Group #: 4
Group Member: Eric Li, Toby Zhu, Gloria Zhang

Phase 3: Database Implementation and Testing

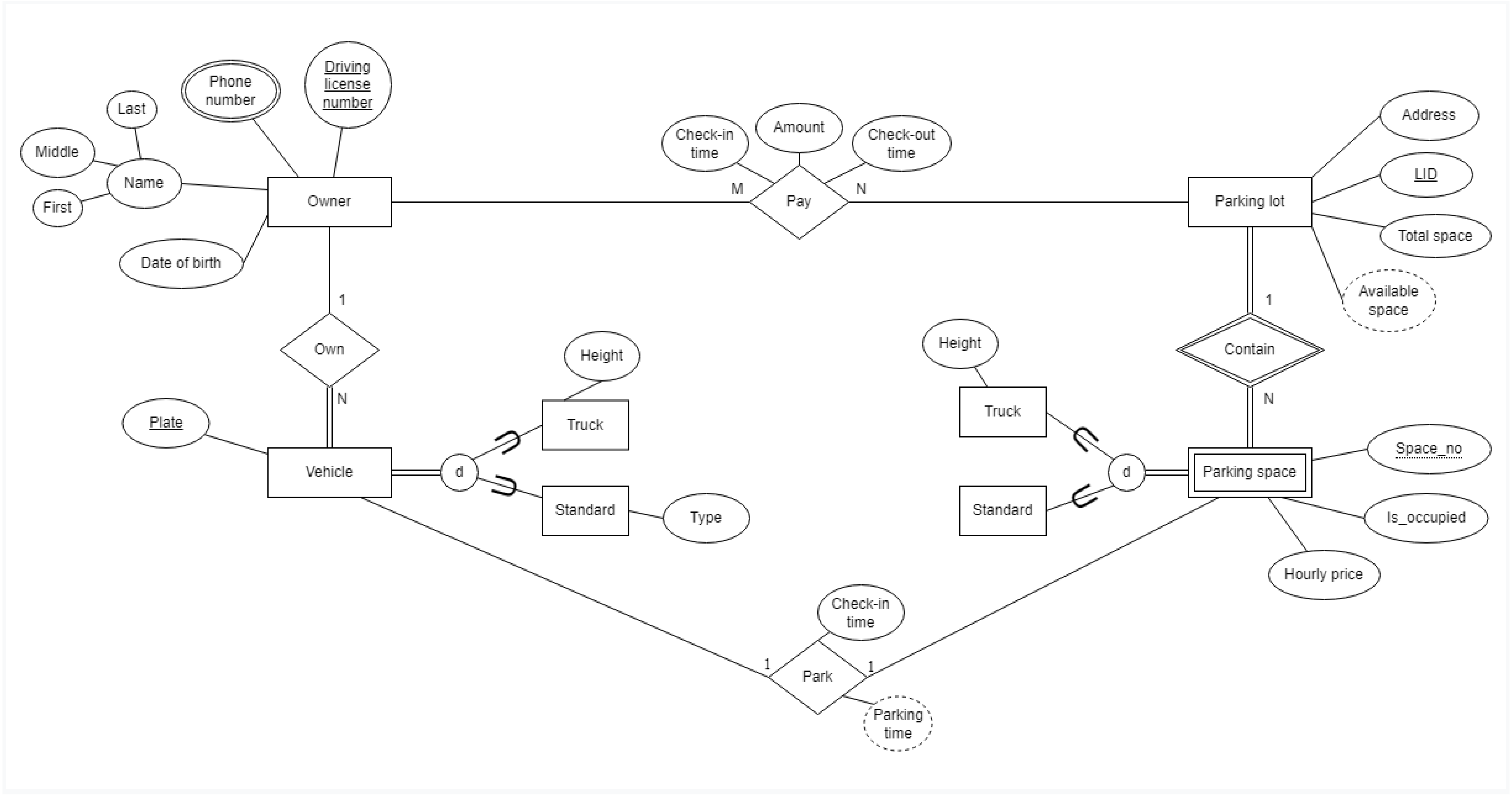
1. Problem Statement

Parking is an integral component of human lives, and how to park efficiently and conveniently has become a huge societal problem to solve. There have been long-lasting complaints about the cumbersome procedure one has to go through to park somewhere. An intelligent database system can reduce this complication. Imagine if you may park in any parking lot without the manual burden of grabbing a ticket and paying every time. The smart parking system that we plan to design is exactly what you will need. The application allows users to see the available spaces in all parking lots, park and check out their vehicles via a simple click, and pay for each parking session automatically. In this app, a database system will store the user and vehicle information, as well as the up-to-date status of all parking lots and parking spaces. Moreover, it will also save the user's parking and payment history for administrative and tracing purposes.

2. Conceptual Database Design

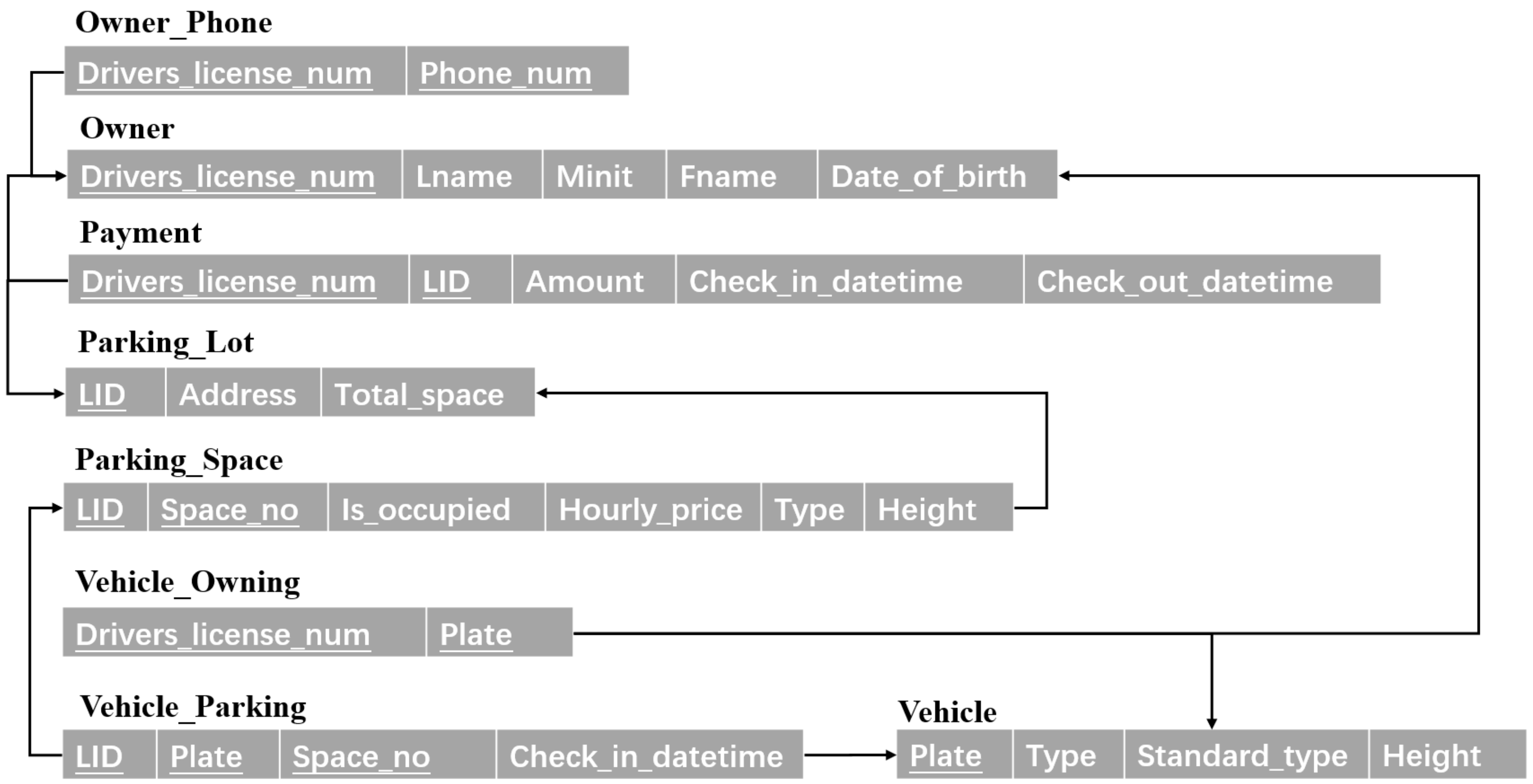
Assumptions:

1. The vehicle owner must have a driving license in order to park in the lot.
2. A vehicle can only be registered with and parked in one parking space at a time.
3. A vehicle can only be registered under one owner.
4. Vehicles can only be truck or standard; there are only parking spaces for trucks and standard vehicles.
5. Parking lots and parking spaces can be empty.



3. Logical Database Design

- Relation Mapping:



- Attribute Metadata: **All attributes below in all tables maintain the domain constraint**.

Owner

Attribute	Description	Data Type	Integrity Constraint
<u>Driver_license_num</u>	The drivers license number of the vehicle owner.	Variable length string	Key constraint Entity integrity constraint
Lname	The last name of the vehicle owner.	Variable length string	NOT_NULL constraint
Minit	The middle initial of the vehicle owner.	1 character	
Fname	The first name of the vehicle owner.	Variable length string	NOT_NULL constraint
Date_of_birth	The date of birth of the vehicle owner.	Variable length string	NOT_NULL constraint

Owner_Phone

Attribute	Description	Data Type	Integrity Constraint
<u>Phone_num</u>	The phone number of the vehicle owner.	Integer	Key constraint Entity integrity constraint
<u>Driver_license_num</u>	The driving license number of the vehicle owner.	Variable length string	Key constraint Entity integrity constraint Referential integrity constraint

Parking_Lot

Attribute	Description	Data Type	Integrity Constraint
<u>LID</u>	The parking lot ID that uniquely identifies each parking lot.	Variable length string	Key constraint Entity integrity constraint
Address	The address of a parking lot.	Variable length string	NOT_NULL constraint
Total_space	The total number of spaces in a parking lot.	Integer	NOT_NULL constraint

Parking_Space

Attribute	Description	Data Type	Integrity Constraint
<u>LID</u>	The parking lot ID that uniquely identifies each parking lot.	Variable length string	Key constraint Entity integrity constraint Referential integrity constraint
<u>Space_no</u>	The space number that uniquely identifies each parking space within a parking lot.	Integer	Key constraint Entity integrity constraint
Type	The type of vehicle - standard or truck.	Variable length string	NOT_NULL constraint Domain constraint: “Standard”, “Truck”
Is_occupied	Indicate whether the space is occupied by a truck.	Boolean	NOT_NULL constraint
Hourly_price	The price to park an hour in this parking space ($).	Integer	NOT_NULL constraint
Height	The height of this parking space (ft).	Integer	

Vehicle

Attribute	Description	Data Type	Integrity Constraint
<u>Plate</u>	The plate number of a vehicle.	Variable length string	Key constraint Entity integrity constraint
Type	The type of a vehicle - standard or truck.	Variable length string	NOT_NULL constraint Domain constraint: “Standard”, “Truck”
Standard_type	The type of a standard vehicle - compact or noncompact.	Variable length string	Domain constraint: “Compact”, “Noncompact”
Height	The height of a truck.	Integer	

Payment

Attribute	Description	Data Type	Integrity Constraint
<u>Driver_license_num</u>	The driving license number of the vehicle owner.	Variable length string	Key constraint Entity integrity constraint Referential integrity constraint
<u>LID</u>	The parking lot ID that uniquely identifies each parking lot.	Variable length string	Key constraint Entity integrity constraint Referential integrity constraint
Amount	The amount of a payment.	Integer	NOT_NULL constraint
Check_in_datetime	The datetime when the vehicle started parking.	DateTime	
Check_out_datetime	The datetime when the vehicle left the parking lot.	DateTime	

Vehicle_Owning

Attribute	Description	Data Type	Integrity Constraint
<u>Driver_license_num</u>	The driving license number of the vehicle owner.	Variable length string	Key constraint Entity integrity constraint Referential integrity constraint
<u>Plate</u>	The plate number of a vehicle.	Variable length string	Key constraint Entity integrity constraint Referential integrity constraint

Vehicle_Parking

Attribute	Description	Data Type	Integrity Constraint
<u>Plate</u>	The plate number of a vehicle.	Variable length string	Key constraint Entity integrity constraint Referential integrity constraint
<u>LID</u>	The parking lot ID that uniquely identifies each parking lot.	Variable length string	Key constraint Entity integrity constraint Referential integrity constraint
<u>Space_no</u>	The space number that uniquely identifies each parking space within a parking lot.	Integer	Key constraint Entity integrity constraint Referential integrity constraint
Check_in_datetime	The datetime when the vehicle started parking.	DateTime	

4. Application Program Design

1. Get_all_lots
   Description: This function displays all parking lot ids.
   Steps: Retrieve and return “LID” attribute of all records from table “Parking_Lot.”

2. List_space
   Description: This function displays space information given the specific parking lot.
   Input: parking lot id (LID)
   Steps: Retrieve and return “Space_no,” “Type”, “Is_occupied” attributes of all records from table “Parking_Space” with the given parking lot id (LID).

3. Compute_idle
   Description: This function computes the number of unoccupied parking spaces given the specific parking lot.
   Input: parking lot id (LID)
   Steps:
   a. Count the number of records in table “Parking_Space” with the given parking lot id (LID) and not occupied.

4. Check_owner
   Description: This function checks if a specific owner exists in the database.
   Input: License_number
   Steps: Retrieve and return “Drivers_license_num,” “Lname,” “Minit,” “Fname,” “Date_of_birth” attributes of all records from the table “Owner” with License_num.

5. Check_vehicle
   Description: This function checks if a specific vehicle exists in the database.
   Input: Plate number
   Steps: Retrieve and return all records from the table “Vehicle” with the given plate number (Plate).

6. Check_parking
   Description: This function checks if a specific vehicle is parked somewhere.

Input: Plate number
Steps: Retrieve and return all records from the table “Vehicle_Parking” with the given plate number.

7. Check_parking_vehicle
Description: This function returns the parked vehicle plate number and the check-in time given a specific occupied parking space.
Input: parking lot id (LID), Space_no
Steps: Retrieve and return “Plate,” “Check_in_date” attributes of all records from the table “Vehicle_Parking” with the given LID and Spac_no.

8. Register_owner
Description: This function inserts information of a vehicle owner into the database.
Input: License number, first name, last name, middle initial, phone number, date of birth
Steps:
    a. Insert the tuple Ti = <License number, last name, middle initial, first name, date of birth> into the table “Owner”.
    b. Insert the tuple Ti = <License number, phone number> into the table “Owner_Phone”.

9. Register_vehicle
Description: This function inserts information of a vehicle into the database.
Input: Plate number, vehicle type, height
Steps: Insert the tuple Ti = <Plate number, vehicle height, height=NULL> into the table “Vehicle”.

10. Get_owner
Description: This function returns the drivers’ license number of a given vehicle
Input: Plate number
Steps: Retrieve and return “Drivers_license_number” attribute of all records from table “Vehicle_Owning” with the given plate number.

11. Park
Description: The function updates the space status for the selected unoccupied parking space and parks the vehicle.
Input: Parking lot id (LID), parking space number, vehicle plate number

Precondition: The input owner exists in the database (guaranteed through Check_owner and Register_owner). The input vehicle exists in the database (guaranteed through Check_vehicle and Register_vehicle). The vehicle is unparked (guaranteed through Check_parking).
Steps:
   a. Reserve the selected space in the corresponding parking space table by updating the Is-occupied attribute to true.
   b. Record the parking time and create the parking record = <LID, Plate, Check_in_date, Space_no>.
   c. Insert the parking record into the "Vehicle_Parking" table.

12. Get_price
    Description: The function returns the hourly price of a specific parking space
    Input: Parking lot id (LID), parking space number
    Steps:
       a. Retrieve and return "Hourly_price" attribute of all records from table "Parking_space" with the LID and parking space number.

13. Check_out
    Description: This function computes the parking price given an occupied parking space, inserts payment information into the database, updates the parking space status, and deletes the corresponding parking tuple.
    Input: Parking lot id (LID), parking space number, plate number, check in time, payment amount
    Precondition: Payment amount is already calculated using Get_price. Vehicle plate number and check in time are retrieved through Check_parking_vehicle. Vehicle owner's diverse license number is retrieved using Get_owner.
    Steps:
       a. Insert the payment record <Drivers_license_num, LID, Check_in_date, Check_out_date, Amount>into the "Payment" table.
       b. Updating the Is-occupied attribute of the parking space to false.
       c. Delete the parking record from the "Vehicle_Parking" table given LID, space_no, and plate.

14. List_owner_detail
    Description: The function displays owner and vehicle information for all owners whose drivers license number is similar to the user's input.
    Input: Drivers license number
    Steps:
       a. (Left) Join the "Owner" table with the "Owner_Phone" table with the same License number.

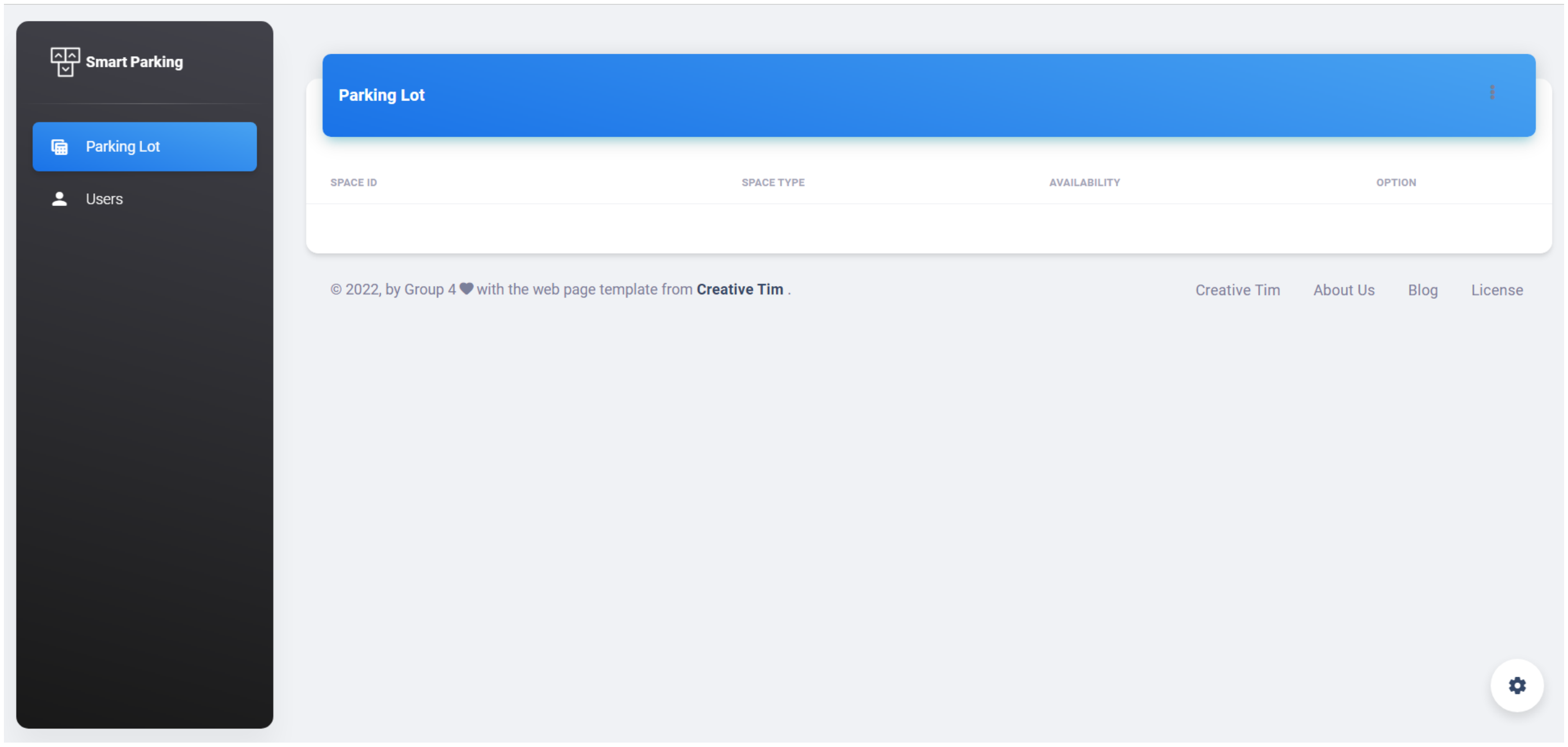
b. (Left) Join the resulting table from above with the “Vehicle_Owning” table with the same License number.
c. (Left) Join the resulting table from above with the “Vehicle” table with the same Plate number.
d. Left) Join the resulting table from above with the “Vehicle_Parking” table with the same Plate number. Rename the new table as “Owner_Detail”.
e. Retrieve and display all tuples from “Owner_Detail” whose Driver's license number is LIKE the user’s input.

15. List_payment_history
Description: The function lists all payment information of the given owner.
Input: License number
Steps: Retrieve and display all tuples with License number from table “Payment”.

5. User Manual

Github Repository: https://github.com/zxllxz2/smart-parking-system

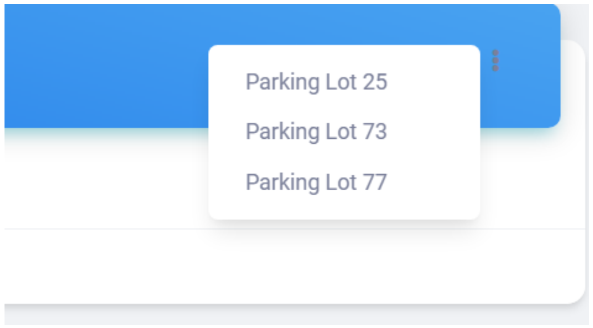
Parking Lot Interface



SPACE ID	SPACE TYPE	AVAILABILITY	OPTION

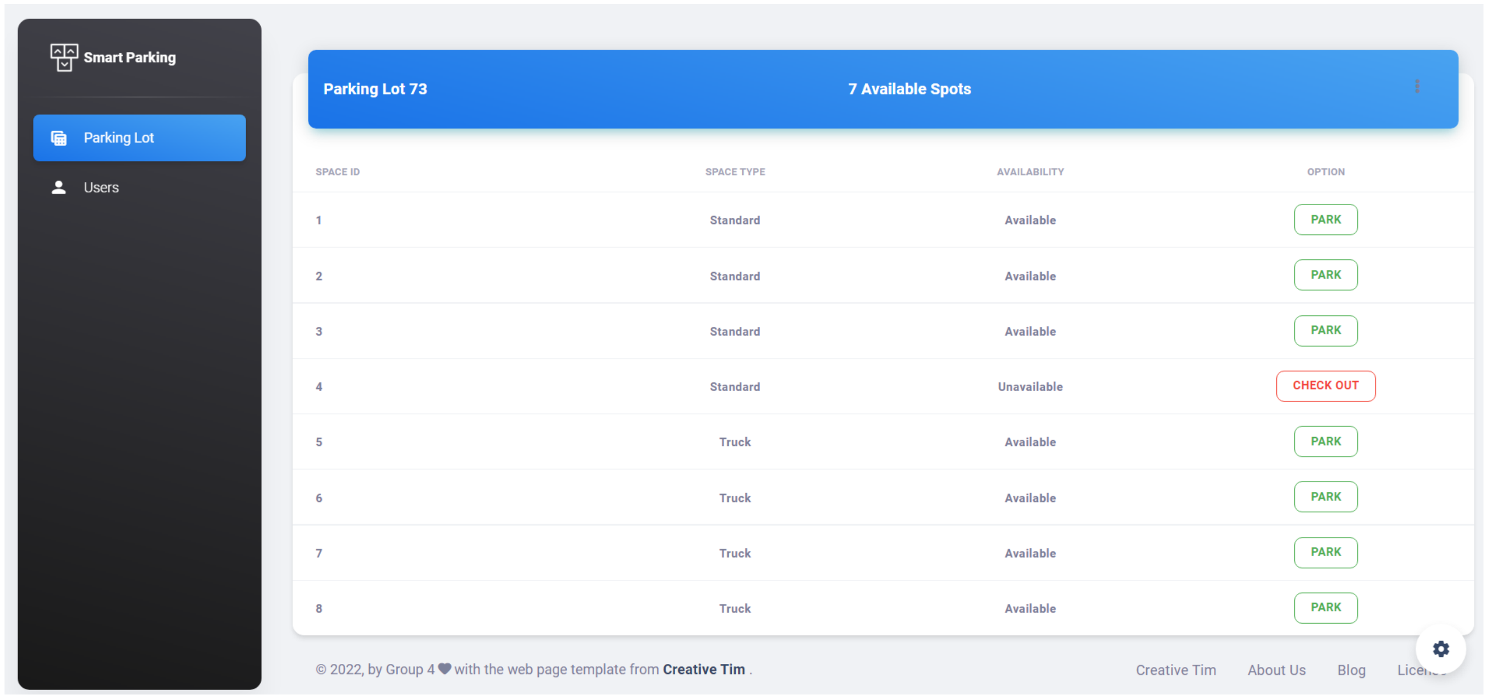
© 2022, by Group 4 ♥ with the web page template from **Creative Tim** .

- The function **get_all_lots** retrieves all parking lot information.



-

- Select any parking lot. The function **list_space** will list all parking spaces in this parking lot, and the function **compute_idle** will show the number of unoccupied spaces in this lot.



Parking Lot 73 7 Available Spots

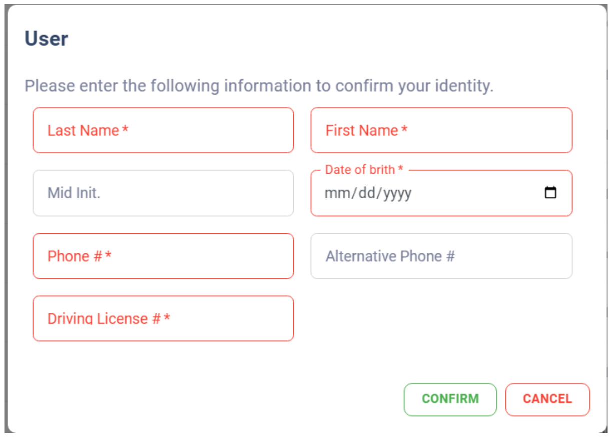
SPACE ID	SPACE TYPE	AVAILABILITY	OPTION
1	Standard	Available	PARK
2	Standard	Available	PARK
3	Standard	Available	PARK
4	Standard	Unavailable	CHECK OUT
5	Truck	Available	PARK
6	Truck	Available	PARK
7	Truck	Available	PARK
8	Truck	Available	PARK

© 2022, by Group 4 ♥ with the web page template from **Creative Tim** .

-

Park a vehicle

- Click the "parking" button to start parking the vehicle. The pop-up window will prompt for the user information.

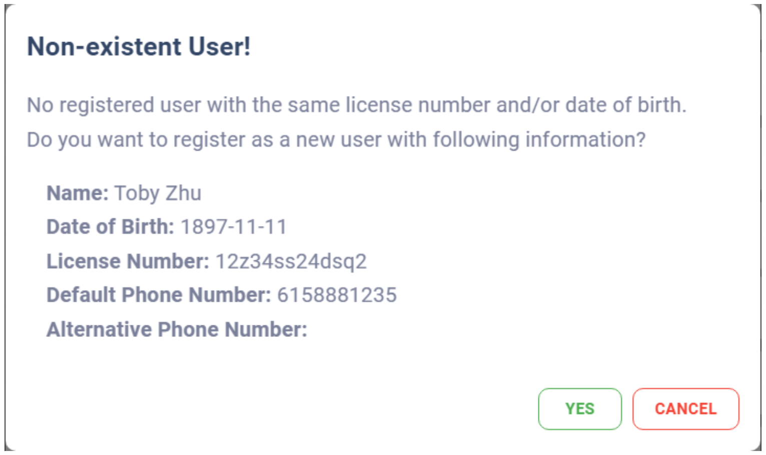


User

Please enter the following information to confirm your identity.

Last Name *	First Name *
Mid Init.	Date of brith * mm/dd/yyyy
Phone # *	Alternative Phone #
Driving License # *	

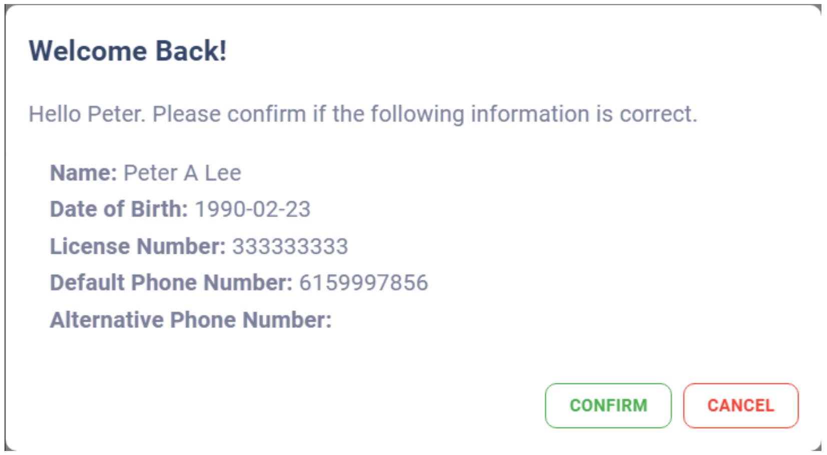
- According to the given information, the function **check_owner** will check the existence of this vehicle in the database. If the user does not exist in the database, the function **register_owner** will insert the new user information into the database. Otherwise, the system will display the information retrieved via the function check_owner.



Non-existent User!

No registered user with the same license number and/or date of birth. Do you want to register as a new user with following information?

Name: Toby Zhu
Date of Birth: 1897-11-11
License Number: 12z34ss24dsq2
Default Phone Number: 6158881235
Alternative Phone Number:

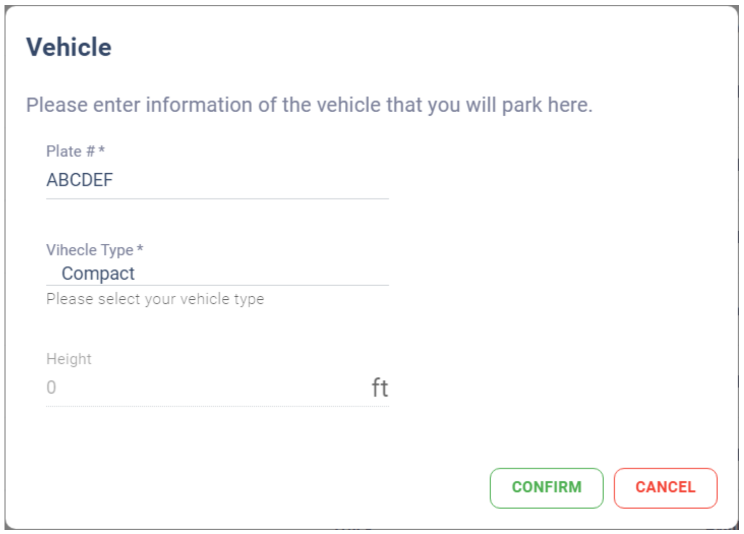


Welcome Back!

Hello Peter. Please confirm if the following information is correct.

Name: Peter A Lee
Date of Birth: 1990-02-23
License Number: 333333333
Default Phone Number: 6159997856
Alternative Phone Number:

- After the user's confirmation, the system will prompt for vehicle information. If the input vehicle is a truck, the system will further ask for its height information.

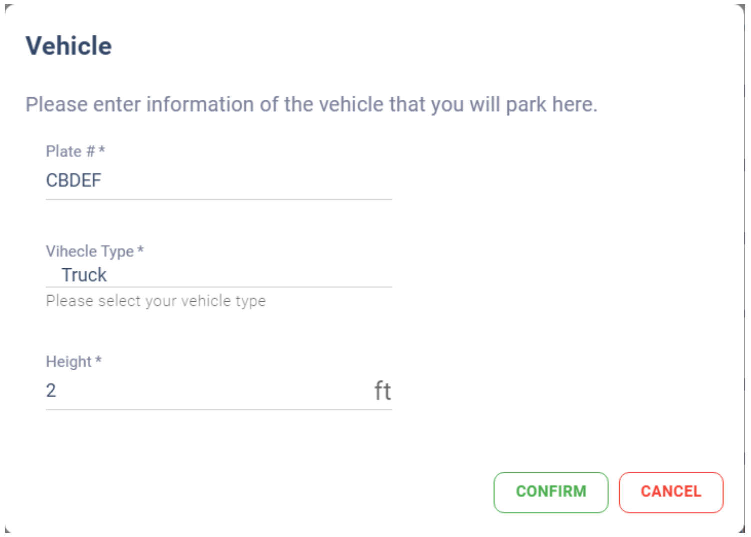

Vehicle

Please enter information of the vehicle that you will park here.

Plate # *
ABCDEF

Vihecle Type *
Compact
Please select your vehicle type

Height
0 ft


Vehicle

Please enter information of the vehicle that you will park here.

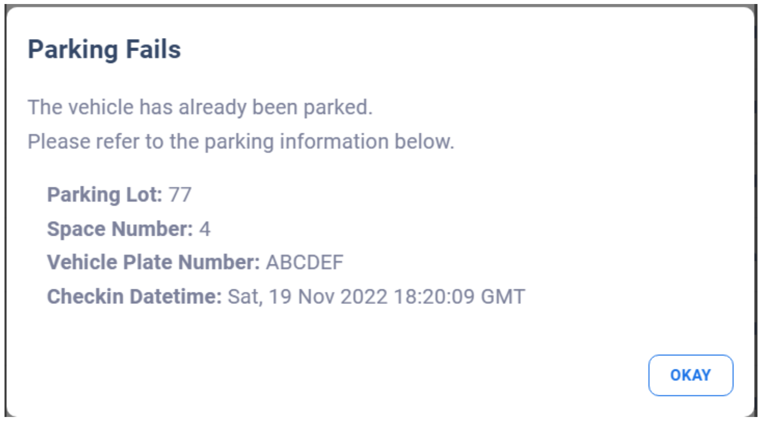
Plate # *
CBDEF

Vihecle Type *
Truck
Please select your vehicle type

Height *
2 ft

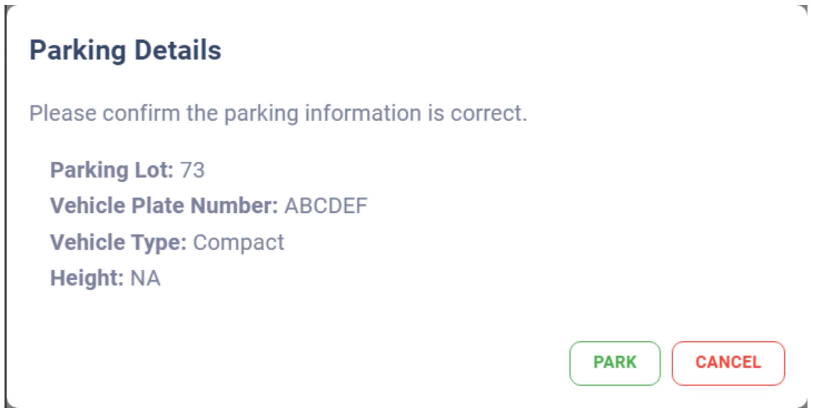
-

- The system will then call the function **check_vehicle** to check the existence of this vehicle in the database. If not found, the system will register the vehicle via calling the function **register_vehicle**. Otherwise, the function **check_parking** will check if the vehicle has already parked in any parking lots. If the vehicle is found to be parked somewhere, the system will display the parking information returned by check_parking. Otherwise, the system will generate a review for the parking information.


Parking Fails

The vehicle has already been parked.
Please refer to the parking information below.

Parking Lot: 77
Space Number: 4
Vehicle Plate Number: ABCDEF
Checkin Datetime: Sat, 19 Nov 2022 18:20:09 GMT

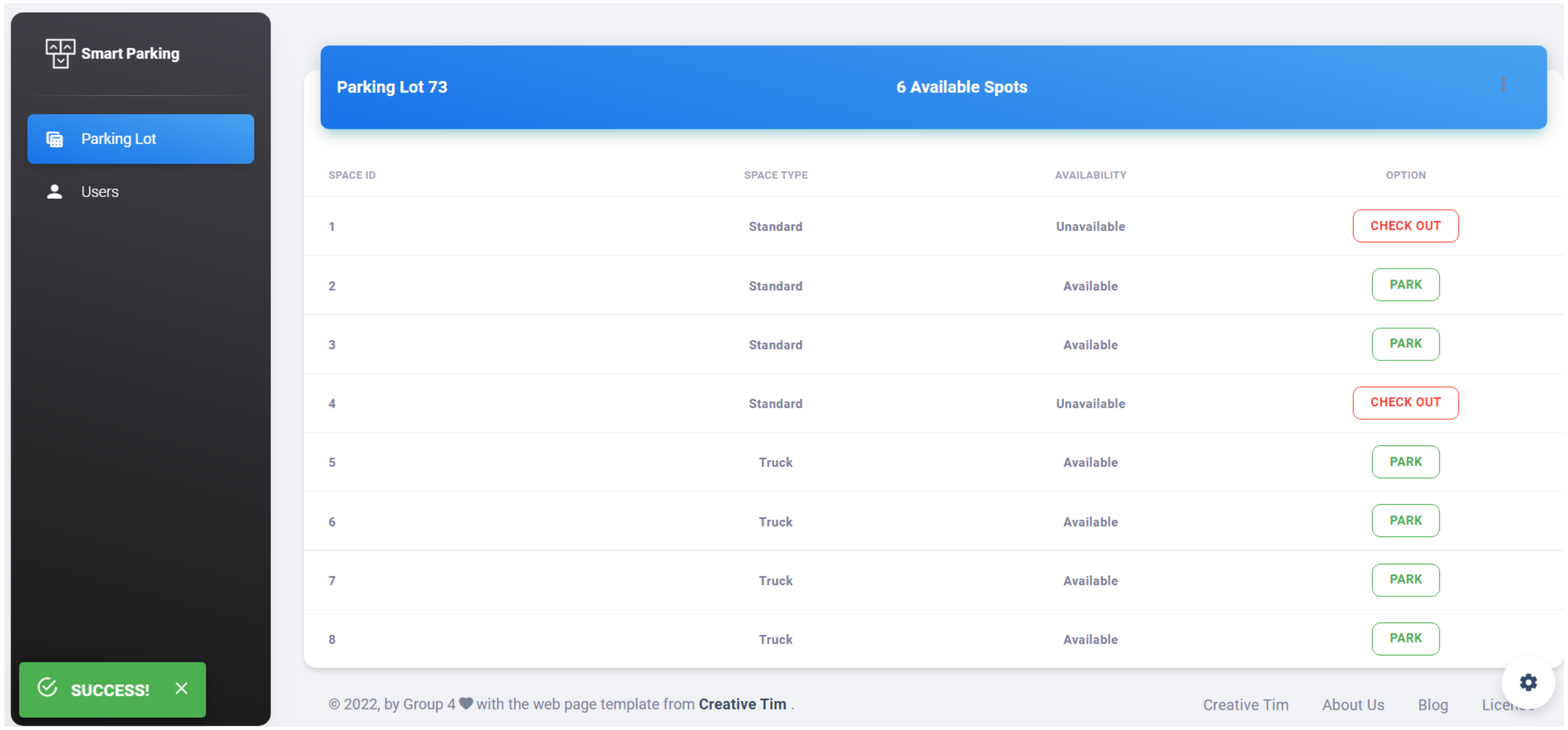

Parking Details

Please confirm the parking information is correct.

Parking Lot: 73
Vehicle Plate Number: ABCDEF
Vehicle Type: Compact
Height: NA

-

- After confirming the parking action, the system will call the function **park** to park the car. It will update the parking space occupying status and insert the parking record into the database. The system will also call the function **compute_idle** to update the occupancy status of this parking lot.



Smart Parking

Parking Lot

Users

SUCCESS!

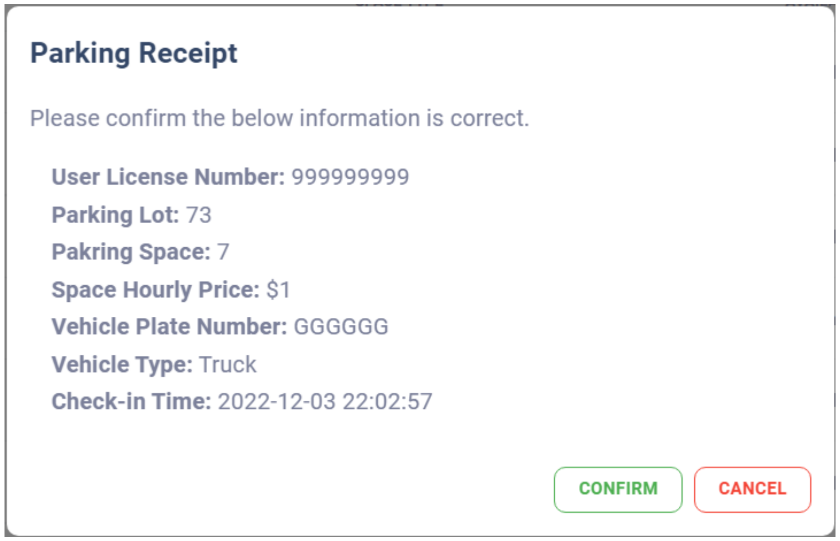
Parking Lot 73 6 Available Spots

SPACE ID	SPACE TYPE	AVAILABILITY	OPTION
1	Standard	Unavailable	CHECK OUT
2	Standard	Available	PARK
3	Standard	Available	PARK
4	Standard	Unavailable	CHECK OUT
5	Truck	Available	PARK
6	Truck	Available	PARK
7	Truck	Available	PARK
8	Truck	Available	PARK

© 2022, by Group 4 ♥ with the web page template from **Creative Tim** .

Creative Tim About Us Blog Licen

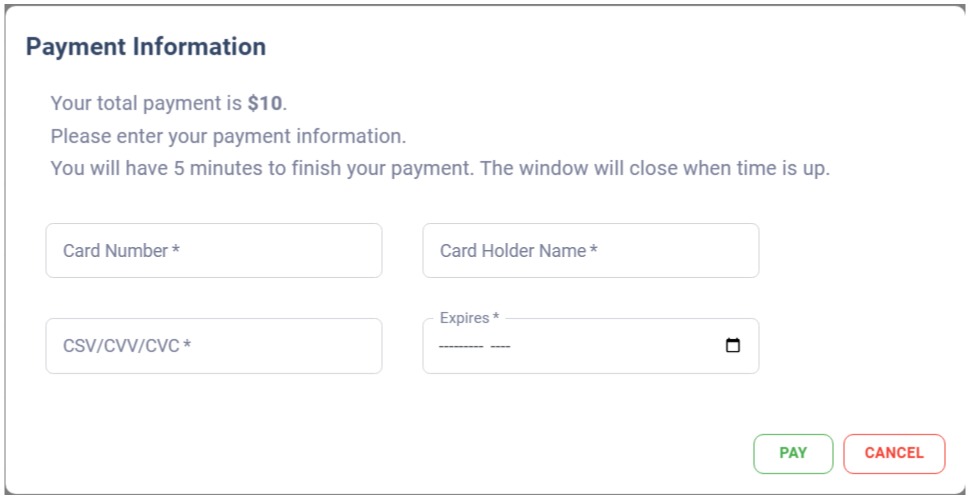
Check out a vehicle

- Click the "checkout" button of the target parking space. The system will call the function **check_parking_vehicle** to get the plate and check-in time of the vehicle parking at this space, call the function **get_owner** to get the owner information of the parked vehicle, and call the function **get_price** to get the pricing information of this parking space. A receipt will be generated for the user.
  - 

    Parking Receipt

    Please confirm the below information is correct.

    User License Number: 999999999
    Parking Lot: 73
    Pakring Space: 7
    Space Hourly Price: $1
    Vehicle Plate Number: GGGGGG
    Vehicle Type: Truck
    Check-in Time: 2022-12-03 22:02:57

- After computing the amount of payment according to parking time and space price, the system will display the amount and ask for payment information. The session will stay for 5 minutes at most and will automatically close after 5 minutes.
  - 

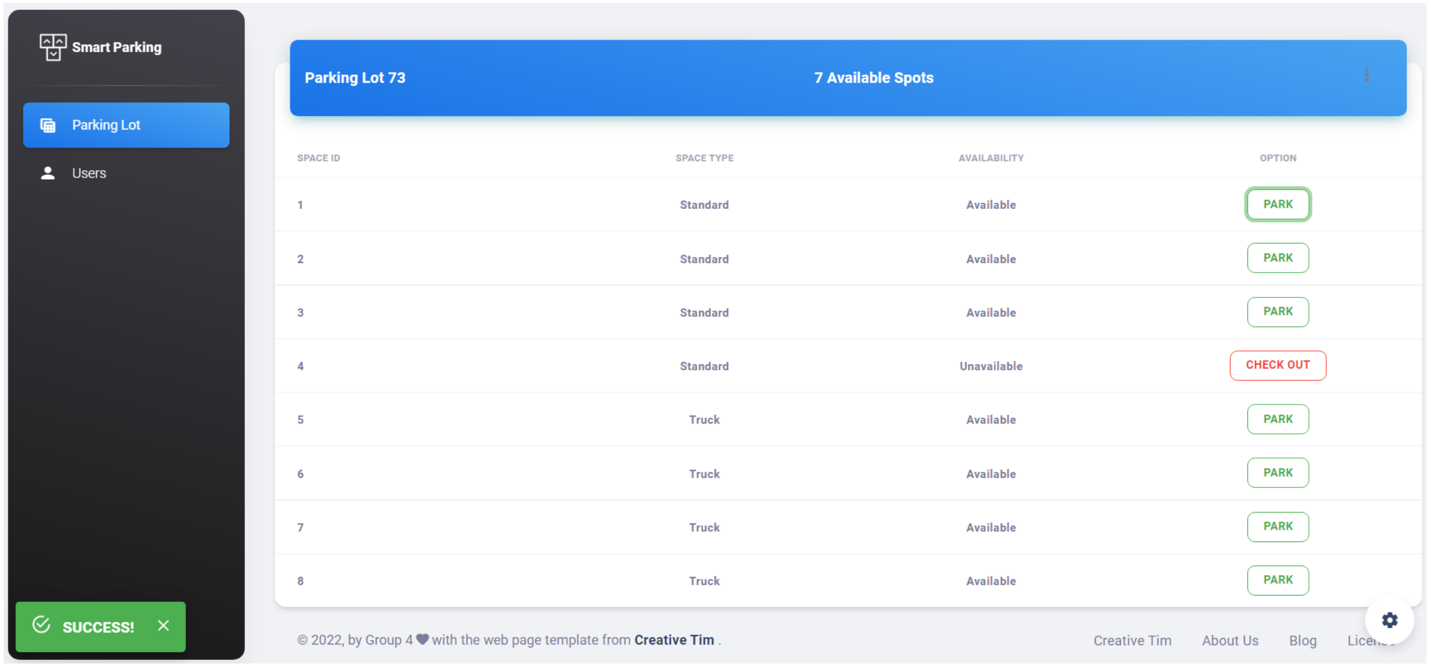
    Payment Information

    Your total payment is **$10**.
    Please enter your payment information.
    You will have 5 minutes to finish your payment. The window will close when time is up.

    Card Number *
    Card Holder Name *
    CSV/CVV/CVC *
    Expires *
    --------- ----

- After confirmation of payment, the system will call the function **check_out** to insert the payment record into the database, update the occupying status of this parking space, and delete the parking record. The system will also call the function **compute_idle** to update the occupancy status of this parking lot.

-



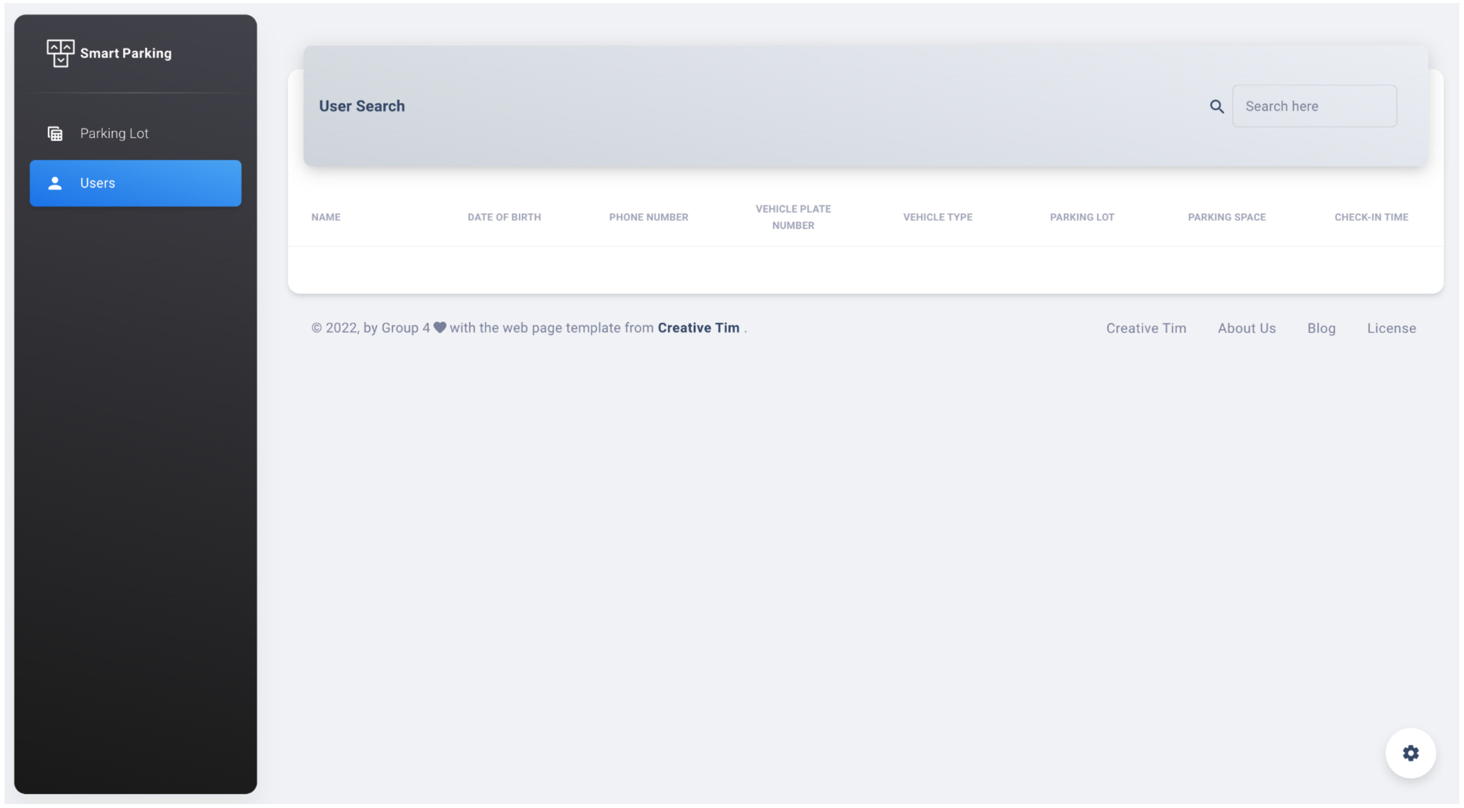
Parking Lot 73 — 7 Available Spots

SPACE ID	SPACE TYPE	AVAILABILITY	OPTION
1	Standard	Available	PARK
2	Standard	Available	PARK
3	Standard	Available	PARK
4	Standard	Unavailable	CHECK OUT
5	Truck	Available	PARK
6	Truck	Available	PARK
7	Truck	Available	PARK
8	Truck	Available	PARK

© 2022, by Group 4 ♥ with the web page template from **Creative Tim** .

Creative Tim About Us Blog Licen

User Interface

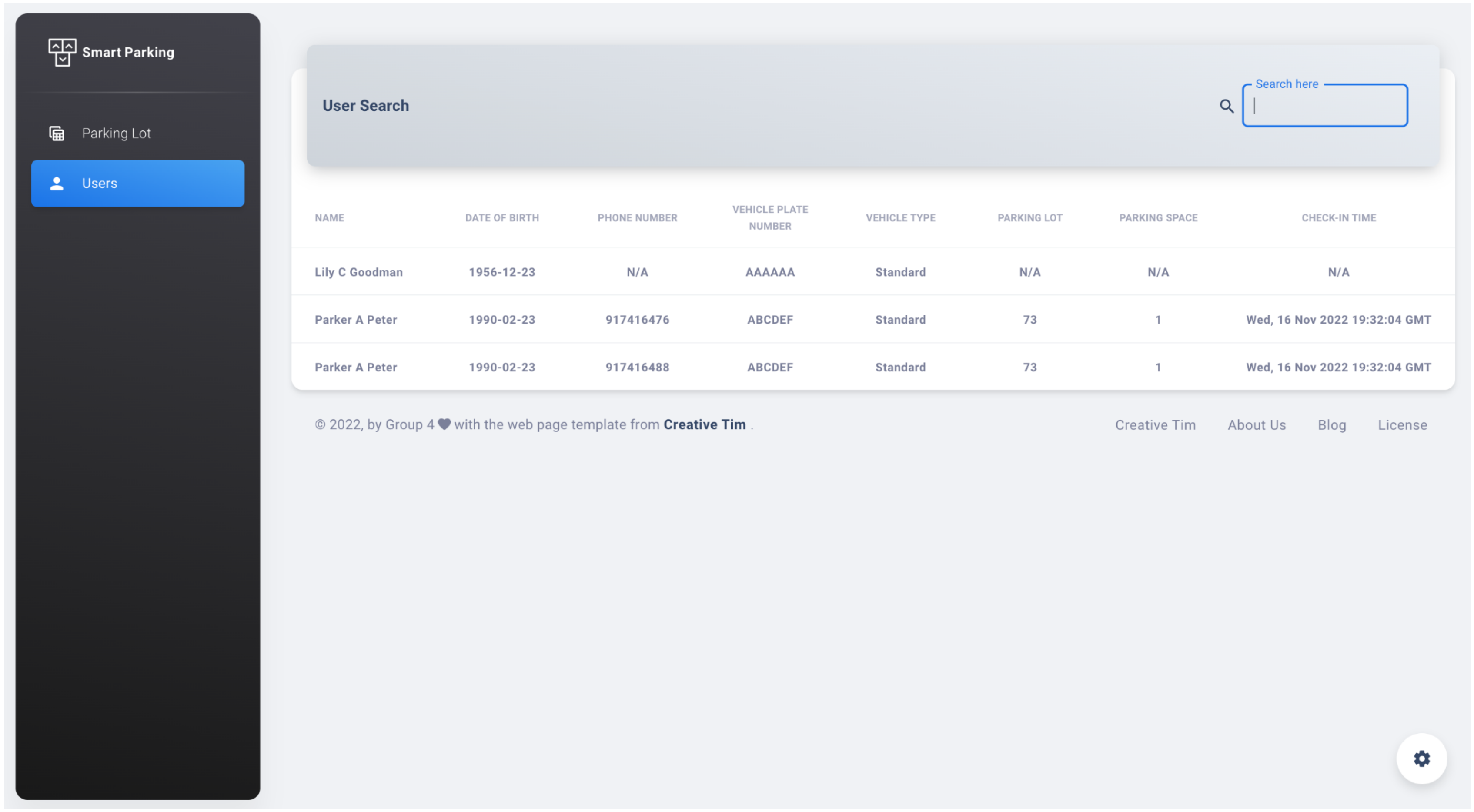


NAME	DATE OF BIRTH	PHONE NUMBER	VEHICLE PLATE NUMBER	VEHICLE TYPE	PARKING LOT	PARKING SPACE	CHECK-IN TIME

© 2022, by Group 4 ♥ with the web page template from **Creative Tim** .

Display all relevant information about a user

- Once the user types a string into the search box and presses "Enter," the program calls function **list_owner_detail** to retrieve and display all attributes relevant to all users whose driver's license number matches or is similar to the user's search input. For example, if the user enters "1," the following records will be displayed.



NAME	DATE OF BIRTH	PHONE NUMBER	VEHICLE PLATE NUMBER	VEHICLE TYPE	PARKING LOT	PARKING SPACE	CHECK-IN TIME
Lily C Goodman	1956-12-23	N/A	AAAAAA	Standard	N/A	N/A	N/A
Parker A Peter	1990-02-23	917416476	ABCDEF	Standard	73	1	Wed, 16 Nov 2022 19:32:04 GMT
Parker A Peter	1990-02-23	917416488	ABCDEF	Standard	73	1	Wed, 16 Nov 2022 19:32:04 GMT

© 2022, by Group 4 ♥ with the web page template from **Creative Tim** .   Creative Tim   About Us   Blog   License

-

Future Implementation

- The **list_payments** backend functions, queries, and servlet are fully implemented. Once the front-end is completed, we can extend the list_payment function to our graphical user interface.