

目录

MainMenu ()	1
adminMenu ()	1
spotOpMenu ()	2
customerOpMenu ()	3
cusMenu ()	4
Class design	4
Menu.h	4
Class Menu	4
Customer.h	5
Class Ticket	5
Class Customer	5
ParkingSpot.h	5
Class ParkingSpot	5
Class ParkingFloor	6
Class ParkingLot	6

MainMenu ()

Once the user's choice is converted to an int, a switch statement is used to call the appropriate function based on the user's input: `adminMenu ()` for option 1, `cusMenu ()` for option 2, and `exit (0)` for option 3.

adminMenu ()

input the correct password.

Display the menu options.

Prompt the user to select an option.

If the user selects option 1, get into the spotOpMenu ().

If the user selects option 2, get into the customerOpMenu ().

If the user selects option 3, change the password.

Option 3 allows the user to change their password. First, the user must enter their old password to confirm their identity. Then they can input a new password. Before updating the password, the user will be asked to enter the new password again to double-check if it's correct. If the two inputs do not match, the user has two more chances to enter the second password correctly. After two chances, the function returns to the main menu without changing the password.

If the user selects option 4, exit the program.

spotOpMenu ()

browseSpot ()

input "*" to check total left spots for each type of spot.

display should contain follow the format.

for all:

floor 0:

big:10 medium:30 small:20

floor 1:

big:20 medium:20 small:20

.....

for specify type:

medium:

floor 0:10 floor 1:20 ...

add ()

create a new spot object and push into the vector " parking_spots"

modify ()

input the floor number and id to modify the type and state "isOccupied".

delete ()

input the floor number and id to delete the spot from the vector " parking_spots". When the spot is occupied, the deleting will result in failure.

customerOpMenu ()

browseCustomer ()

input "*" to browse information of all customers checked in.

input the license plate number to browse the information of specific customer.

add ()

add a customer into vector "customerList"

modify ()

input the license plate number to modify the information of a specific customer.

delete ()

input the license plate number to delete the customer information from vector "customerList"

cusMenu ()

cusBrowse ()

if already checked in, show the current parking time, where is the car parked.

If not, require the user to input the type of their car and display the availability of parking spot for that type of car on each floor.

check-in ()

input license plate number, type, floor number and id to check-in spot.

check-out ()

Display the duration of parking and the amount due for parking fees and query the user if they wish to settle the payment.

Class design

Menu.h

Class Menu

Customer.h

Class Ticket

Constructor	Ticket()
<i>Member variable</i>	
Date type	Variable name
Time_t	ID
Time_t	Type
<i>Member method</i>	
Return type	Method
Double	Payment()

Class Customer

Constructor	Customer(string license)
<i>Member variable</i>	
Date type	Variable name
String	Liecence_plate_number
Ticket	Ticket

ParkingSpot.h

Class ParkingSpot

Constructor	ParkingSpot(int id, ParkingSpotType type)
<i>Member variable</i>	
Date type	Variable name
Int	ID

ParkingSpotType	Type
bool	isOccupied

Class ParkingFloor

Constructor	ParkingFloor(int floorNumber)
<i>Member variable</i>	
Date type	Variable name
Int	floorNumber
vector<ParkingSpot>	parking_spots

Class ParkingLot

Constructor	ParkingLot()
<i>Member variable</i>	
Date type	Variable name
vector<ParkingFloor>	parking_floors
<i>Member method</i>	
Return type	Method
vector<ParkingFloor>	getFloors()