COMPUTERIZED PARKING SYSTEM

Farheen Master

System-wide Constraints:

- Technical Support is out of scope for this project.
- If at any point in time, system fails or malfunctions, system can issue a failsafe warning to Garage employees, which is out of scope for this project.

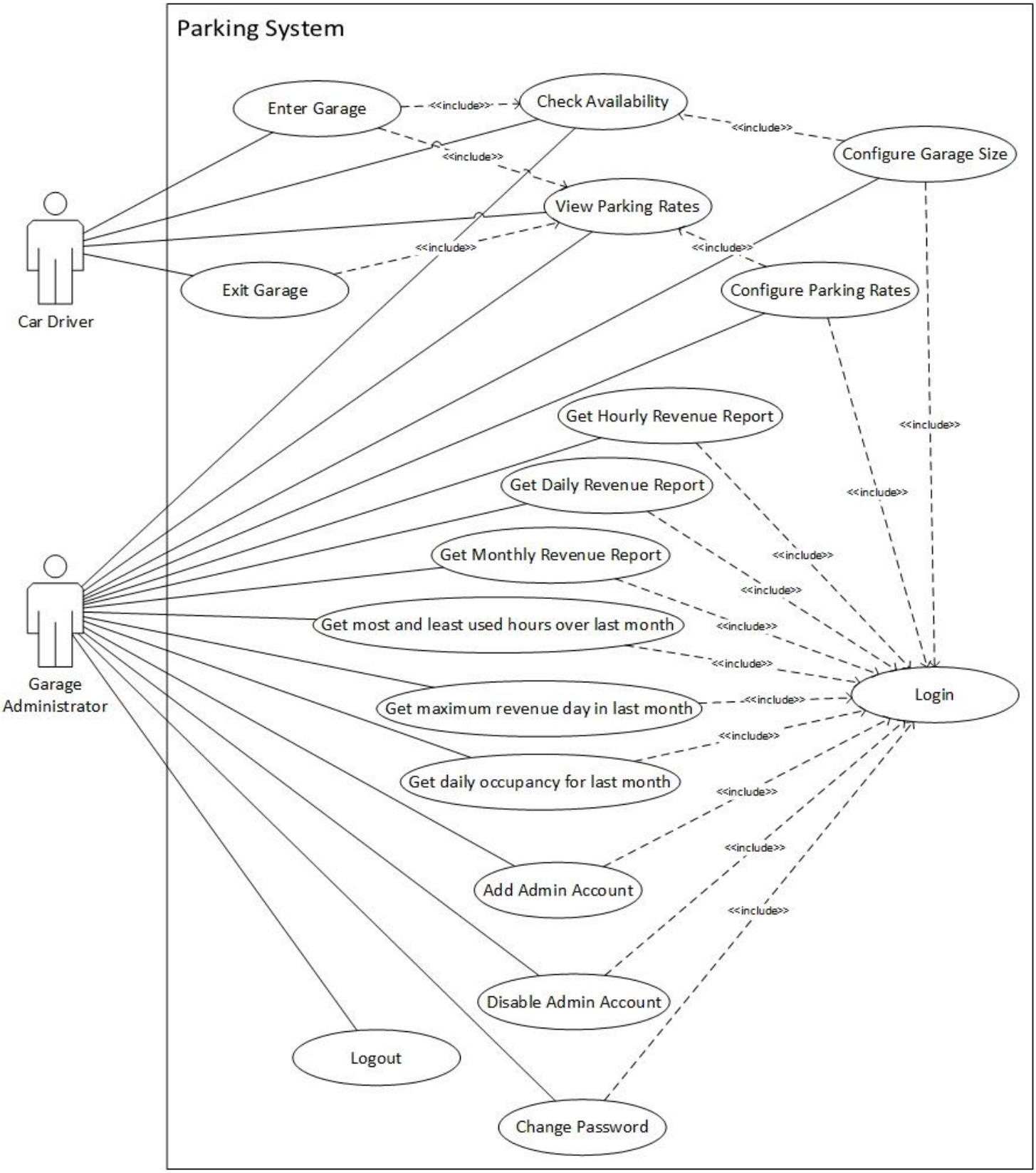
Special Requirements/Required Technology:

- A display screen for displaying the parking availability, parking rates, usage reports, etc.
- A keyboard to enter commands to the system, ticket details, for logging in, etc.
- The system comes with a pre-generated Administrator account with username and password as admin, admin.

Assumptions:

- A car takes up only one parking slot for parking.
- Parking Availability is checked only at the entry gate, not at the exit gate.
- Parking Availability and rates are shown only when prompted.
- If parking rates are changed while car is parked in the garage, the new rates will be used to calculate the parking fee.
- If the Parking Garage is full, the driver has to back out his car.

Use Case Model



Use Case UC1: Enter Garage

Primary Actor: Car Driver
Stakeholders and Interests:
- Car Driver: Wants to park car in a parking spot in the parking garage.
- Garage Administrator: Wants to record the tickets issued for parking.

Success Guarantee (Postconditions): Driver has a valid ticket number and has parked in the garage.

Main Success Scenario (or Basic Flow):
1. Car Driver drives up to the garage entry gate and checks the Parking Availability.
2. Car Driver prompts system for a ticket number.
3. System displays the current garage parking rates.
4. System issues a unique ticket number for the Car Driver.
5. System records the current time as the entry time for the issued ticket.
6. System opens the entry gate, lets the car in and closes the gate.
7. System now has one less available parking slot.

Extensions (or Alternative Flows):
1a. The parking garage is full:
    1. System does not allow driver to get ticket number.
4a. System does not issue ticket number:
    1. The Car Driver can call for Technical Support.
6a. Entry Gate does not open:
    1. The Car Driver can call for Technical Support.
6b. Entry Gate does not close:
    1. The system will issue warning to the Garage employees.

Frequency of Occurrence: Could be nearly continuous.

Use Case UC2: Exit Garage

Primary Actor: Car Driver
Stakeholders and Interests:
- Car Driver: Wants to exit the parking lot.
- Garage Administrator: Wants drivers to pay for parking in full.

Preconditions: The car was parked in the garage.
Success Guarantee (Postconditions): Driver has paid for the parking correctly and the then occupied parking slot is now available.

Main Success Scenario (or Basic Flow):
1. Car Driver drives up to the exit gate and views the parking rates.
2. System prompts Car Driver to enter correct ticket number.
3. Driver enters ticket number.
4. System calculates parking fee for that ticket and displays the amount.
5. Driver enters the parking fee.
6. System opens the gate, lets the car leave and closes the gate.
7. System records the current time as the exit time for the ticket.
8. System makes one more parking slot available.

Extensions (or Alternative Flows):

2a. System does not prompt for ticket number:
    1. The Car Driver can call for Technical Support.

3a. Driver has lost ticket:
    1. System prompts Driver to pay the parking fee for the entire day.

3b. Driver enters an already used ticket number:
    1. System prompts Driver to try again or call for Technical Support.

3c. Driver enters invalid ticket number:
    1. System prompts Driver to try again or call for Technical Support.

5a. Driver wants to pay by cash:
    1. Driver enters cash amount.
        1a. Driver has entered more cash amount than the parking fee.
            1. System returns the change.
        1b. Driver has entered less cash than the parking fee.
            1. System prompts for the remaining amount.
            Step 5 is repeated until remaining amount is zero.

5b. Driver wants to pay by credit card:
    1. Driver enters 16 digit credit card number, 3 digit security code, expiry month and year.
    2. System checks if credit card number is 16 digit long, security code is 3 digit long and month and year are valid and in the future.
        2a. Credit Card details are not valid:
            1. System prompts for the remaining amount.
            Step 5 is repeated until remaining amount is zero.
        2b. Credit card details are valid.
            1. System charges the predetermined parking fee to the driver's credit card.

5c. Driver does not have money to pay:
    1. System prompts Driver to enter name, driver's license number and records the exception.

6a. Exit Gate does not open:
    1. The Car Driver can call for Technical Support.

6b. Exit Gate does not close:
    1. The system will issue warning to the Garage employees.

Technology and Data Variations List:

5. Cash amount and Credit card details are entered using a keyboard.

Frequency of Occurrence: Could be nearly continuous.

Open Issues:

- Credit cards cannot be verified by a payment authorization system as it is out of scope for this project.
- If Driver does not have money to pay, system will record his name and driver's license number. Any other process required is out of scope for this project.

Use Case UC3: Check Availability

Primary Actor: Car Driver, Garage Administrator

Stakeholders and Interests:

- Car Driver: Wants to find out if a parking slot is available.

- Garage Administrator: Wants to find out if a parking slot is available.

Preconditions:

Success Guarantee (Postconditions): Car Driver / Garage Administrator has determined if a parking slot is available or not.

Main Success Scenario (or Basic Flow):

1. Car Driver / Garage Administrator drives up to or walks up to the entry gate and prompts for system to display garage availability.
2. System displays the number of available parking slots out of the total parking slots in the garage.

Frequency of Occurrence: Could be nearly continuous.

Use Case UC4: View Parking Rates

Primary Actor: Car Driver, Garage Administrator

Stakeholders and Interests:

- Car Driver: Wants to view current parking rates.
- Garage Administrator: Wants to view current parking rates.

Success Guarantee (Postconditions): Car Driver / Garage Administrator has viewed the current parking rates.

Main Success Scenario (or Basic Flow):

1. Car Driver / Garage Administrator drives up to or walks up to the entry or exit gate and prompts for system to display current parking rates.
2. System displays the current parking rates for the garage.

Frequency of Occurrence: Could be nearly continuous.

Use Case UC5: Login

Primary Actor: Garage Administrator

Stakeholders and Interests:

- Garage Administrator: Wants to log in to the parking system.

Success Guarantee (Postconditions): Garage Administrator has successfully logged into the system and has access to the parking records.

Main Success Scenario (or Basic Flow):

1. Garage Administrator prompts system to display the login screen.
2. System prompts Garage Administrator to enter username and password.
3. Garage Administrator enters username and password.
4. System checks if username and password are valid.
5. Garage Administrator is logged into the parking system.

Extensions (or Alternative Flows):

3a. Garage Administrator has forgotten password:
    1. System resets account for the Garage Administrator and sets the default password.

4a. Username and password combination is incorrect:

1. System displays error and prompts Garage Administrator to try again.

Technology and Data Variations List:

3. The system does not allow numbers, spaces or special characters to be entered for username or password.

Open Issues:

- Any other authorization process for the Garage Administrator is out of scope for the project.
- Emailing Garage Administrator to reset password is out of scope for this project.

Use Case UC6: Logout

Primary Actor: Garage Administrator
Stakeholders and Interests:

- Garage Administrator: Wants to log out of the parking system.

Success Guarantee (Postconditions): Garage Administrator has successfully logged out of the system.

Main Success Scenario (or Basic Flow):

1. Garage Administrator logs out of the parking system.
2. System displays message that Garage Administrator has successfully logged out.

Open Issues:

- Any other authorization process for the Garage Administrator is out of scope for the project.

Use Case UC7: Configure Garage Size

Primary Actor: Garage Administrator
Stakeholders and Interests:

- Garage Administrator: Wants to change number of parking slots for the garage.

Preconditions: Garage Administrator is identified and authenticated into the system.
Success Guarantee (Postconditions): Garage Administrator has successfully changed the total number of parking slots in the parking garage.

Main Success Scenario (or Basic Flow):

1. Garage Administrator views the current parking availability.
2. System prompts Garage Administrator to configure number of parking slots in the garage.
3. Garage Administrator configures the parking garage size.

Extensions (or Alternative Flows):

3a. New size is zero:
    1. System displays error that parking size cannot be zero and prompts Garage Administrator to try again.
3b. New size is negative:

1. System displays error that parking size cannot be negative and prompts Garage Administrator to try again.

3c. New size is less than number of occupied slots:
1. System displays error that parking size cannot be less than number of occupied parking slots and prompts Garage Administrator to try again.

3d. New size is same as before:
1. System acknowledges no change and does not make any changes.

Use Case UC8: Configure Parking Rates

Primary Actor: Garage Administrator
Stakeholders and Interests:
- Garage Administrator: Wants to change parking rates.

Preconditions: Garage Administrator is identified and authenticated into the system.
Success Guarantee (Postconditions): Garage Administrator has successfully changed the parking rates.

Main Success Scenario (or Basic Flow):
1. Garage Administrator views the current parking rates.
2. System prompts Garage Administrator to configure parking rates.
3. Garage Administrator configures the parking rates for 30 minutes, 1 hour, 2 hours, 10 hours and full day.

Extensions (or Alternative Flows):
3a. Garage Administrator enters negative amounts.
1. System displays error and prompts Garage Administrator to try again.

Open Issues:
- System does not check if the parking rates entered are logically correct, as it is out of scope for the project.

Use Case UC9: Get Hourly Revenue Report

Primary Actor: Garage Administrator
Stakeholders and Interests:
- Garage Administrator: Wants to find out revenue generated by the parking garage hourly for a particular day.

Preconditions: Garage Administrator is identified and authenticated into the system.
Success Guarantee (Postconditions): Garage Administrator has successfully viewed correct hourly revenues for the particular day.

Main Success Scenario (or Basic Flow):
1. Garage Administrator enters a particular date to find out hourly revenues for that day.
2. System displays revenue in dollars for each hour of that day.

Extensions (or Alternative Flows):
1a. Date is in future:
1. System displays error and prompts Garage Administrator to try again.

Use Case UC10: Get Daily Revenue Report

Primary Actor: Garage Administrator
Stakeholders and Interests:
- Garage Administrator: Wants to find out revenue generated by the parking garage daily for a particular month.

Preconditions: Garage Administrator is identified and authenticated into the system.
Success Guarantee (Postconditions): Garage Administrator has successfully viewed correct daily revenues for the particular month.

Main Success Scenario (or Basic Flow):
1. Garage Administrator enters a particular month to find out daily revenues for that month.
2. System displays revenue in dollars for each day of that month.

Extensions (or Alternative Flows):
1a. Month is in future:
    1. System displays error and prompts Garage Administrator to try again.

Use Case UC11: Get Monthly Revenue Report

Primary Actor: Garage Administrator
Stakeholders and Interests:
- Garage Administrator: Wants to find out revenue generated by the parking garage monthly for a particular year.

Preconditions: Garage Administrator is identified and authenticated into the system.
Success Guarantee (Postconditions): Garage Administrator has successfully viewed correct monthly revenues for the particular year.

Main Success Scenario (or Basic Flow):
1. Garage Administrator enters a particular year to find out monthly revenues for that year.
2. System displays revenue in dollars for each month of that year.

Extensions (or Alternative Flows):
1a. Year is in future:
    1. System displays error and prompts Garage Administrator to try again.

Use Case UC12: Get most and least used hours over last month

Primary Actor: Garage Administrator
Stakeholders and Interests:
- Garage Administrator: Wants to find out hours with highest and lowest occupancy over last month.

Preconditions: Garage Administrator is identified and authenticated into the system.
Success Guarantee (Postconditions): Garage Administrator has successfully viewed which hours had highest and lowest occupancy with their occupancy percentages over the last month.

Main Success Scenario (or Basic Flow):

1. Garage Administrator enters command into the system to find out the most and least used hours for the parking garage over the last month.
2. System displays the hours that had highest and lowest occupancy with their occupancy percentages over the last month.

Use Case UC13: Get maximum revenue day in last month

Primary Actor: Garage Administrator
Stakeholders and Interests:

- Garage Administrator: Wants to find out which day had the most revenue in the last month.

Preconditions: Garage Administrator is identified and authenticated into the system.
Success Guarantee (Postconditions): Garage Administrator has successfully viewed which day had the most revenue in the last month.

Main Success Scenario (or Basic Flow):

1. Garage Administrator enters command into the system to find out which day had the most revenue in the last month.
2. System displays the day which had the most revenue in the last month with the revenue in dollars.

Use Case UC14: Get daily occupancy for last month

Primary Actor: Garage Administrator
Stakeholders and Interests:

- Garage Administrator: Wants to find out the daily occupancy for the last month.

Preconditions: Garage Administrator is identified and authenticated into the system.
Success Guarantee (Postconditions): Garage Administrator has successfully viewed daily occupancy for the last month.

Main Success Scenario (or Basic Flow):

1. Garage Administrator enters command into the system to find out daily occupancy for the last month.
2. System displays daily occupancy count for each day in the last month.

Use Case UC15: Add Admin Account

Primary Actor: Garage Administrator
Stakeholders and Interests:

- Garage Administrator: Wants to add account for a garage Administrator.

Preconditions: Garage Administrator is identified and authenticated into the system.
Success Guarantee (Postconditions): Garage Administrator has successfully added a Garage Administrator account into the system.

Main Success Scenario (or Basic Flow):

1. Garage Administrator enters command into the system to add another Garage Administrator account.
2. System prompts Garage Administrator to enter username.
3. Garage Administrator enters a username.
4. System creates a Garage Administrator account with the username and default password.

Extensions (or Alternative Flows):

3a. Username already exists in the system:
   1. System prompts Garage Administrator to enter another username.

Special Requirements:

- The system comes with a pre-generated Administrator account with username and password as admin, admin.

Technology and Data Variations List:

3. The system does not allow numbers, spaces or special characters to be entered for username or password.

Open Issues:

- Any Garage Administrator can create any other Garage Administrator account. Higher privileges are not provided to select accounts as this is out of scope for this project.

Use Case UC16: Disable Admin Account

Primary Actor: Garage Administrator
Stakeholders and Interests:

- Garage Administrator: Wants to disable account for a Garage Administrator.

Preconditions: Garage Administrator is identified and authenticated into the system.
Success Guarantee (Postconditions): Garage Administrator has successfully disabled the Garage Administrator account in the system.

Main Success Scenario (or Basic Flow):

1. Garage Administrator enters command into the system to disable a Garage Administrator account.
2. System prompts Garage Administrator to enter username.
3. Garage Administrator enters a username.
4. System disables the Garage Administrator account with that username and records username of the Garage Administrator who authorized it.

Extensions (or Alternative Flows):

3a. Username does not exist in the system:
   1. System prompts Garage Administrator to enter another username or cancel.

3b. Garage Administrator enters own username:
   1. System prompts Garage Administrator to enter another username or cancel.

Special Requirements:

- The system comes with a pre-generated Administrator account with username and password as admin, admin.

Technology and Data Variations List:

3. The system does not allow numbers, spaces or special characters to be entered for username or password.

Open Issues:

- Any Garage Administrator can disable any other Garage Administrator account. Higher privileges are not provided to select accounts as this is out of scope for this project.

Use Case UC17: Change Password

Primary Actor: Garage Administrator
Stakeholders and Interests:

- Garage Administrator: Wants to change password for his account.

Preconditions: Garage Administrator is identified and authenticated into the system.
Success Guarantee (Postconditions): Garage Administrator has successfully changed his account password into the system.

Main Success Scenario (or Basic Flow):

1. System prompts Garage Administrator to enter old password.
2. Garage Administrator enters old password.
3. System prompts Garage Administrator to enter new password.
4. Garage Administrator enters new password.
5. System prompts Garage Administrator to re-enter the new password.
6. Garage Administrator re-enters the new password.
7. System changes the account password for the Garage Administrator.

Extensions (or Alternative Flows):

2a. Password is incorrect:
    1. System prompts Garage Administrator to re-enter password or Exit.

6a. Password re-entered does not match the new password entered previously:
    1. System prompts Garage Administrator to re-enter password or Exit.

Technology and Data Variations List:

2, 4, 6. The system does not allow numbers, spaces or special characters to be entered for username or password.