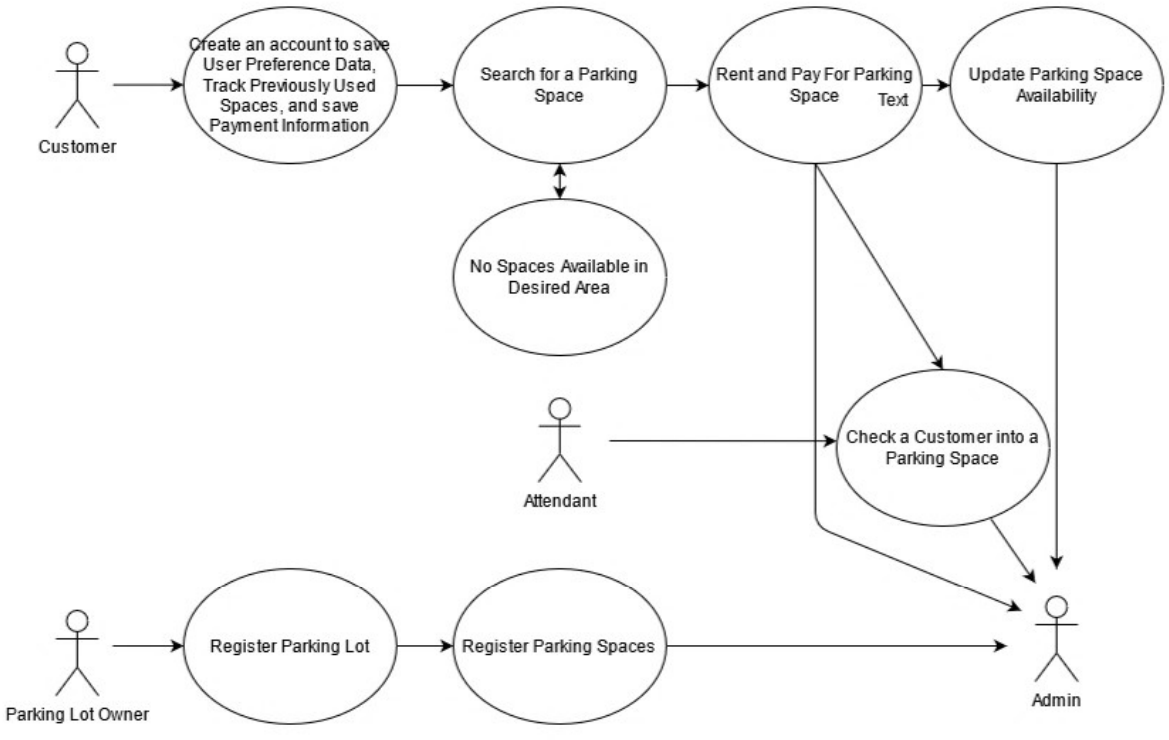
Use Case: User makes account and checks into parking lot.
Parking lot owner registers a parking lot.



Name: User makes account and checks into parking lot and Parking lot owner registers a parking lot.

Participating Actor: User, Parking Lot Owner

Entry Condition:

- Use Access mobile app to create account
- Parking Lot Owner access their account

Exit Condition:

- User gets checked into a parking space
- Parking Lot Owner successfully registers parking

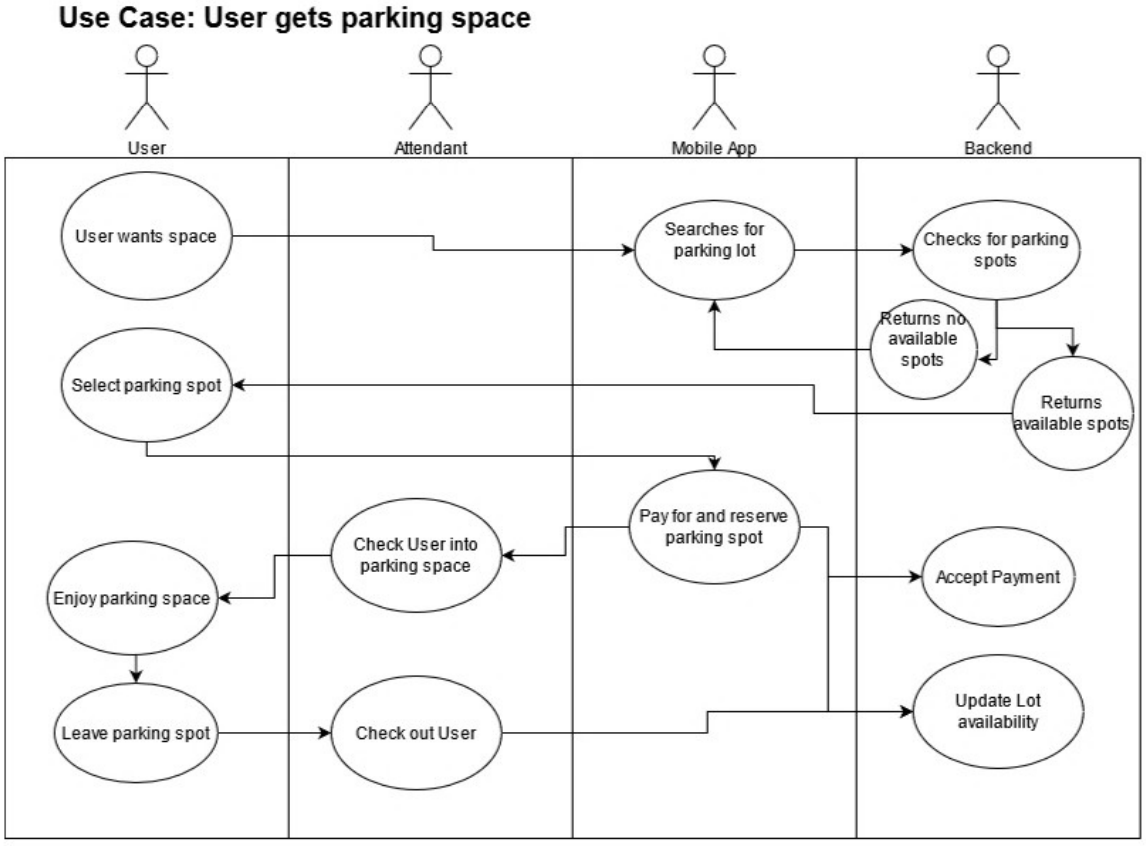
Event Flow(User):

1. User creates an account
2. User searches For Parking Space
3. User Pays For Parking
4. Attendant checks User into their parking space

Event Flow(Owner):

1. Log into account
2. Register New Parking Lot
3. Register Parking Space

Use Case: User gets parking space



Name: User Reserves Parking Space

Participating actor: User

Entry condition:

- User is looking for a parking space.
- User has not checked into reserved parking spot.

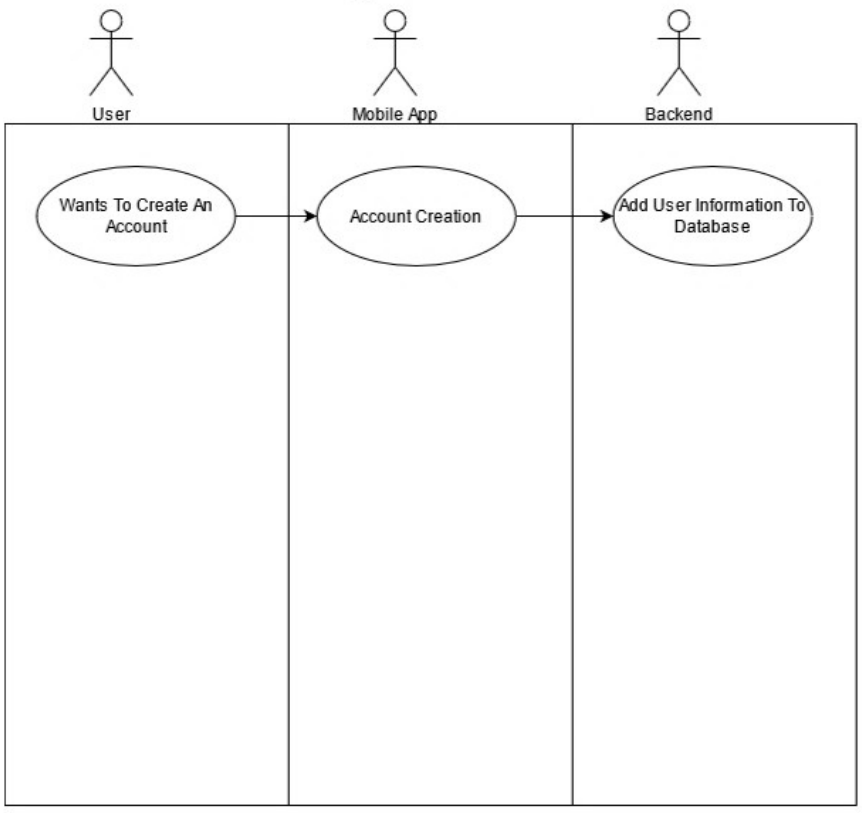
Exit condition:

- User checks out of parking spot

Event flow:

1. User searches for a nearby parking spot.
2. Backend returns any available parking spots.
3. User selects parking spot.
4. User pays for parking spot.
5. Backend accepts payment and updates lot availability.
6. Attendant checks user into space.
7. Attendant checks user out of space when done.
8. Backend updates lot availability.

Use Case: Creating An Account



Name: Customer creating an account for parking space

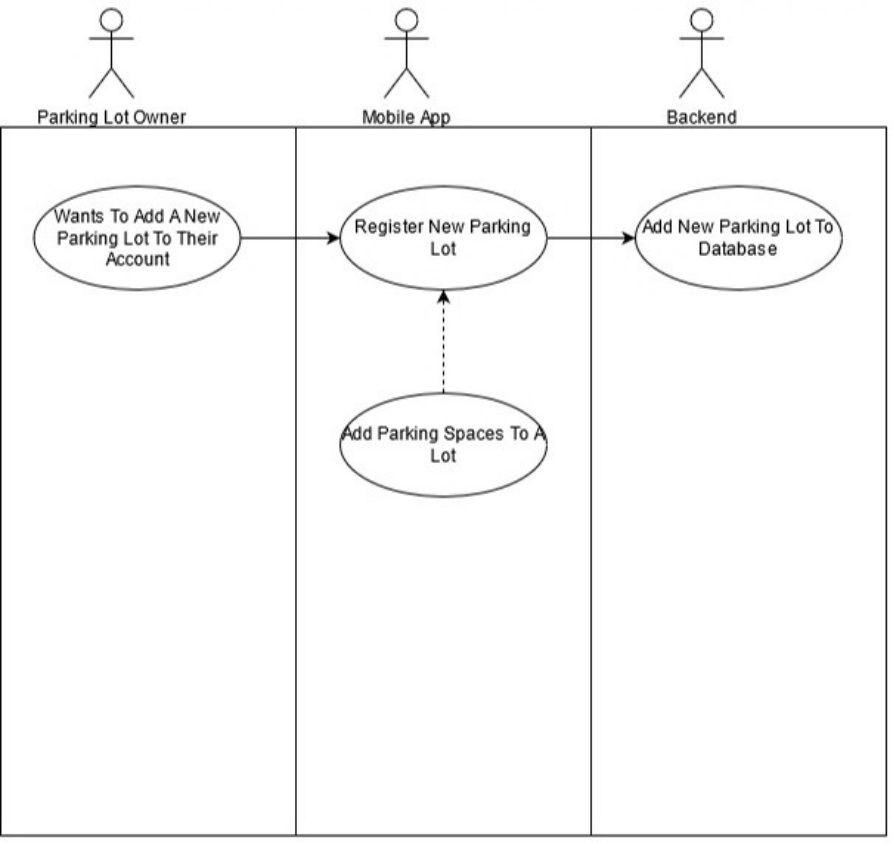
Participating actor: Customer, Mobile App, Backend

Entry Condition:
•Customer wants to purchase a parking space.
•Customer successfully accesses the create account portion of the web app.

Exit condition:
•Customer's new account is created and their account information is stored.

Event flow:
1. Customer wants to purchase a parking space.
2. Customer accesses the mobile app and selects the account registration.
3. Customer fills out the necessary information and registers.
4. Database is updated with new account information.

Use Case: Parking Lot Owner Adding A New Parking Lot



Name: Parking Lot Owner Adding A Parking Lot

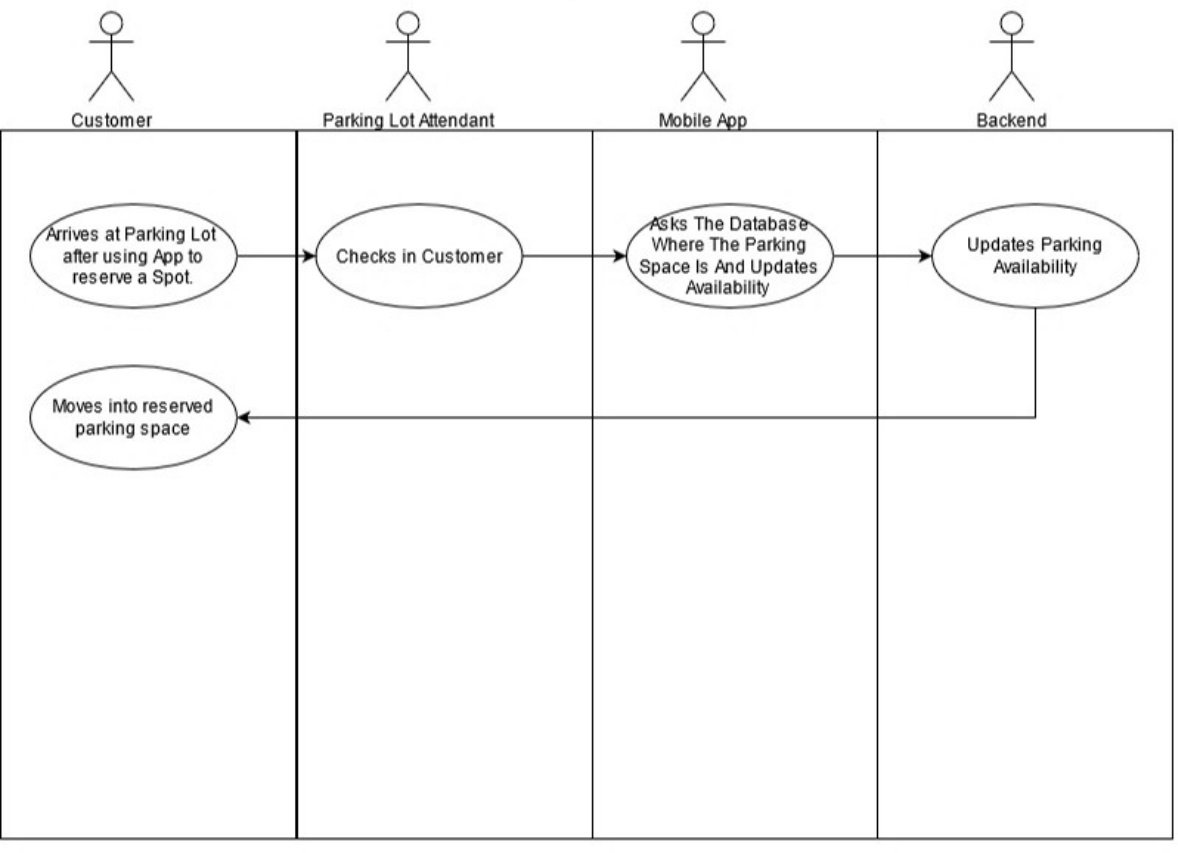
Participating actor: Parking Lot Owner, Mobile App, Backend

Entry Condition:
•Parking Lot Owner wants to add a new parking lot to his account.
•Parking Lot Owner successfully logs into his account and tries to register a new parking lot.

Exit condition:
•Successfully add a new parking lot to the database for the user.

Event flow:
1. User logs into their account.
2. Selects register a new parking lot.
3. User fills out the necessary information and registers.
4. Database is updated with new parking lot and information.

Use Case: Parking Lot Attendant Checking In a User With a Reservation



Name: Parking Lot Attendant Checking in a User with a reservation

Participating actor: Attendant, User

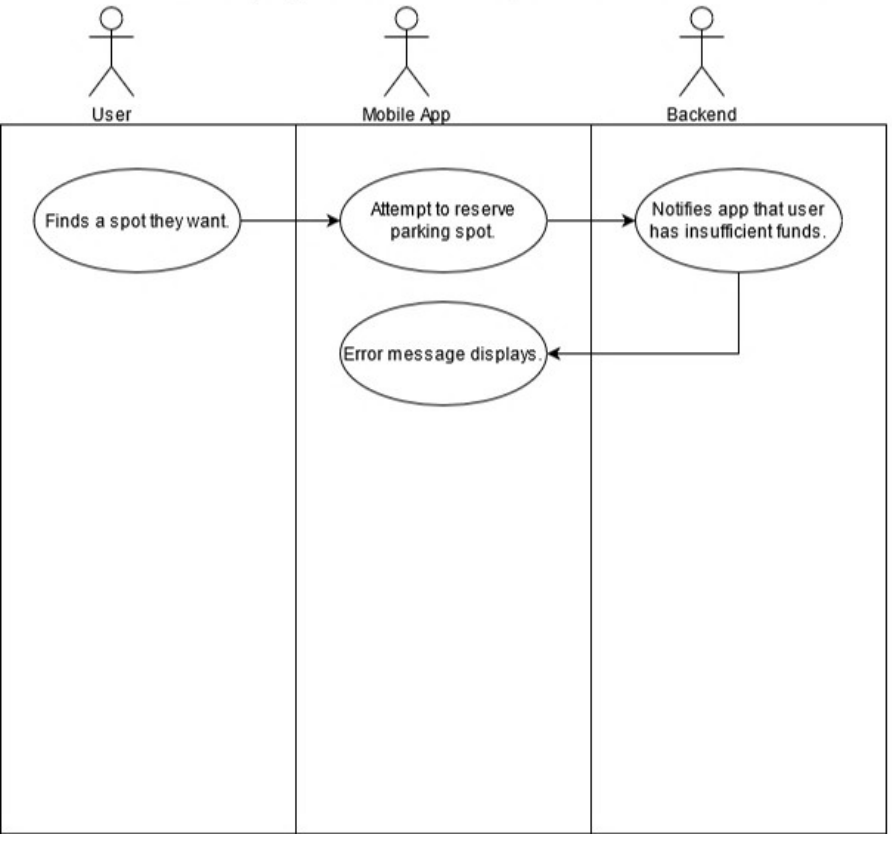
Entry condition:
- User arrives at parking spot with a reservation.

Exit condition:
- User checks into spot

Event flow:

1. User arrives at parking lot.
2. Attendant checks in Customer
3. Backend Updates lot availability.
4. User is shown where to park.

Use Case: Trying to reserve a spot with insufficient funds



Name: Trying to reserve a spot with insufficient funds

Participating actor: Customer, Mobile App, Backend

Entry Condition:
•Customer wants to purchase a parking space.
•Customer doesn't have enough funds in account to reserve parking space.

Exit condition:
•Error is thrown on Mobile App.

Event flow:
1. Customer wants to purchase a parking space.
2. Customer attempts to reserve a parking spot
3. Backend notifies app of insufficient funds.
4. App throws error message.