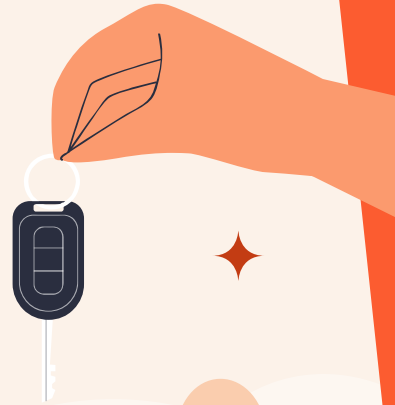
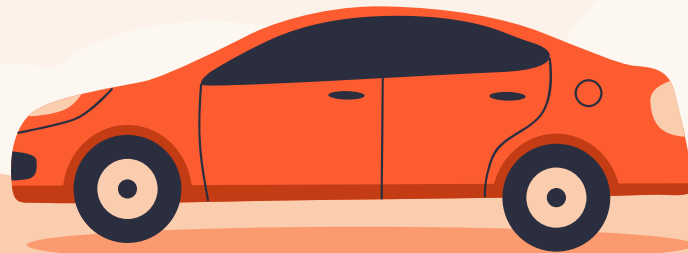


Valet Database

By Group 14: Kyle Burchett, Brian Tramuel, Kennedy Hooper,
Tre Watson, Max Saavedra



Why a Valet System?

Given the vast amount of information involved in managing a vehicle system, including different makes, models, colors, and other various sorts of information involved in identifying a vehicle, our group decided to use a valet system as our project. This, along with the continuous in-and-out process provided at restaurants and hotels, makes a valet management system all the more reasonable.

- With enough devices allows all employees to keep track of customer vehicles
- More convenient and faster than a paper tag based system
- Real time tracking allows management to know about vehicle whereabouts at all times.



Main uses, roles, business requirements, and priorities



Purpose of the Database

The purpose of the database is to help manage valet parking operations by tracking establishment locations, car parking locations, and employee actions.

This may include:

- Assigning and tracking parking spaces for each vehicle
- Ensuring accountability in case of damages or disputes
- Supporting real-time data access for a faster service
- Monitoring which employee parked and retrieved each vehicle

In the future:

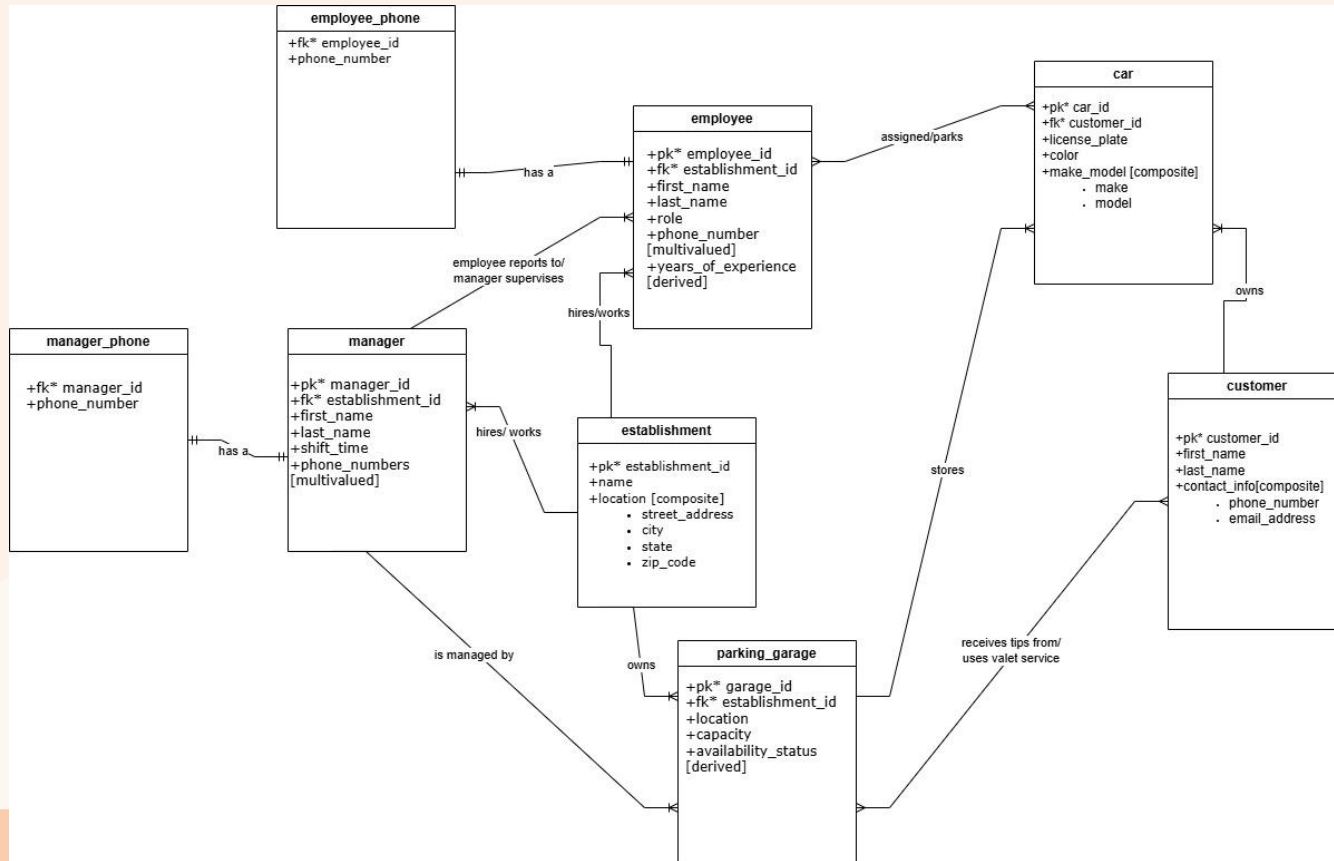
- Mobile app integration
- Digital Ticketing

Requirements & Priorities

- Vehicle Tracking
- Customer information
- Parking Spot Allocation



Conceptual design (ER Diagram)



Logical Design (Schema Examples BCNF)

Table: **employee**

columns:

employee_id INT PRIMARY KEY,
first_name VARCHAR(50),
last_name VARCHAR(50),
role VARCHAR(50),
years_of_experience INT,
establishment_id INT,
FOREIGN KEY (establishment_id)
REFERENCES
Establishment(establishment_id)

Table: **manager**

columns:

manager_id INT PRIMARY KEY,
first_name VARCHAR(50),
last_name VARCHAR(50),
shift_time VARCHAR(50),
establishment_id INT,
FOREIGN KEY (establishment_id)
REFERENCES
Establishment(establishment_id)

Table: **car**

columns:

car_id INT PRIMARY KEY,
customer_id INT,
license_plate VARCHAR(15),
color VARCHAR(30),
make VARCHAR(50),
model VARCHAR(50),
FOREIGN KEY (customer_id)
REFERENCES
Customer(customer_id)

Table: **customer**

columns:

customer_id INT PRIMARY KEY,
first_name VARCHAR(50),
last_name VARCHAR(50),
phone_number VARCHAR(15),
email_address VARCHAR(100)

MySQL Views

US6

As a customer, I want to know if parking is available so that I can plan ahead.

```
1 • CREATE
2     ALGORITHM = UNDEFINED
3     DEFINER = `root`@`localhost`
4     SQL SECURITY DEFINER
5     VIEW `available_parking` AS
6     SELECT
7         `parking_garage`.`garage_id` AS `Garage number`,
8         `parking_garage`.`availabilty_status` AS `Parking spots available`
9     FROM
10        `parking_garage`
```

	Garage number	Parking spots available
▶	1	100

This view is useful to quickly see how many parking spots are available within every single garage in the system.

MySQL Views

US3

As a cashier, I want to process payments for customers so that I can complete the service transaction.

```
1 • CREATE
2     ALGORITHM = UNDEFINED
3     DEFINER = `root`@`localhost`
4     SQL SECURITY DEFINER
5     VIEW `order_summary` AS
6     SELECT
7         `customer`.`first_name` AS `First Name`,
8         `customer`.`last_name` AS `Last Name`,
9         `customer`.`phone_number` AS `Phone Number`,
10        `customer`.`email_address` AS `Email Address`,
11        `car`.`model` AS `Car Model`,
12        `car`.`license_plate` AS `Liscence Plate`
13    FROM
14        (`customer`
15     JOIN `car` ON ((`customer`.`customer_id` = `car`.`customer_id`)))
```

	First Name	Last Name	Phone Number	Email Address	Car Model	Liscence Plate
▶	Kyle	Burchett	111-111-1112	dburche4@charlotte.edu	altima	12345
	Brian	Tramuel	111-111-1234	NULL	focus	12347

This view is helpful to see the information of any customers that interface with the valet system, making it possible to communicate with them and track sales.

MySQL Stored Procedures

US7

As a manager, I want to see the condition of the parking garage, and employees so that I can make sure everything is safe and running smoothly.

```
1 • CREATE DEFINER='root'@'localhost' PROCEDURE `list_cars`(IN garage_id INT)
2 BEGIN
3     Select `car`.`car_id` AS `Car id`,
4           `car`.`customer_id` AS `Customer Id`
5     FROM `valet`.`car`
6     WHERE `car`.`garage_id` = `garage_id`
7     ORDER BY `car`.`customer_id`;
8 END
```

```
1 • call valet.list_cars(1);
2
```

Result Grid | Filter Rows:

	Car id	Customer Id
▶	1	1
	3	2

This procedure will be useful to check the exact condition of a specific garage, to see the count of cars in the garage. Simply providing the ID to allow for follow up in the other tables of the database, if need be.

Issues

Overall in designing a valet database, it is important to be aware of potential issues that could arise and possibly hinder the efficiency and effectiveness. Taking the time to think about and address some of these issues now may help in the future. Some issues we thought about were:

- Some tables don't have the foreign keys to connect data between cars, customers, employees, and garages.
- IDs don't increase automatically in some tables, so it's easy to make mistakes when adding new data.
- Some columns can be empty even when they should always have a value like car ID or customer ID.



Future features

Looking at the future we thought it was important to consider how our database could grow and improve. Here's some things we thought of:

- A Functional Mobile App to use for valet service
- Text Notifications when the car is ready for pickup
- Option to tip valets through the app
- Live wait times that shows customers the wait time as they wait for there car
- A option to recover your ticket if you lost it, can get the ticket info from a recovery phone number or email
- Multiple language accommodations in case of international customers



Questions?



THANK YOU!

