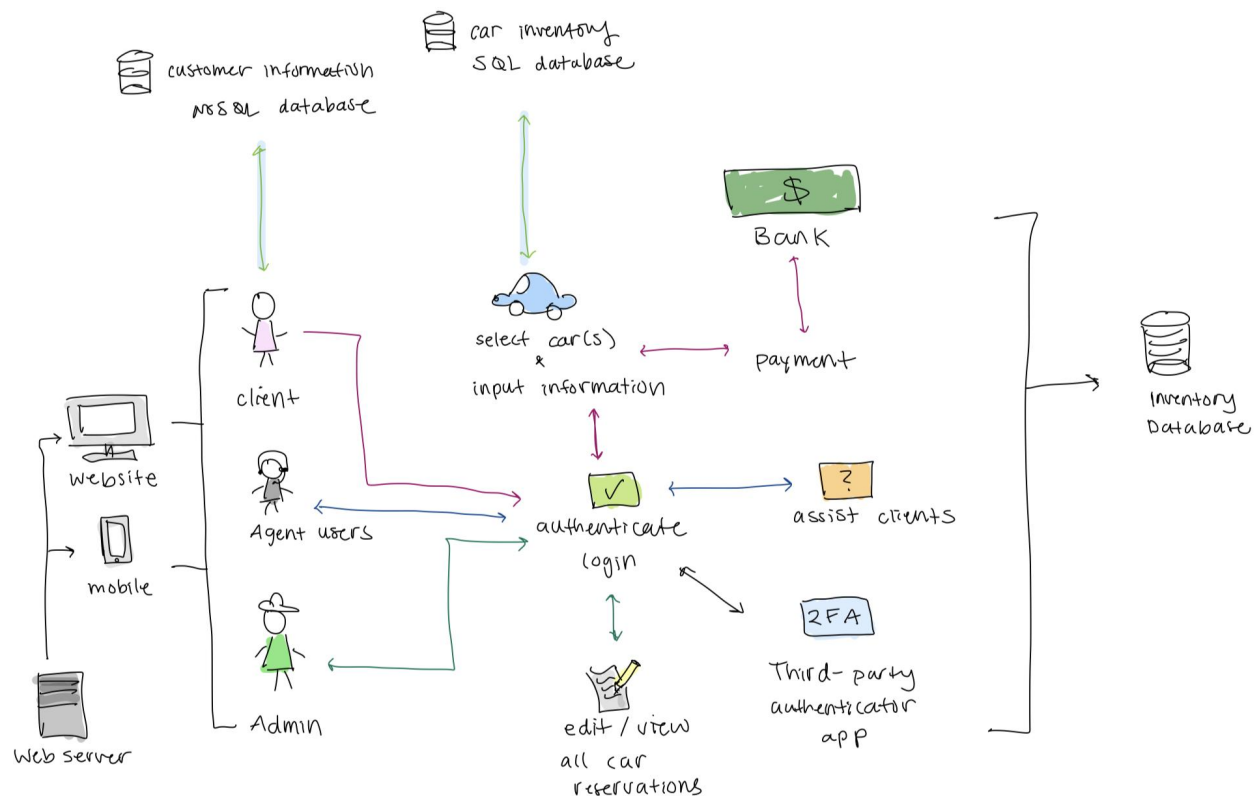
CS250 Group 5

NoSQL Example Database:

Name	Username	Password	Email	DOB	Card Info
John Doe	username1	password	john.doe@example.com	1990-01-01	["Visa **** 1234", "Amex **** 5678"]
Jane Smith	username2	password2	jane.smith@example.com	1992-02-02	["MasterCard **** 2345"]
Bob Brown	username3	password3	bob.brown@example.com	1988-03-03	["Visa **** 3456"]
Alice King	username4	password4	alice.king@example.com	1995-04-04	["Amex **** 4567", "Visa **** 6789"]
Omar Lee	username5	password5	omar.lee@example.com	1987-05-05	["Amex **** 7890"]

```
Key Value: Name          Info:
Ex. John Doe      ---->   Username
                          Password
                          Email
                          DOB
                          card_info
```

Software Architecture System Diagram:

car inventory SQL database

customer information NoSQL database

Bank

$

client

select car(s) & input information

Payment

Inventory Database

website

✓ authenticate login

? assist clients

Agent users

mobile

2FA

Admin

edit / view all car reservations

Third-party authenticator app

Web server

The software architecture system diagram for the car rental system first shows that the application can be accessed through a computer or mobile device by three types of users: clients, agents, and administrators. All devices should be connected to the web. After logging into the application, the users can authenticate their login by email, phone, or a third-party authenticator application.

The clients will be able to select their car, fill out and upload their information, and then make a payment. The payment will be connected to their bank account unless they opt to pay in person.

The agent users are there to help clients. They can assist with any technical issues or any questions the clients may have about their car reservations.

The admin users edit the available cars in the inventory or any customer information. They can also view all car reservations, past or present, within their branch.

All information is collected and stored in their inventory database, where security is prioritized.

The updates made are the two data management databases on the top that are highlighted in blue. The NoSQL database will be used to store customer information. The rest of the rental process will have access to this information. The SQL database will hold car information. It will be available to public users when they start looking for a car.

Car Rental SQL Database:

Users:

Fields: User ID, Username, Password, Email, Date of Birth, IsAdmin

User ID	Username	Password	Email	Date of Birth	IsAdmin
(int)	(string)	(string)	(string)	(date)	(bool)

Locations:

Fields: ID, Location Name, Coordinate, StockLevel

Location ID	Location Name	Coordinates	StockLevel
(int)	(string)	(float)	(int)

Vehicles:

Fields: Vehicle ID, Make, Model, Type, Year, License, Price, IsAvailable, Location ID

Vehicle ID	Make	Model	Type	Year	License	Price	IsAvailable	Location ID
(int)	(string)	(string)	(string)	(int)	(int)	(float)	(bool)	(int)

Payment:

Fields: Payment ID, User ID, Amount, Payment Date, IsPaid

ID Number	User ID	Amount	Payment Date	IsPaid
(int)	(int)	(float)	(timestamp)	(bool)

Analysis:

We used one database with 4 tables (Users, Locations, Vehicles, and Payment). This design aligned well with our original design, and split up the data accordingly. The Users table stores user information, and can be used by both a user and admin. The Location table stores appropriate information to distinguish between various locations. The Vehicles table has the most fields of all the tables, as it has lots of information to store in order to differentiate all of the vehicles. This ensures safety and protection for the application. The payment table includes information about payment, but does not save any sensitive information such as card information in order to ensure extra protection and safety for customers. Rather, it saves information regarding the transaction similar to what a receipt would show. We chose to use an SQL database, as it provides a useful query language for interacting with the database. This makes it easier to retrieve, update, and analyze data. In addition, SQL databases have a fixed schema. This is crucial for maintaining data integrity and consistency, something that is very important in our car rental system. A noSQL database is a possible alternative. It would allow for more flexibility and scalability. However, the SQL database ensures certain guarantees, which are important especially for transactions.