Parkify

by

Ahad Abuyabis

&

Group Members

Parkify

Chapter 1
(SE101)

ABSTRACT

This report talks about the application of software engineering, the report starts by presenting the problem we are facing in reality and the proposed solutions to solve this problem, and then we will start with the system itself, which starts collecting the requirements, followed by the system requirements .We will implement an application that will become the solution to the problem that prompted us to implement this application.

Problem definition:

The difficulty of having parking at commercial malls

Solution:

There are many previous solutions, such as devices that are at the parking gates for regulation or lighting above each parking lot that lights up green when the parking is available and red when it is not available. There are other applications such as setty, easypark

Aims and objectives of the project:

-To avoid congestion at parking
-It is also useful in speeding up the process when there is an emergency need to go to the mall in a short time
-Organization of traffic at malls

SYSTEM ANALYSIS

*

2.2. Requirements specification
2.2.1 Functional Requirement

Sign up

User Requirement:

1. The user shall be able to create a new account.

System Requirement:

1.1 - The system shall be able to display the registration form user with a list of fields:

(first name, last name, email, password, phone number)

1.2 - The system shall be able to validate the data entered There are these constraints:

1.3 - Password length should be at least 8 characters

Contains uppercase and lowercase letters
numbers, and at least 1 Special Character.

1.4 - The system shall be able to check if the email or phone is already in the database and then
move to the log-in page.

2.2. Requirements specification
2.2.1 Functional Requirement

log in

User requirement :

2. The user shall be able to log in to his account

System requirements :

2.1 The system shall be able to display a login form to the user, which contains two fields :
(email or phone number, and password).

2.2 The system shall be able to check against the input database.
If there is matching user information go to the main interface. Otherwise, If they are incorrect they get an error message as seen below
(sorry incorrect input try again or sign up)

2.2. Requirements specification
2.2.1 Functional Requirement

Filtering

User requirement :

3. User Shall be able to filter his parking options.

System requirements :

3.1 The system shall providethe filter with the following categories:

a) Location:

- mall entrances/exits.
- near specific gates/cinemas.

b) Price:

- Monthly package
- Free

c) Availability:

- Available
- Reserved

3.2 The system shall dynamically update search results in real-time as filters are applied, displaying only relevant parking options.

3.3 The system shall display search results as a list with the following information for each parking space:

- Location name or interactive map marker for precise identification.
- Clear icons or color indicators for availability status (e.g., green for available, red for reserved).
- Price (if applicable), including details for monthly packages.

3.4 The system shall provide additional functionalities such as:

- "Reset Filters" button to clear all selections and start a new search.

2.2. Requirements specification
2.2.1 Functional Requirement

Reservation system

User Requirement:

4. The user shall be able to reserve a parking space

System Requirement:

4.1 - The system should provide real-time information on parking spot availability in the mall

4.2 - The system should be available 24/7 to accommodate different user schedules and needs.

4.3 - Users should be able to search for available parking spots based on their preferred date and time.

4.4 - The system should allow users to select and reserve a parking spot in advance.

4.5 - Users should have the option to modify or cancel their reservations if needed.

4.6 - the system shall be able to find a parking area from the list of areas, registered by parking admins.

2.2. Requirements specification
2.2.1 Functional Requirement

booking location

User requirement :

5. Users must be able to view the details of a selected parking area.

System requirements :

5.1-The system must show the user the available parking spaces via an image that includes (the parking name and number) and the unavailable parking spaces.

5.2- The system shall be able to allow user to search for parking space based on their current location

5.3- The system shall be able to provide a detailed GPS map and up-to-date about the location of the parking

5.4-The system must show the user the closest gate to the desired stop

5.5-The system must mark that the current position has been taken if the user reserves this position

5.6-The system shall be able to send a text message to the user with the name of his parking lot and the nearest gate to it

2.2. Requirements specification
2.2.1 Functional Requirement

Handle Special parking

User requirement:

6.The user can choose a specific park or subscribe to monthliy pakging.

System requirements:

6.1- The system shows the user the paid specific parking lots near his location or near the gate.

6.2- The system shows the user the parking lots that he can pay it, either a specific place he chooses, or a monthly packages .

6.3- The system shows the user the fees to be paid depend on the user choice.

6.4- The system shows parking information to the user.

6.5- The admin handles the subscription.

2.2. Requirements specification
2.2.1 Functional Requirement

Payment

User requirements:

7-The users shall be able to pay by card..

System requirements:

7.1 The system shall be able to enter card information.
7.2 The system shall be able to Add the account number, expiration date, and card code.
7.3 THE system shall be able to Send a verification code to the user if the INFORMATION ON the card is correct.
7.4 THE system shall be able to ACCEPT The process if the code is VALID.
7.5 THE system shall be able to reject The process ifthe code is INVALID.
7.6 The system shall be able to withdraw money from the card.
7.7 The system shall be able to notify the user that the withdrawal was successful.
7.8 The system shall be able to reject the transaction if there is no MONEY.
7.9 The system shall be able to save the card for future operations.
7.10 The system shall be able to delete the card if the user asks for it.
7.11 The system shall be able to add more than one card.

2.2. Requirements specification
2.2.1 Functional Requirement

Notification

User requirement:

8. The user shall be able to allow notifications.

System requirement:

8.1- The System shall be able to give notifications about the reservation coordinates.

2.2. Requirements specification
2.2.1 Functional Requirement

Customer services

User requirement:

9.The user shall be able to ask a question.

System requirement :

9.1 -The system shall be able to take input from user such as: (name, email, subject, message)

9.2-The system shall be able to provide communication between user and customer service.

9.3-The system shall be able to contact the user after 30 minutes.

Non-functional Requirements

Usability

- Visual design: Consider a clear and concise layout with easily identifiable icons and buttons. Use colors and fonts that are accessible for various vision abilities
- Offline functionality: Allow users to save parking details or reservations for later, even if they lose internet connection
- Driving directions: Integrate with map services to provide clear and reliable navigation to the reserved parking area
- Accessibility: Ensure the app is compatible with screen readers and other assistive technologies for users with disabilities.

Performance

- Real-time updates: Display parking availability and updates in real-time to avoid frustration and wasted time
- Fast loading times: Optimize the app to load quickly, especially during peak usage periods
- Battery efficiency: Implement features that minimize battery drain for extended app use

Security

- Secure login: Use strong password encryption and authentication methods to protect user accounts.
- Payment security: Integrate with secure payment gateways to protect credit card information during transactions
- Data privacy: Clearly explain your data collection and usage policies to comply with relevant regulations.

Reliability

- Offline map access: Allow users to download map data for offline use in case of network outages
- Automatic updates: Regularly update the app to fix bugs and improve performance.

Maintainability

- Logging and error tracking: Implement logging and error tracking mechanisms for easier troubleshooting
- User feedback: Provide feedback channels for users to report issues and suggest improvements.

Chapter 3: System design

*

3.1 Tools and Methodology

Canava

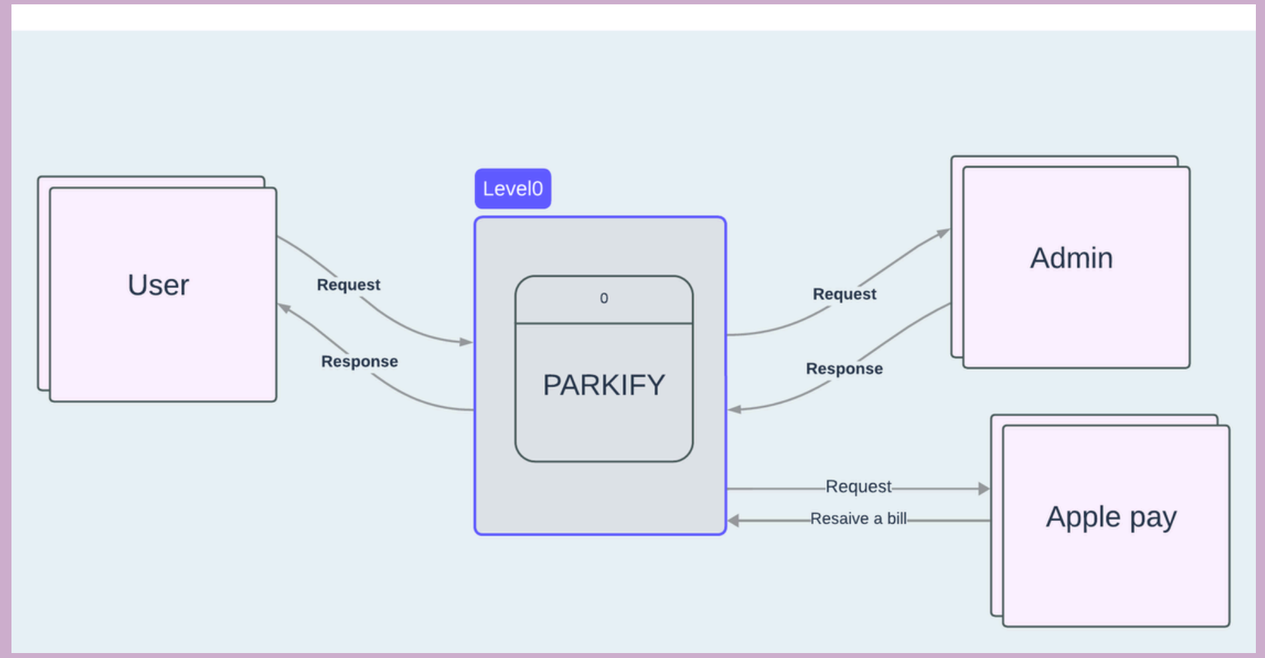
Google Meet

Lucid App

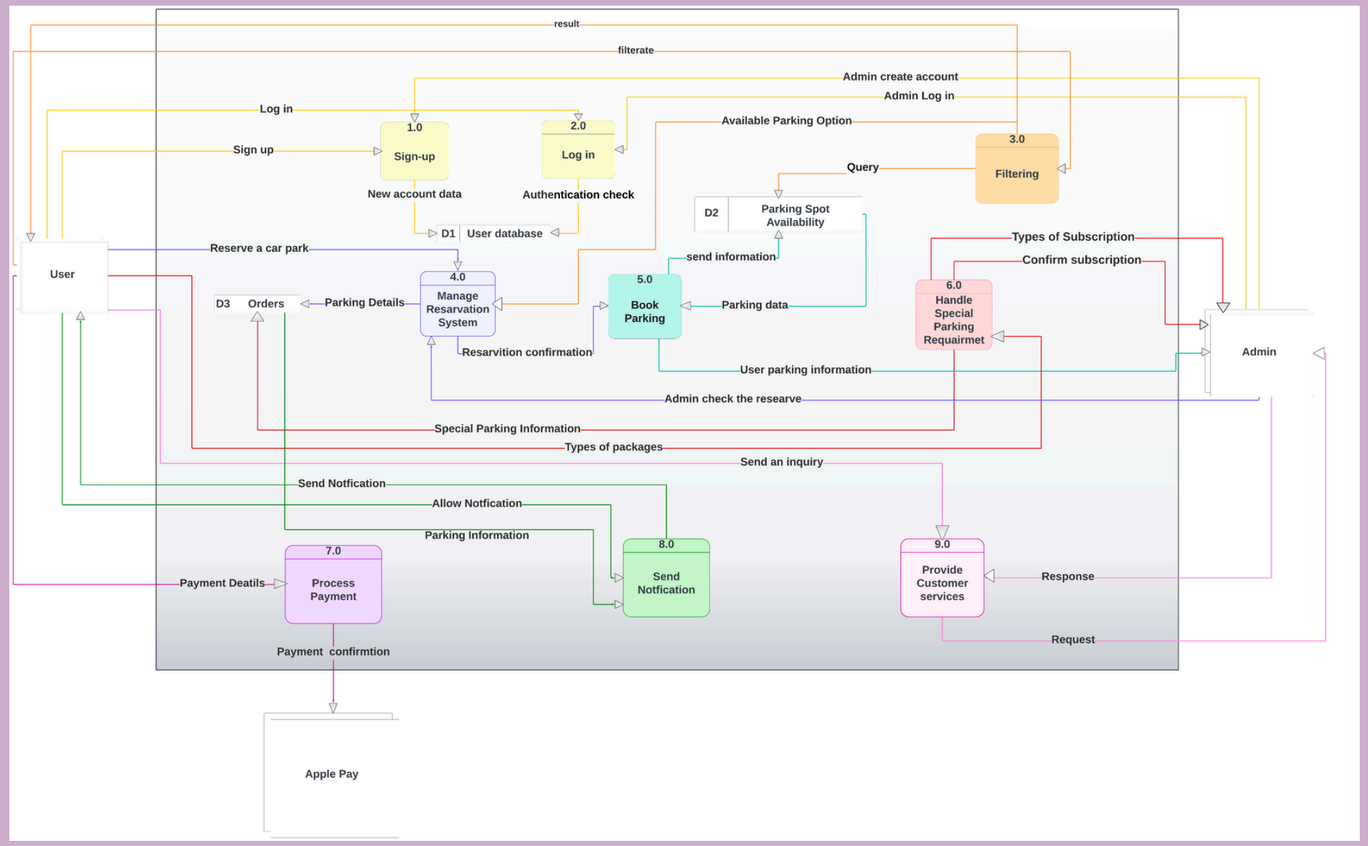
Draw io

visio
microsoft

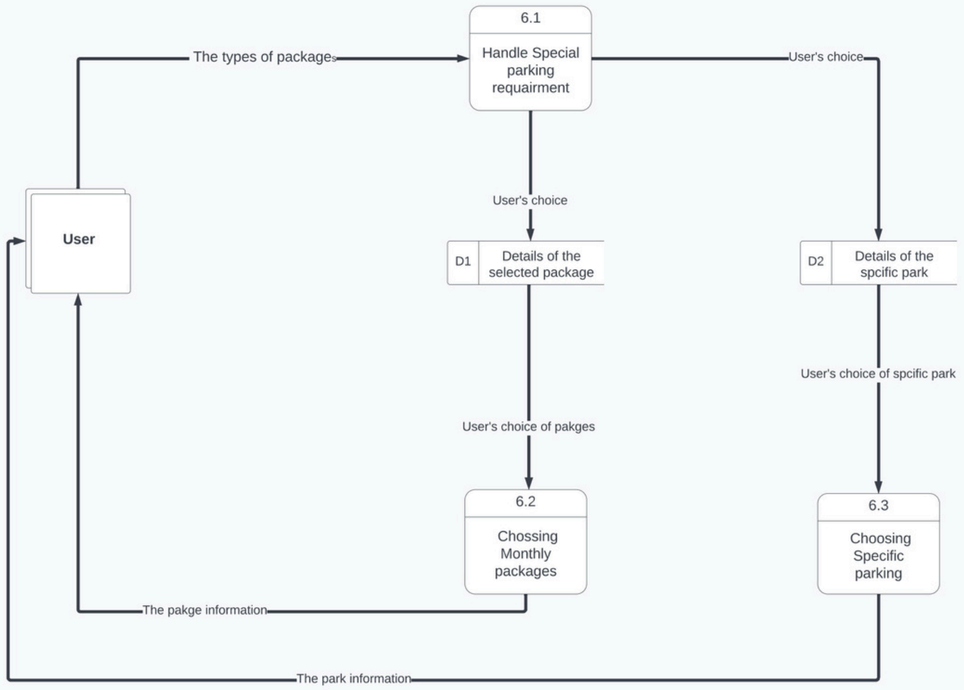
Level 0


Level0
User
Request
Response
0
PARKIFY
Request
Response
Admin
Request
Resaive a bill
Apple pay

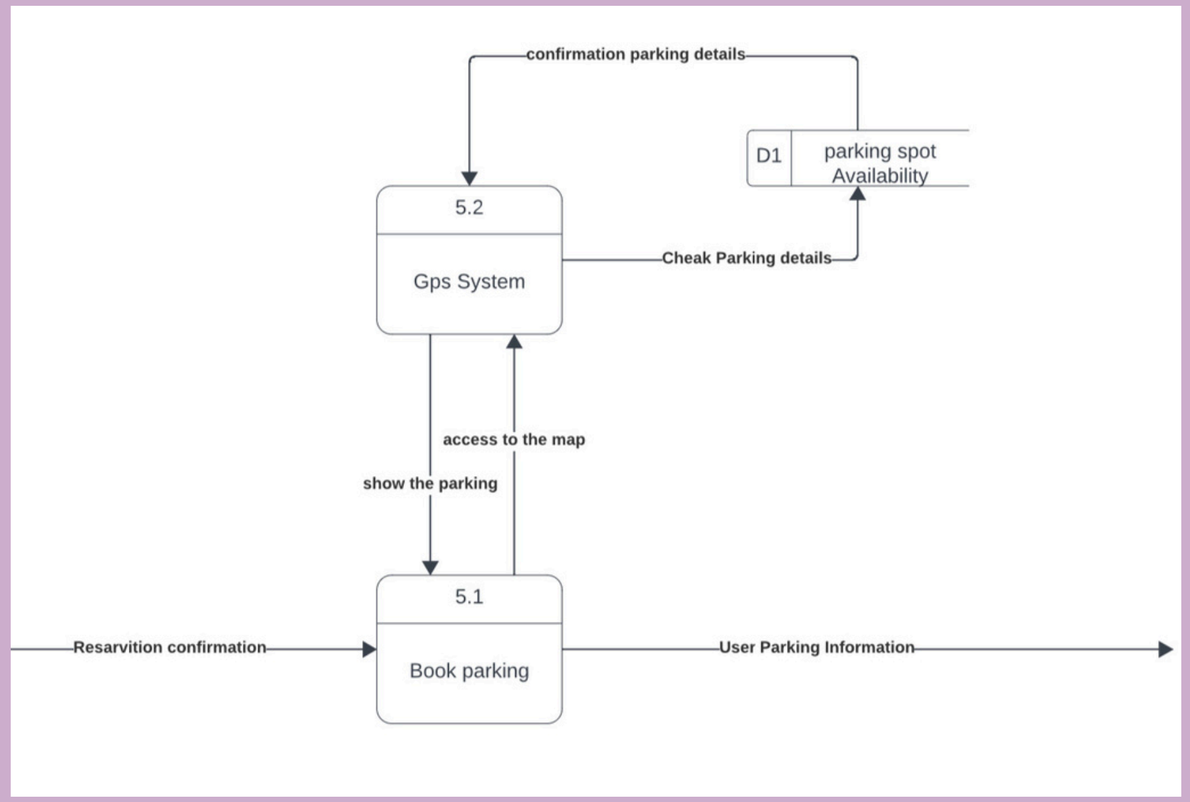
Level 1


result
filterate
Admin create account
Admin Log in
Log in
Available Parking Option
1.0
Sign-up
2.0
Log in
3.0
Filtering
Sign up
Query
New account data
Authentication check
D2
Parking Spot Availability
D1
User database
Types of Subscription
Confirm subscription
Reserve a car park
User
send information
4.0
Manage Resarvation System
5.0
Book Parking
6.0
Handle Special Parking Requairmet
D3
Orders
Parking Details
Parking data
Admin
Resarvition confirmation
User parking information
Admin check the researve
Special Parking Information
Types of packages
Send an inquiry
Send Notfication
Allow Notfication
Parking Information
7.0
Process Payment
8.0
Send Notfication
9.0
Provide Customer services
Response
Payment Deatils
Request
Payment confirmtion
Apple Pay

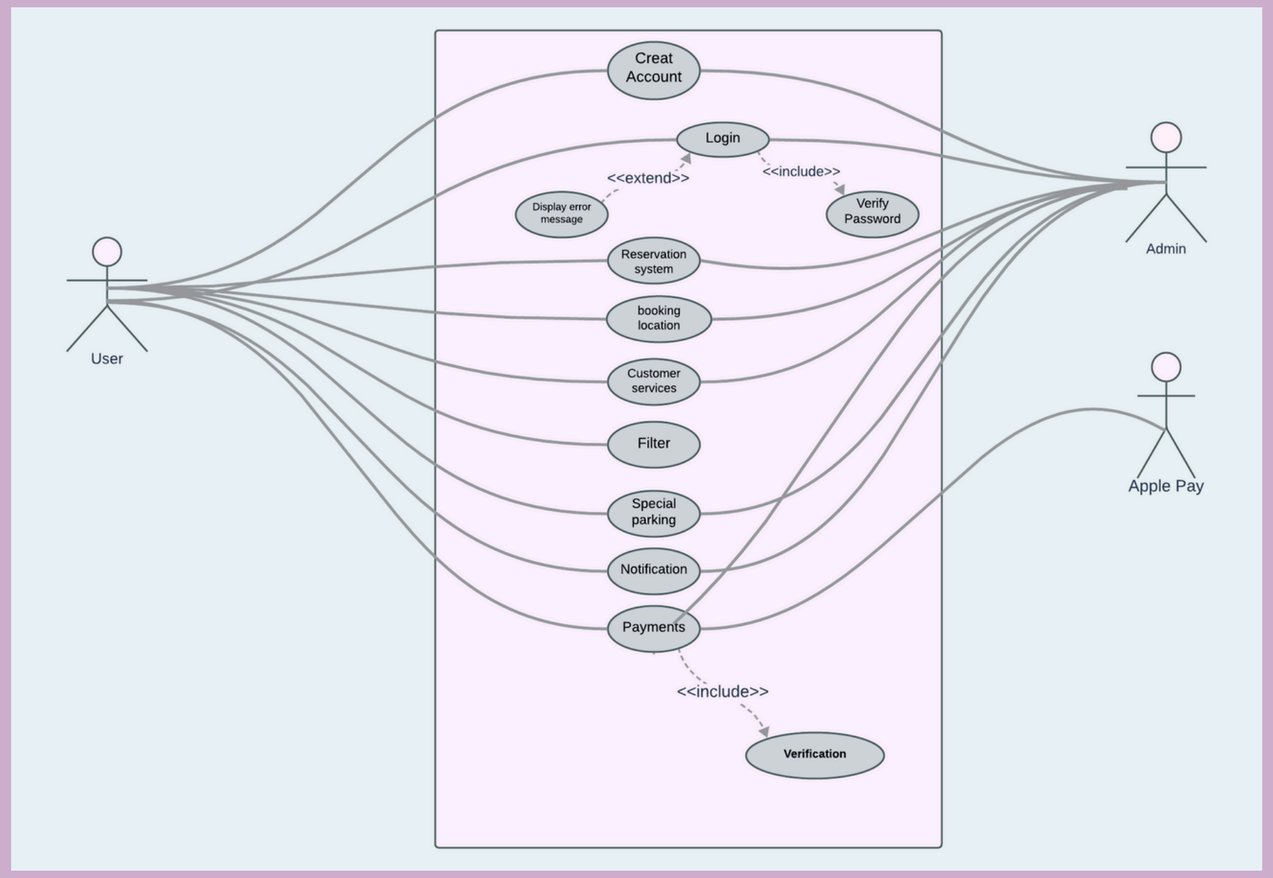
Child Diagram



Child Diagram



Use Case


Creat Account
Login
<<extend>>
<<include>>
Display error message
Verify Password
Reservation system
booking location
Customer services
Filter
Special parking
Notification
Payments
<<include>>
Verification
User
Admin
Apple Pay

Use Case Scenarios

1- Sign up

Actors	Users / admin
Description	1. The system will display the sign-up form to the user, which contains a list of fields (first name, last name, email, confirmation email, password, confirmation password, phone number and date of birth) then validate the entered data with these constraints: 2. Theemailandphonenumbermustbeunique. 3. There is no other record in the database that has the same values. 4. The password should be at least 8 characters with letters in uppercase, lowercase, and number and at least 1 Special Character. 5. If the user leaves an empty field an error message will appear. 6. The system will check if the email or phone is already in the database and then move to the log-in page.
DATA	Form fields data: (first name, last name, email, confirmation email, password, confirmation password, phone number)
Stimulus	Has an email or phone number and the user clicks on the sign-up button.
Response	Move to the next step the log-in page.
Comments	Only the admin has the accessibility to the database.
Pre-Conditions	The user should have a real email or a real phone number.

Use Case Scenarios

2- Log in

Actors	Users / admin
Description	1. The system will display a login form to the user, which contains two fields: (email or phone number, and password). 2. The email and password must match an existing one by signing up. 3. If the user leaves an empty field an error message will appear. 4. If the email and password are valid the user will go to the home page.
DATA	• User information (email or phone number, password) the system will check the entered data with the database. if there is a match the user will be navigated to the home page. otherwise, display the message contains (there is no such account, try again or sign up).
Stimulus	• Have account and users click on the login button.
Response	• The user will be navigated to the home page.
Comments	• If the user forgot the password then move to the recovery screen that has one field (email or phone number) by sending a verification code.
Pre-Conditions	• The user has to sign up before logging in.

Use Case Scenarios

3. Filtering

Actors	Users.
Description	1. Users can refine search results based on: a. Location: Specific areas, levels, proximity to landmarks. b. Price: special parking ,specific parking,free c. Availability: Real-time updates on occupied/vacant spaces. 2. Streamlines the search process by eliminating irrelevant options. 3. Promotes informed decision-making by displaying key parking details 4. user can reset his filter options
DATA	• Available parking spaces • Parking location details (proximity to mall entrances/exits, gates/cinemas) • Parking prices (monthly package, free) Parking availability status (available, reserved)
Stimulus	• User clicks on the "Filter" button • User selects desired filters within each category
Response	• The system displays the filter interface with categories for Location, Price, and Availability • The system dynamically updates the search results to show only parking options that match the selected criteria
Comments	• Multiple filters can be applied within a category • The system should handle technical issues gracefully, displaying appropriate error messages if it cannot retrieve or filter parking options.
Pre-Conditions	• User has accessed the parking search feature • The system has retrieved a list of available parking options

Use Case Scenarios

4- Reservation System

Actors	Users / admin
Description	1- The system should provide real-time information on parking spot availability in the mall 2- The system should be able to easily access from user to the reservation system 3- The system should be available 24/7 to accommodate different user schedules and needs. 4- The system should be able to give the user search for available parking spots based on their preferred date and time. 5- The system should allow users to select and reserve a parking spot in advance. 6- The system shall be able to find a parking area from the list of areas, registered by parking admins.
DATA	User data name, number, email time for booking, booking location , available parking spots
Stimulus	The user click reserve a parking spot for his car
Response	The user done about reserve a parkin spot
Comments	only the Admins has the accessibility to the database
Pre-Conditions	The user must have an account

Use Case Scenarios

5-booking locations

Actors	Users.
Description	**1-The system must show the user the available parking spaces via an image that includes (the site letter and number) and the unavailable parking spaces.** Reserved sites will be marked with a misleading sign indicating that they are not available. **2-The system must show the user the closest gate to the desired stop** By clarifying the name of the gate and the nearest important centers to it, such as (shopping stores, cinemas, pharmacies) to achieve customer satisfaction in choosing the nearest gate to perform his purpose. **3-The system must mark that the current position has been taken if the user reserves this position.** **4-The system shall be able to send a text message to the user with the name of his parking lot and the nearest gate to it** A text message will be sent via notifications to the user.
DATA	User data (name, number, email, time for booking, booking location(site letter and number)
Stimulus	User chose their booking area.
Response	Your reservation request has been registered !
Comments	only the Admins has the accessibility to the database
Pre-Conditions	The user must have an account, pre-reservation.

Use Case Scenarios

6- Special parking

Actors	User, admin
Description	1. The user has the authority to choose special parking, and choose between monthly packages or specific parking. 2. The admin has the authority to show the user detailed information about the distinguished parking lots near the user or the user monthly subscription.
DATA	• Database(user's name, user's number , The special parking information)
Stimulus	• The user choose to subscribe to a monthly package, or choose a specific park.
Response	• The parking lot booked for the user.
Comments	• The user cannot complete the subscription until payment is confirmed. • The user cannot log in unless he enters the login information or creates an account.
Pre-Conditions	• The user must have an account. • The user must have paid.

Use Case Scenarios

7-Payment

Actors	User
Description	1- The system shall be able to enter card information. 2- The system shall be able to Add the account number, expiration date, and card code. 3- The system shall be able to reject if the account number is uncorrected. 4- The system shall be able to reject if the expiration date is uncorrected. 5- The system shall be able to reject if the card code is uncorrected. 6- The system shall be able to send a verification code to the user if the information on the card is correct. 7- The system shall be able to accept the process if the code is valid. 8- The system shall be able to reject the process if the code is invalid. 9- The system shall be able to resend the code if itis incorrect. 10- The system shall be able to withdraw money from the card if there is money. 11- The system shall be able to cancel the process if there is no money. 12- The system shall be able to re-operation the process if there is no money. 13- The system shall be able to notify the user that the withdrawal was successful. 14- The system shall be able to save the card for future operations. 15- The system shall be able to delete the card if the user asks for it. 16- The system shall be able to add more than one card.
DATA	• Bank account(User'sname,the account number, expiration date, and card code).
Stimulus	• User Pay for the special Parking
Response	• Payment WAS done if the PROCESS was successful.
Comments	• only the Admins has the accessibility to the database
Pre-Conditions	• The user must have an account.

Use Case Scenarios

8-Notification

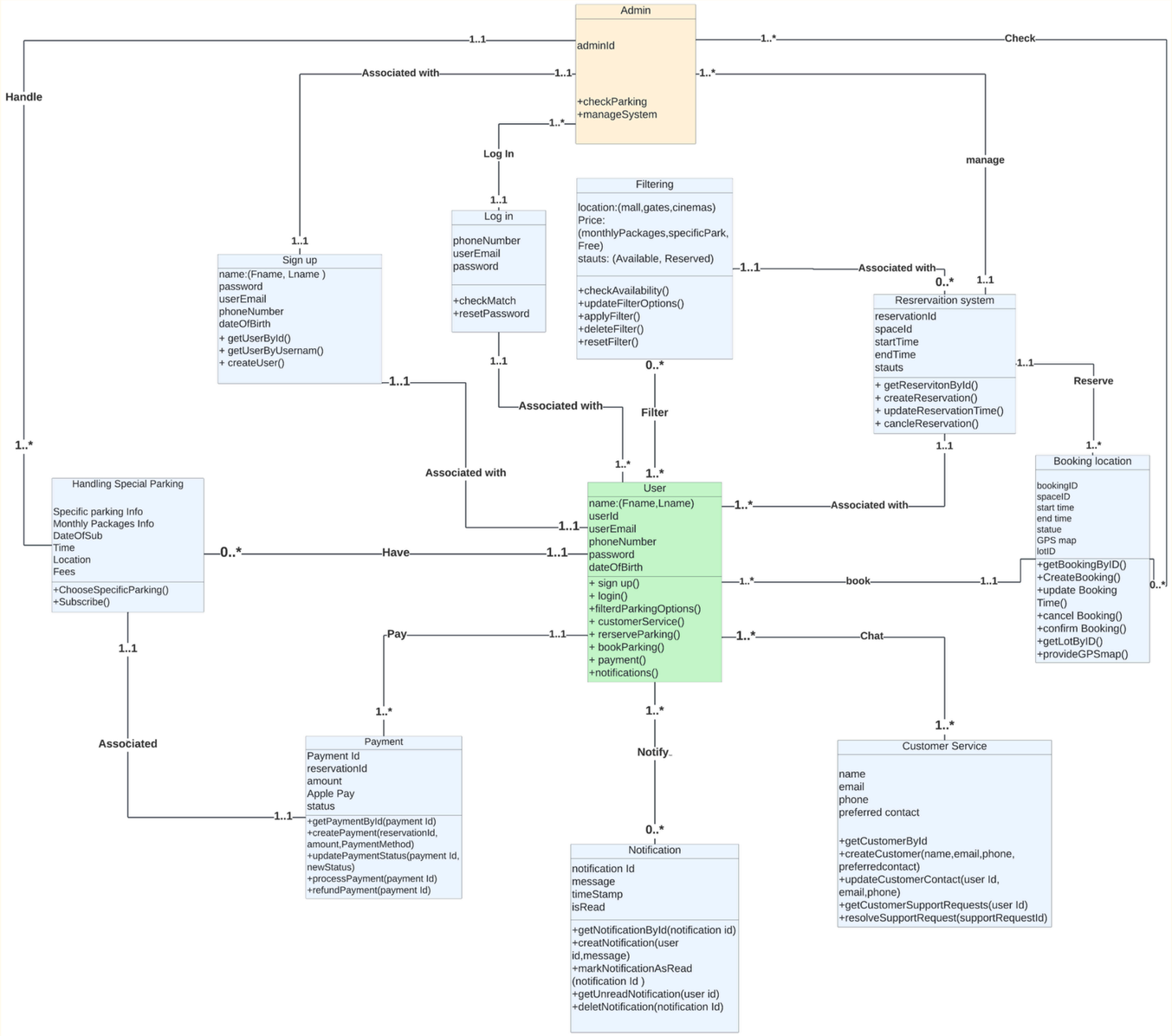
Actors	User
Description	1. The User shall be able to give the Notifications based on the parking facilities. 2. The System won't be The User will have to enable Notifications. 3. the Notifications will start one hour before the begging of the parking. 4. The System will also notify The User of the updates that are happening in parking place.
DATA	• Notification to start the counting of the parking spot
Stimulus	• Enable Notification. .
Response	• Show the Notification before the parking spot start counting in time
Comments	• if the user wants to turn off/on as the user like.
Pre-Conditions	• Enable Notification. • User must have an account.

Use Case Scenarios

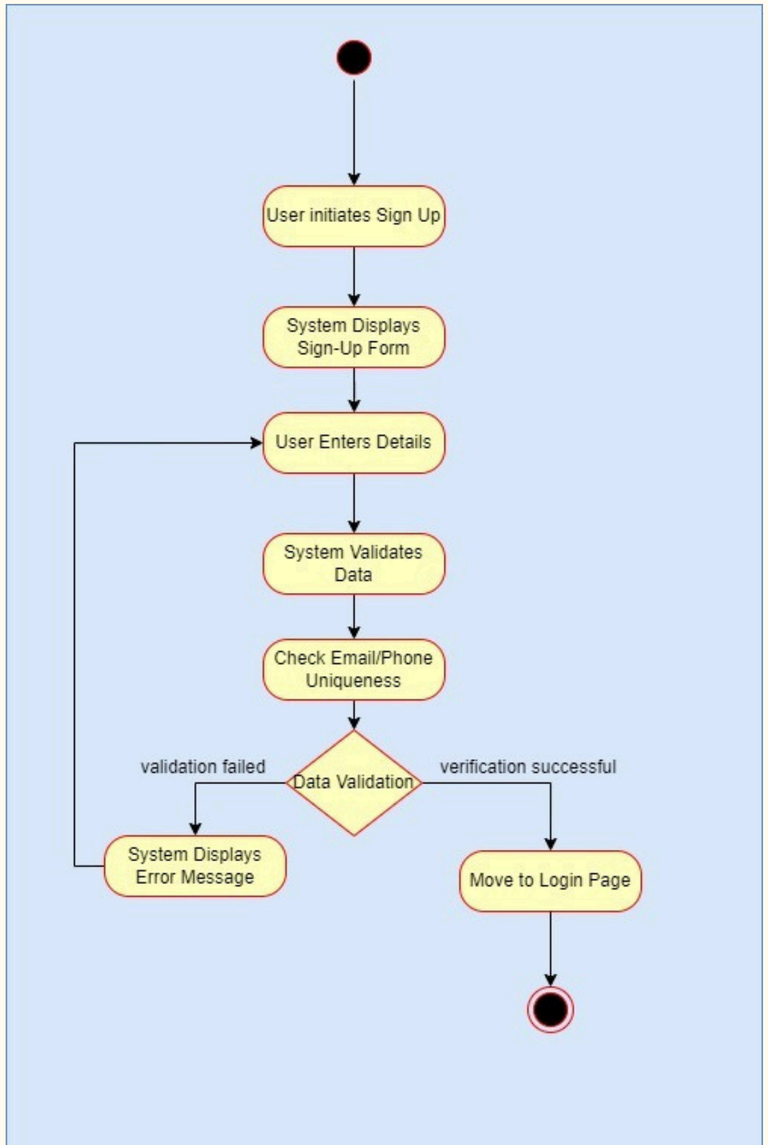
9- customer services

Actors	Users.
Description	1. When the user click on the Customer Service page the system will display a form that has the following fields: (name, email, subject, message) 2. The user should fill the form with an existing email. 3. All fields must be filled or an error message will appear. 4. After filling in the fields successfully a message will appear letting the user know that the message has been sent successfully and that customer service should reply to them after 30 minutes.
DATA	The Name and email and phone number and subject and massage.
Stimulus	When the user finishes writing the massage click submit.
Response	The message was sent, Thanks Thanks for contacting me.
Comments	None.
Pre-Conditions	None.

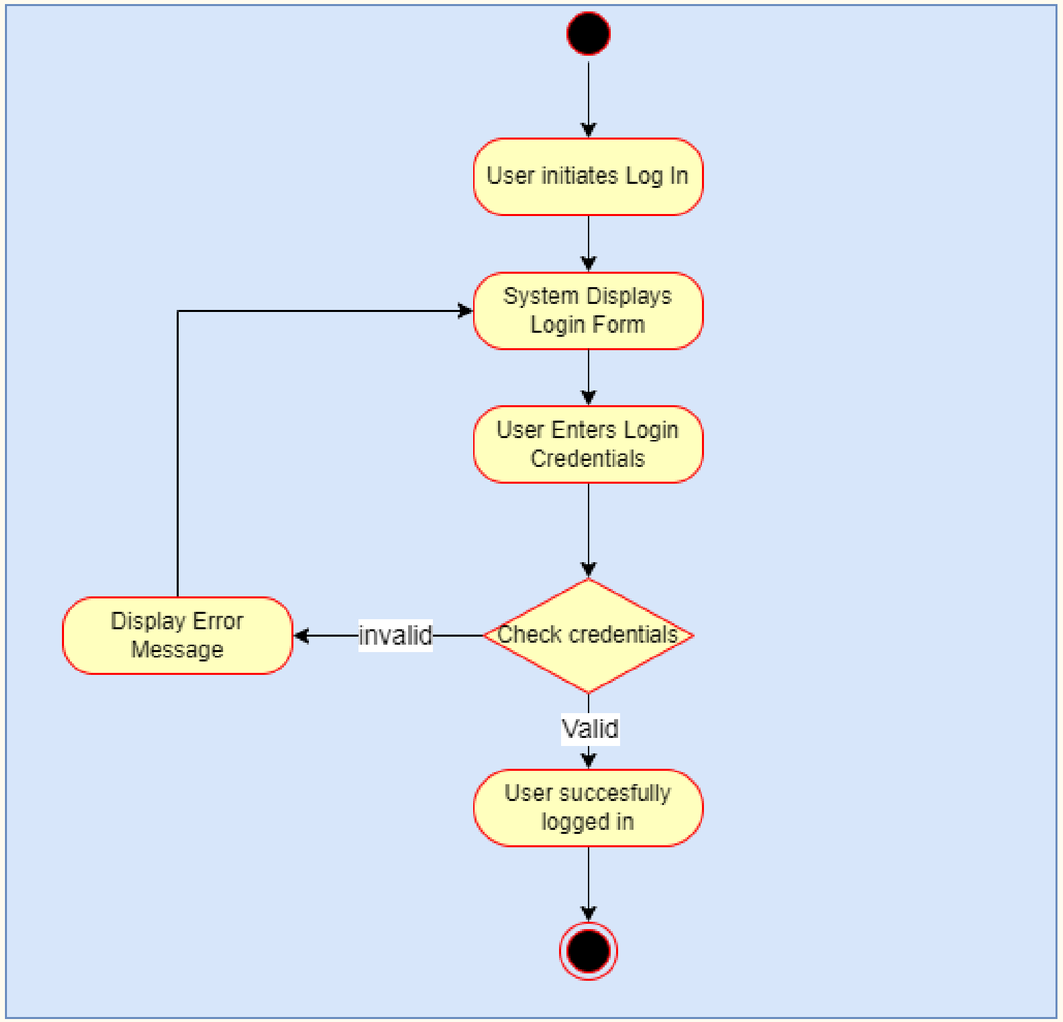
Class Diagram



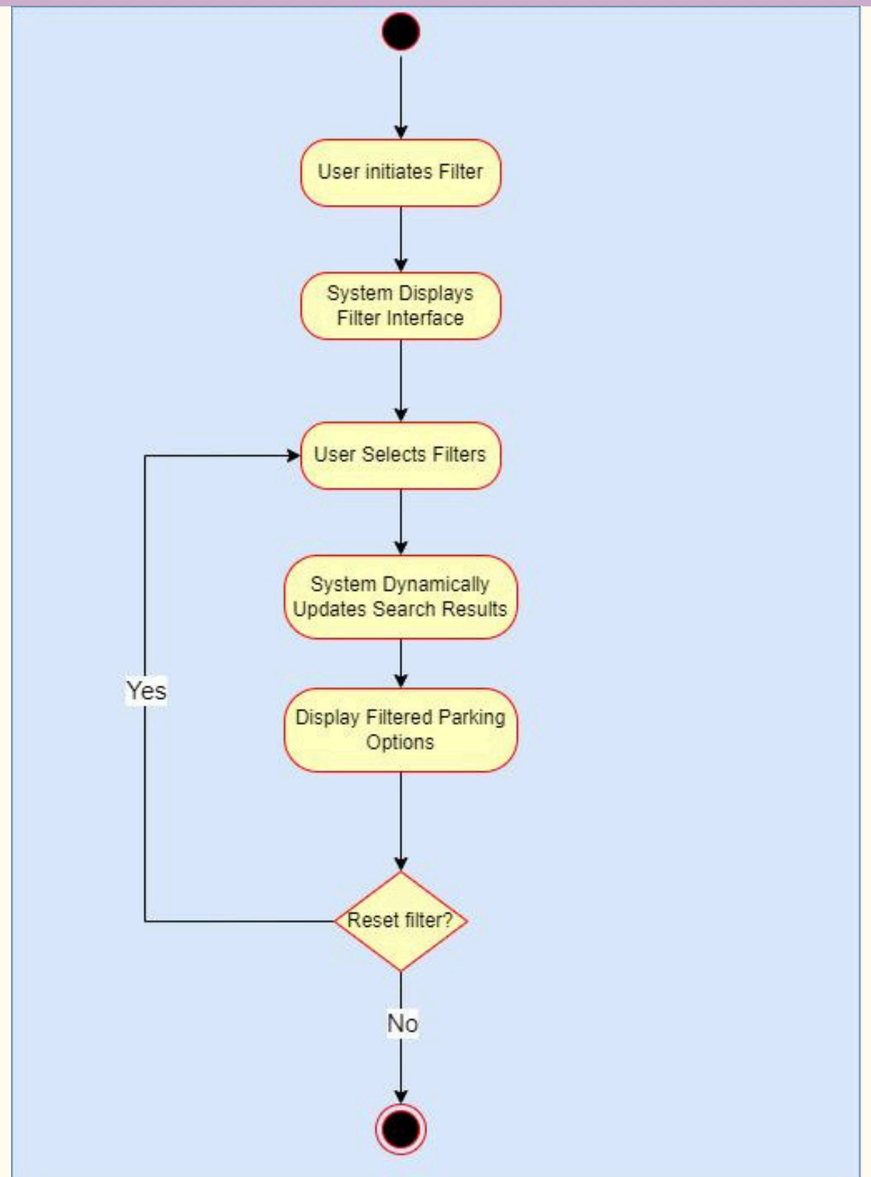
Activity diagram
Sign up



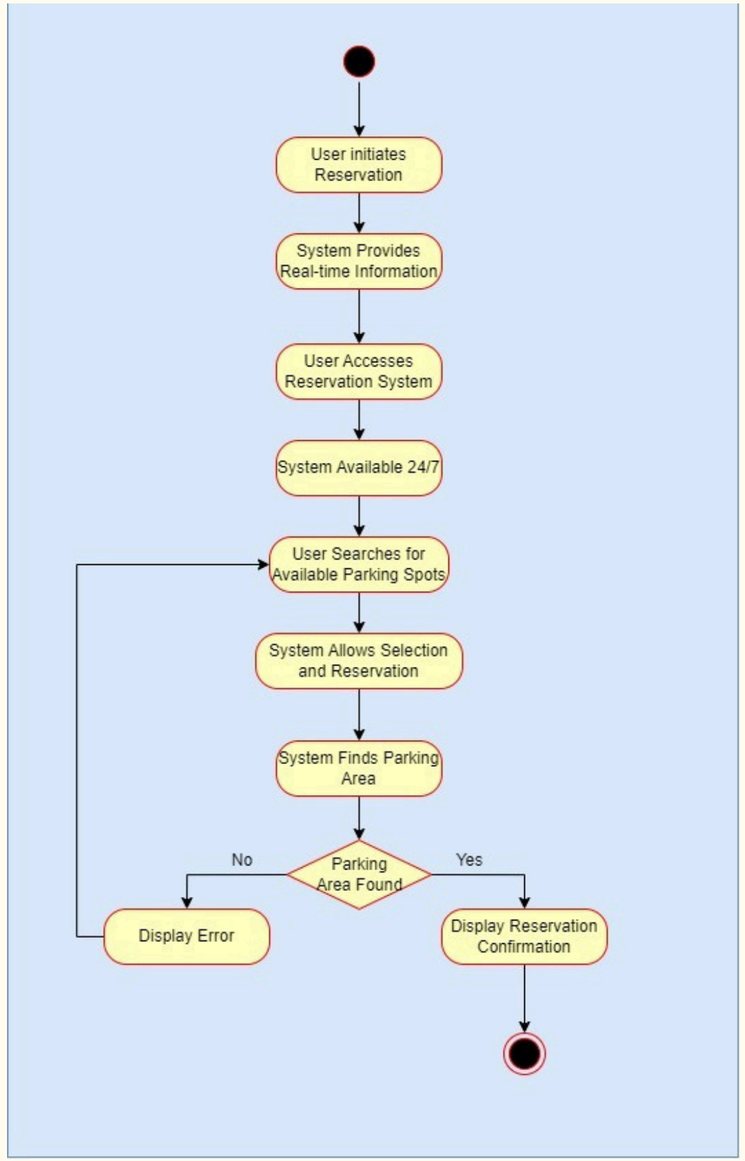
Activity diagram
Log In



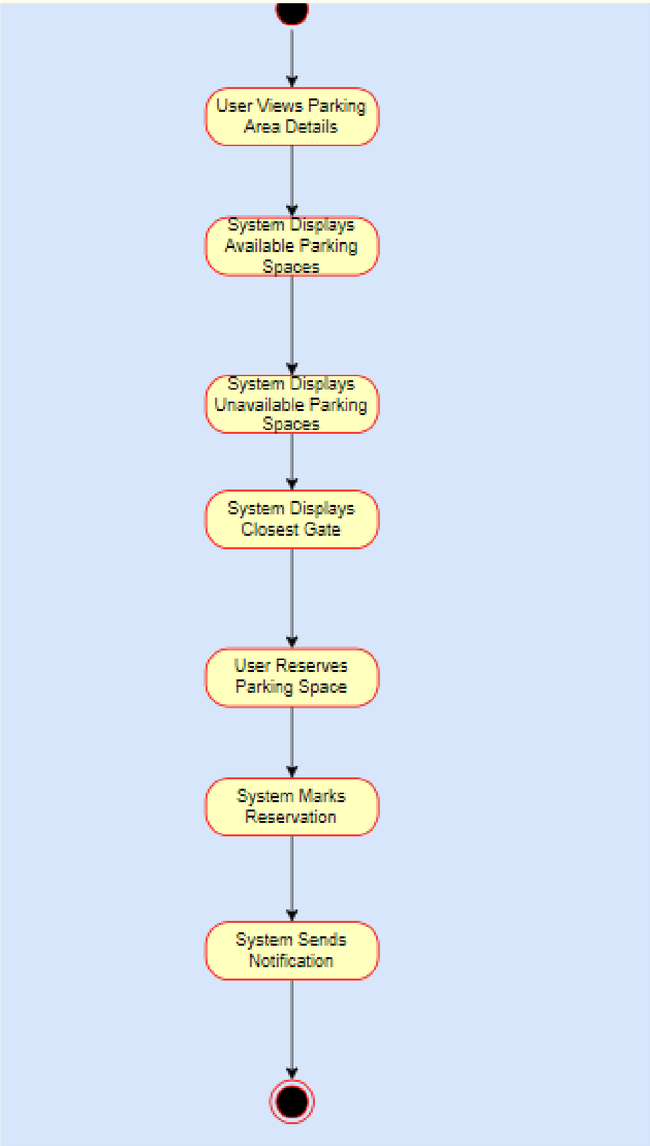
Activity diagram Filtering



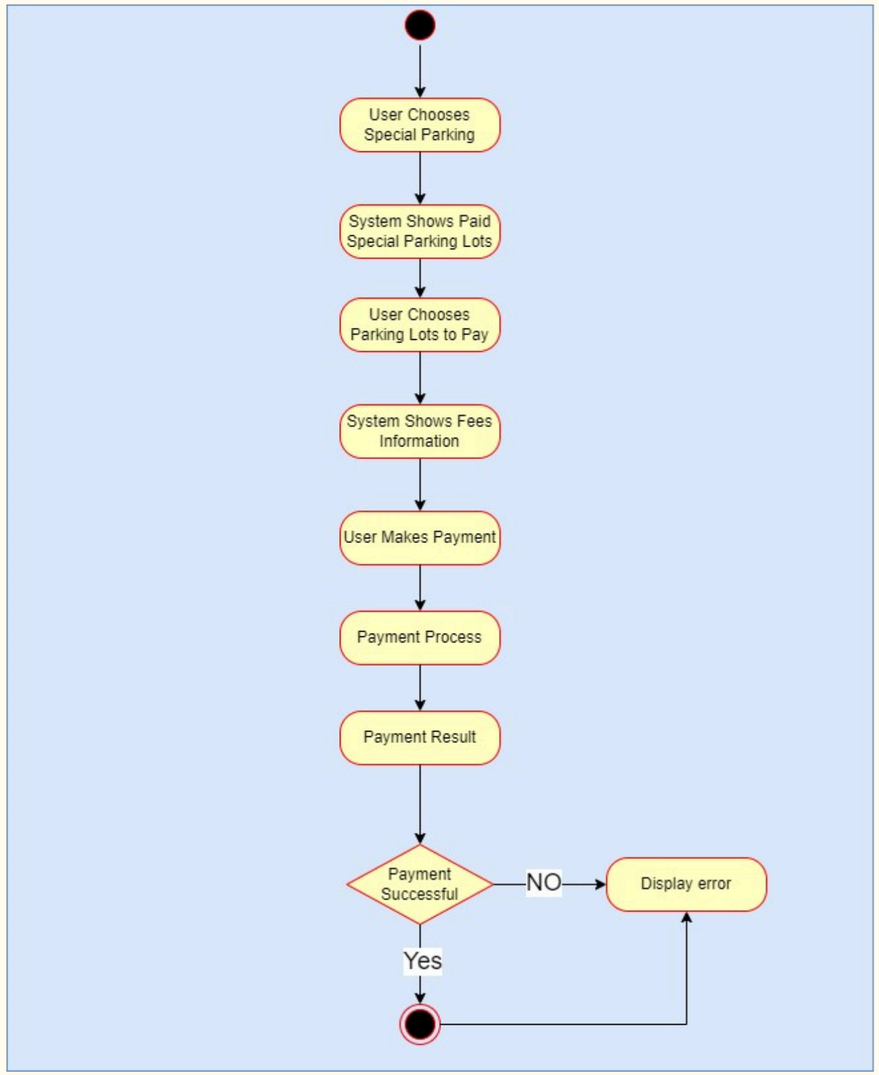
Activity diagram
Reservation system



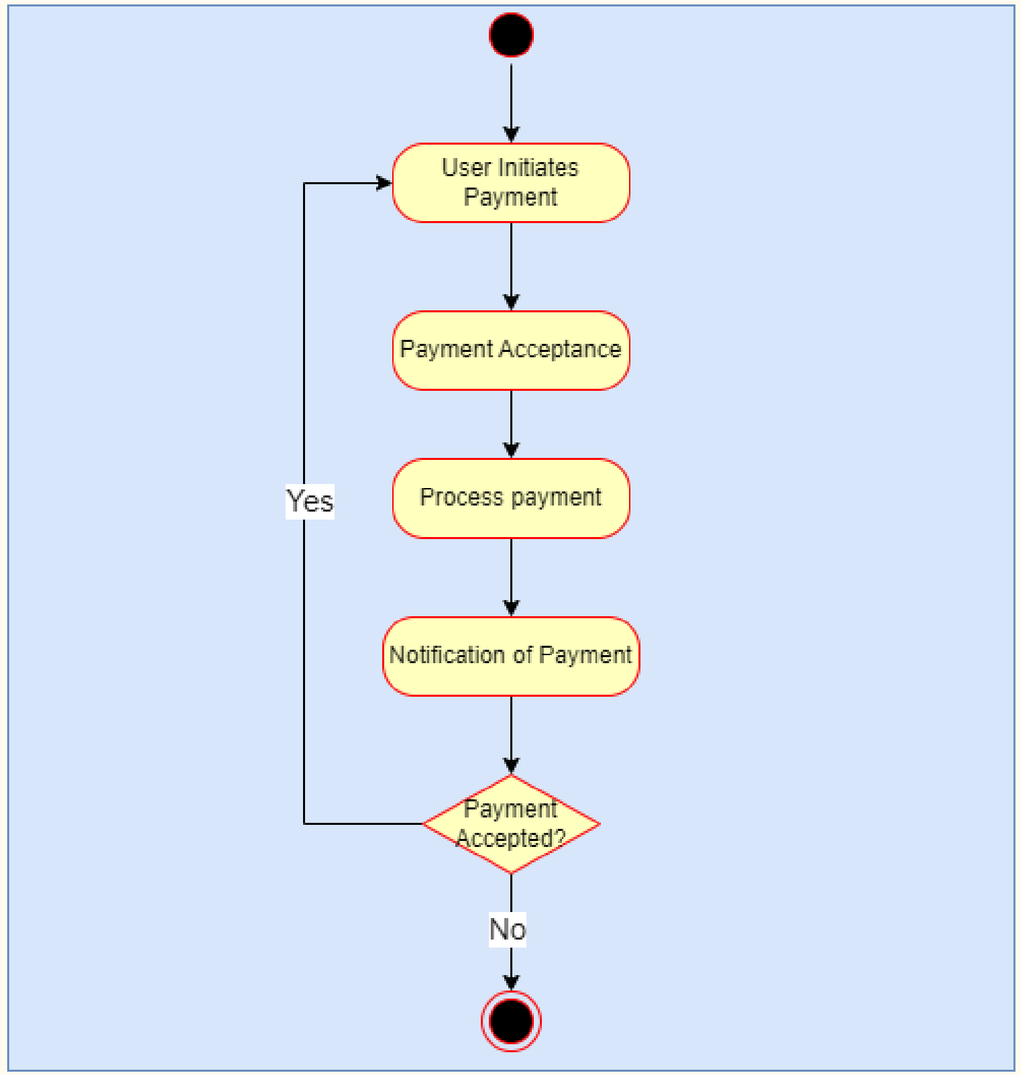
Activity diagram booking location



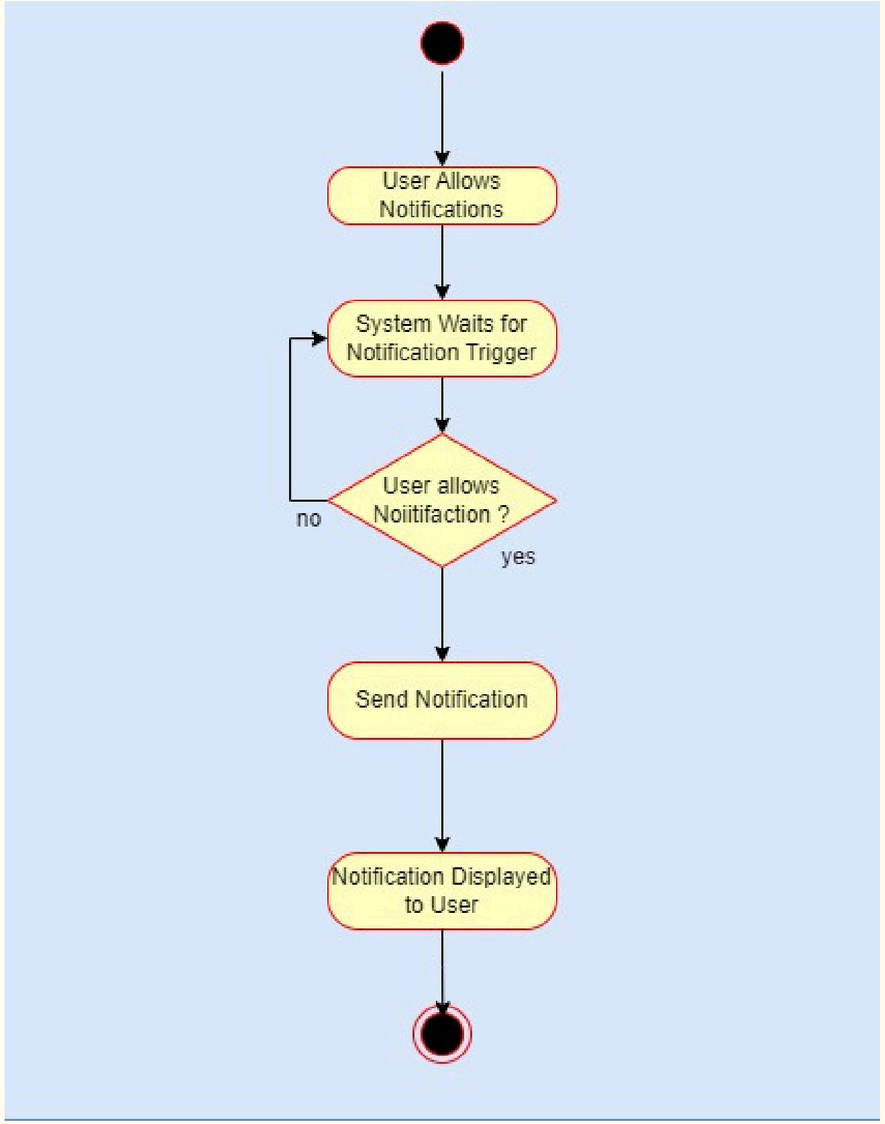
Activity diagram
Special Parking



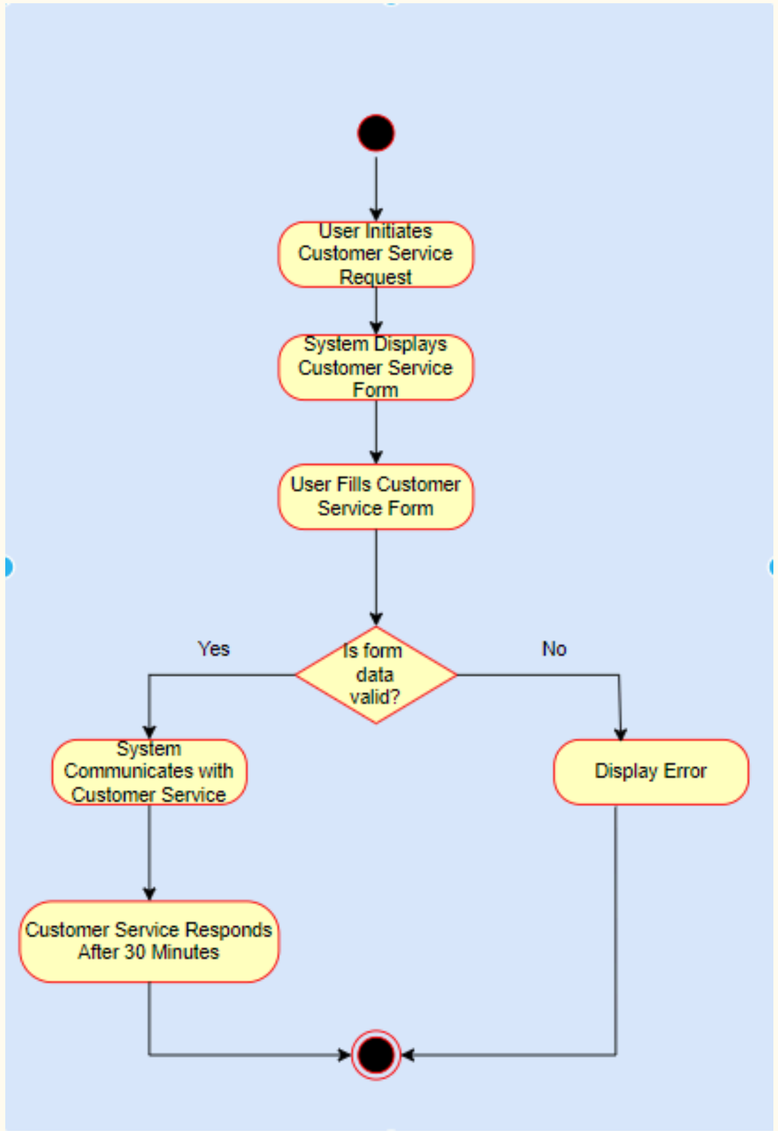
Activity diagram Payment



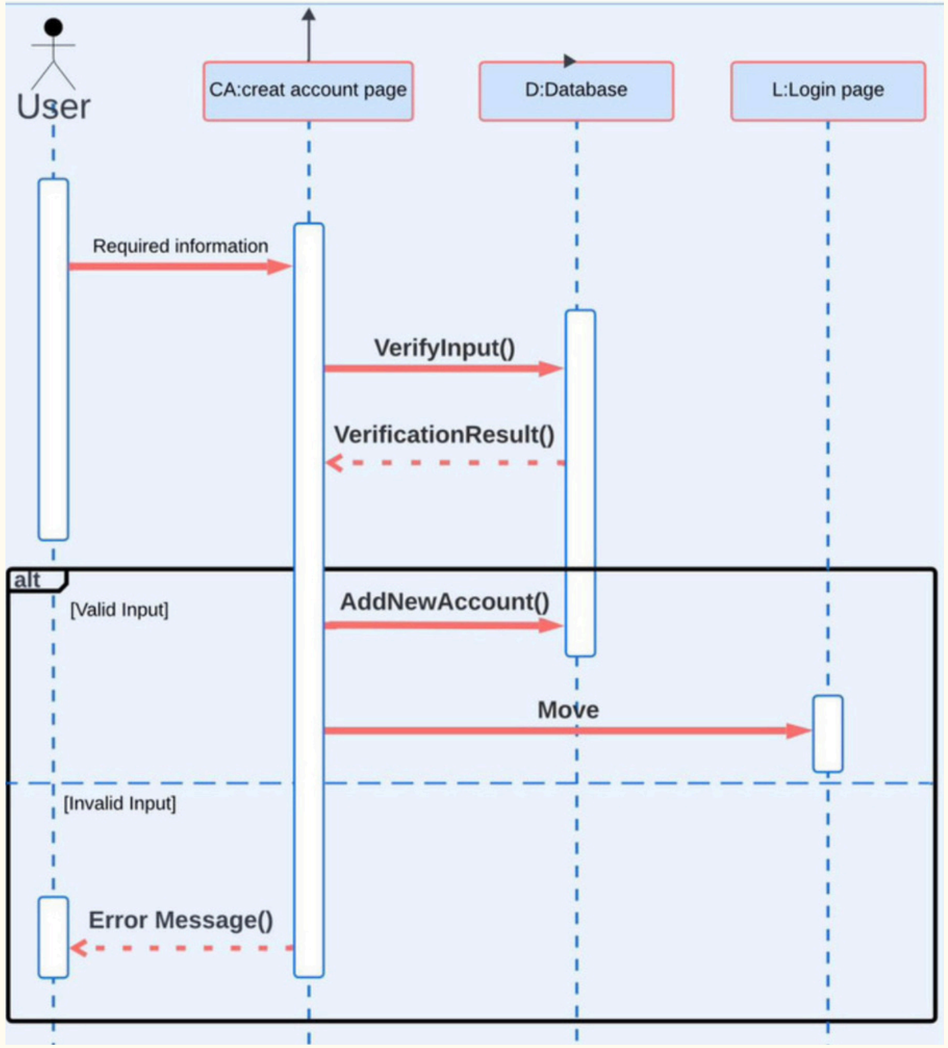
Activity diagram Notification



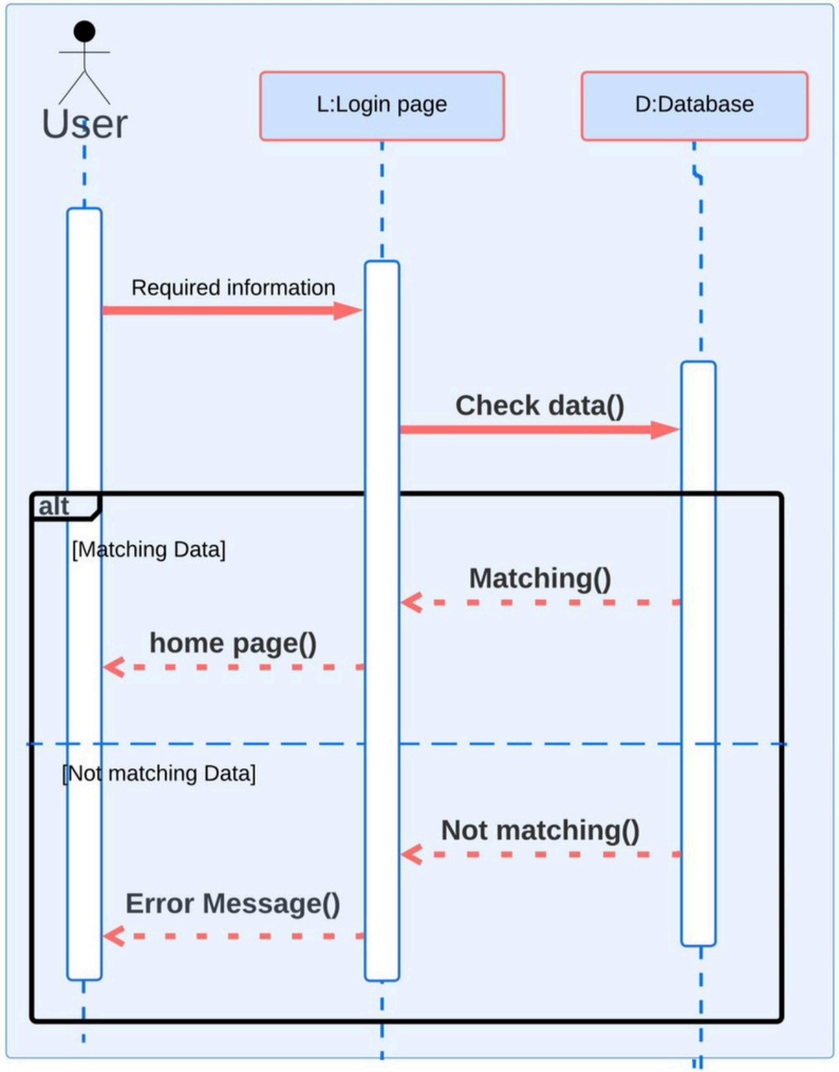
Activity diagram coustomer services



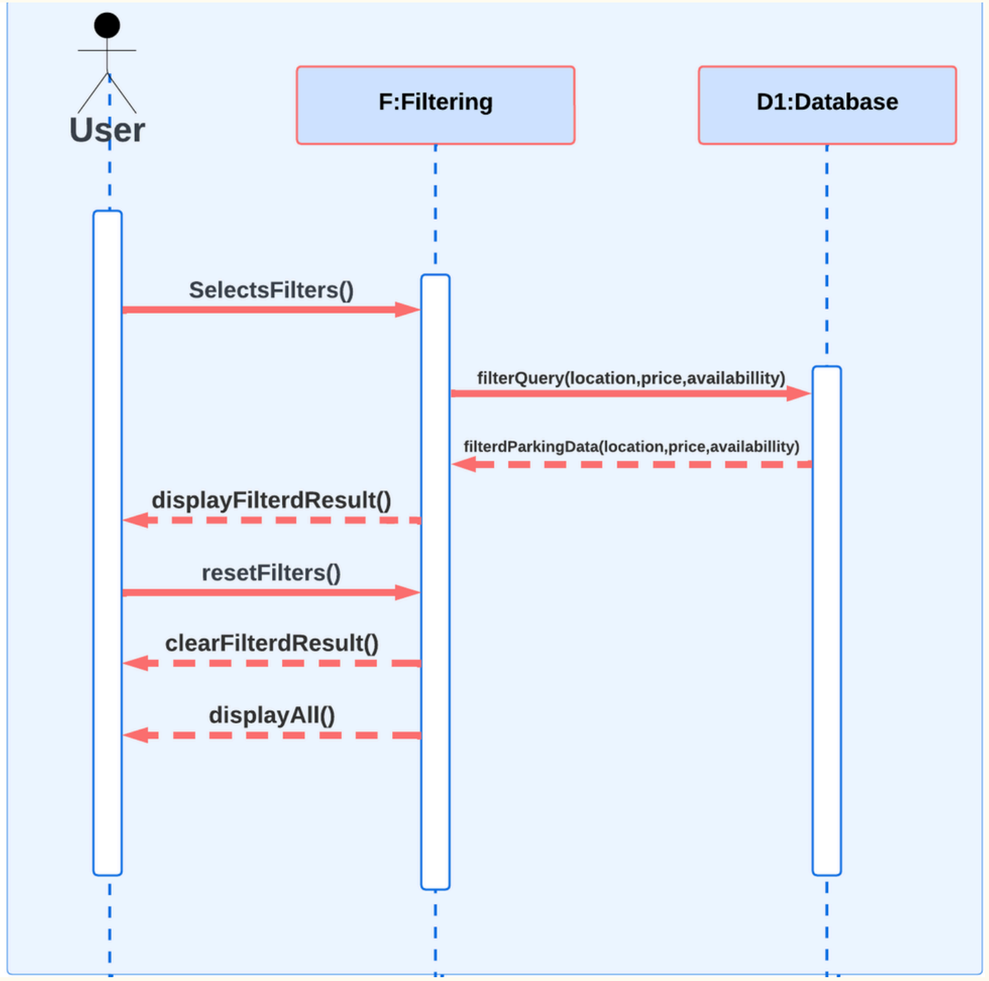
Sequence diagram sign up



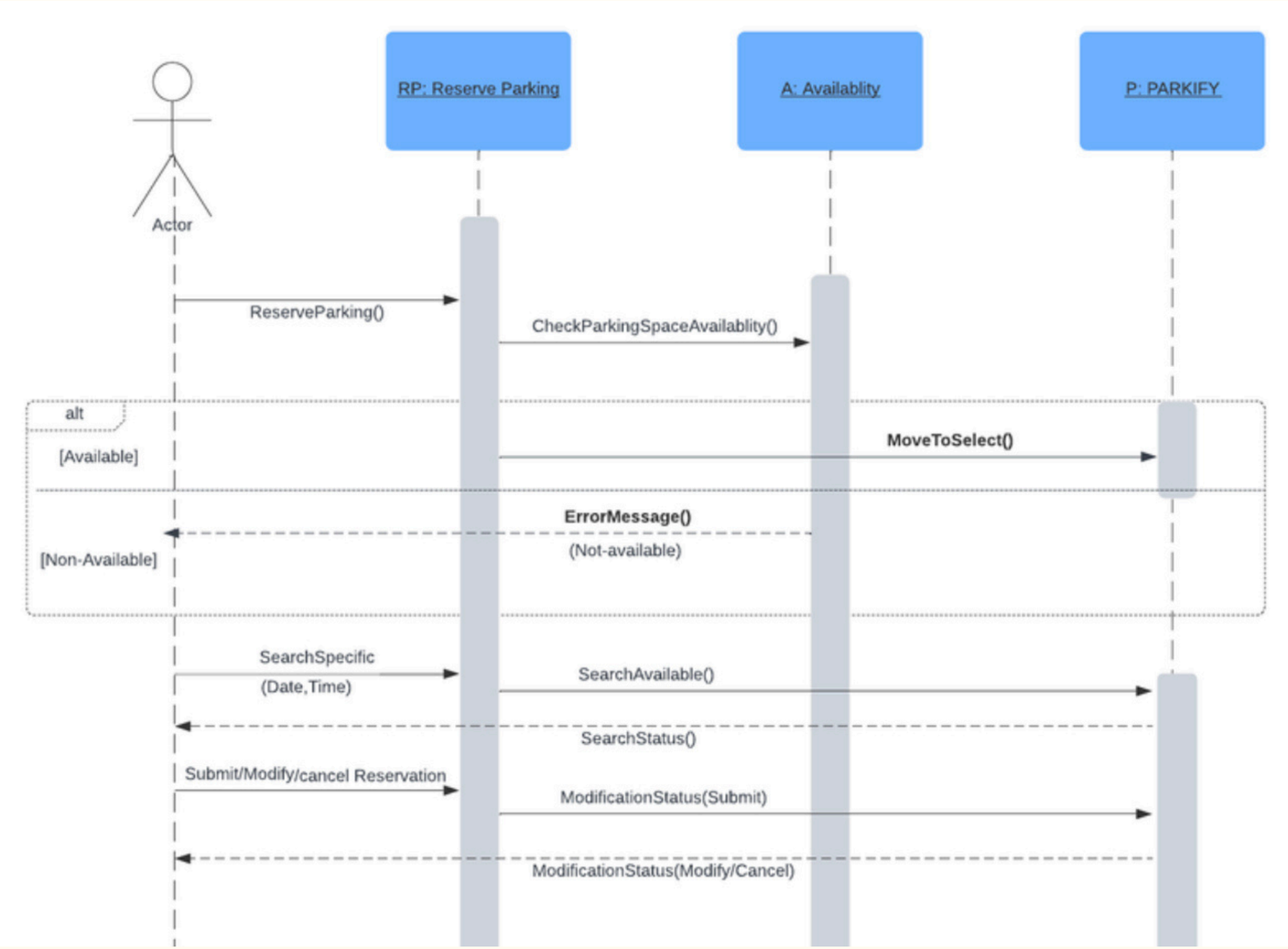
Sequence diagram Log In



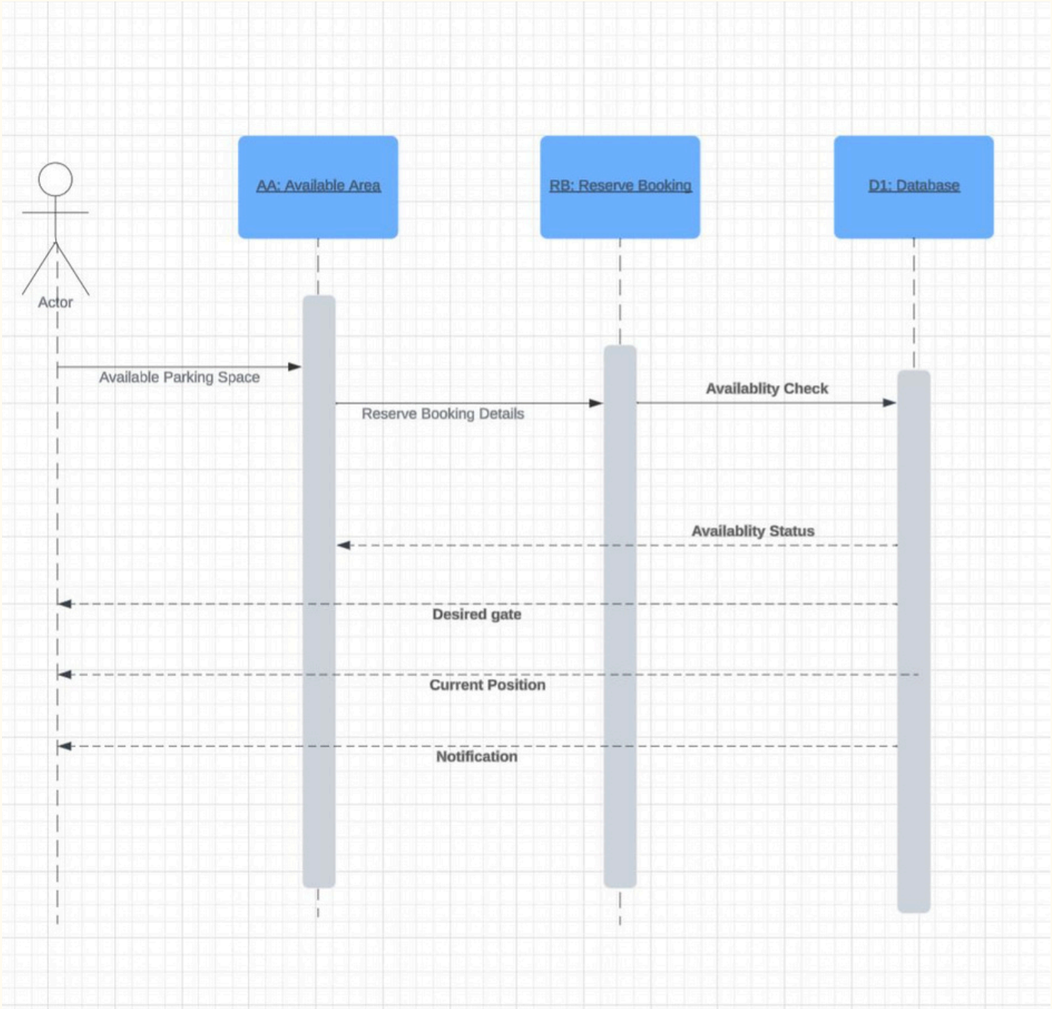
Sequence diagram Filtering



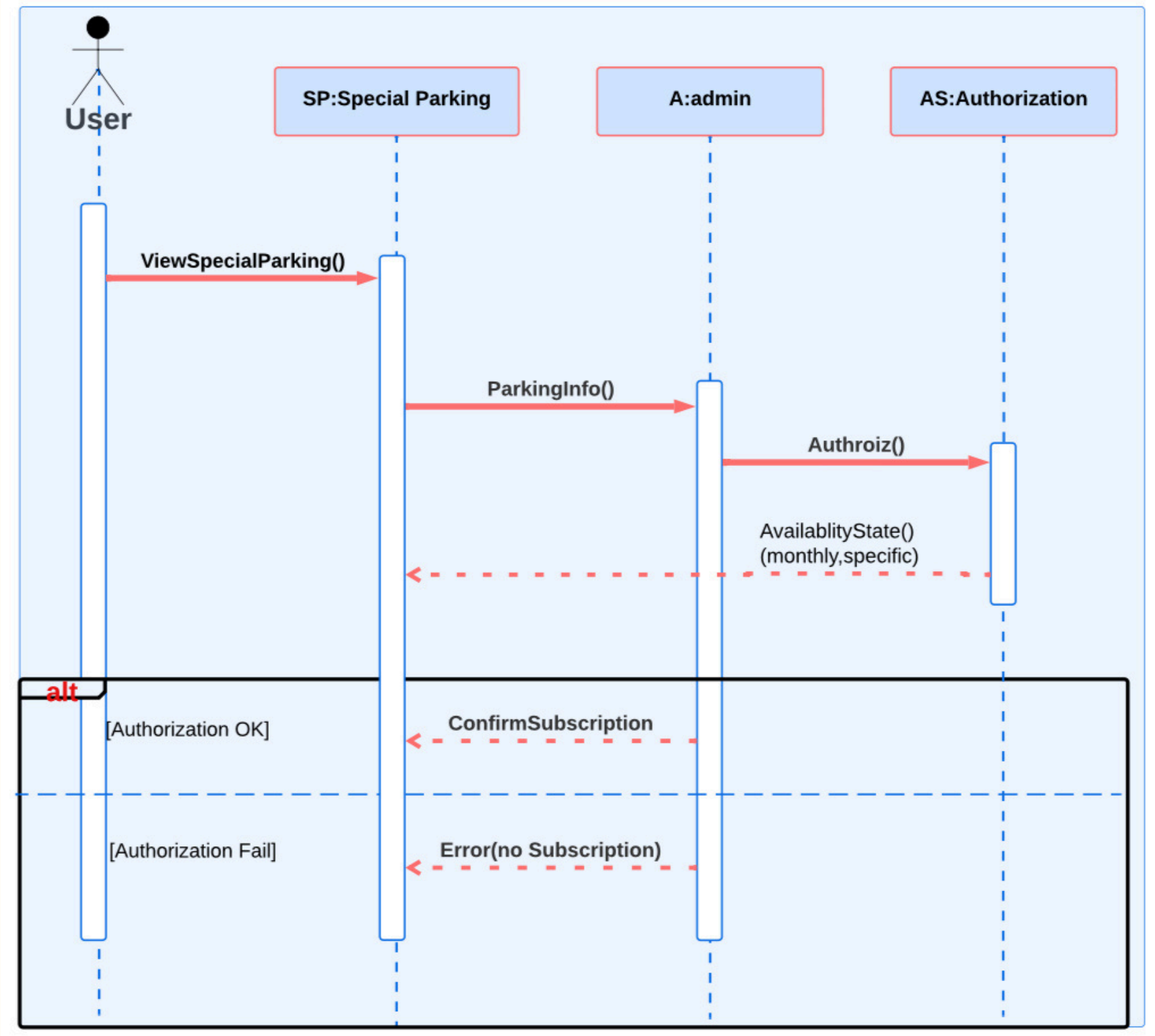
Sequence diagram Reservation system



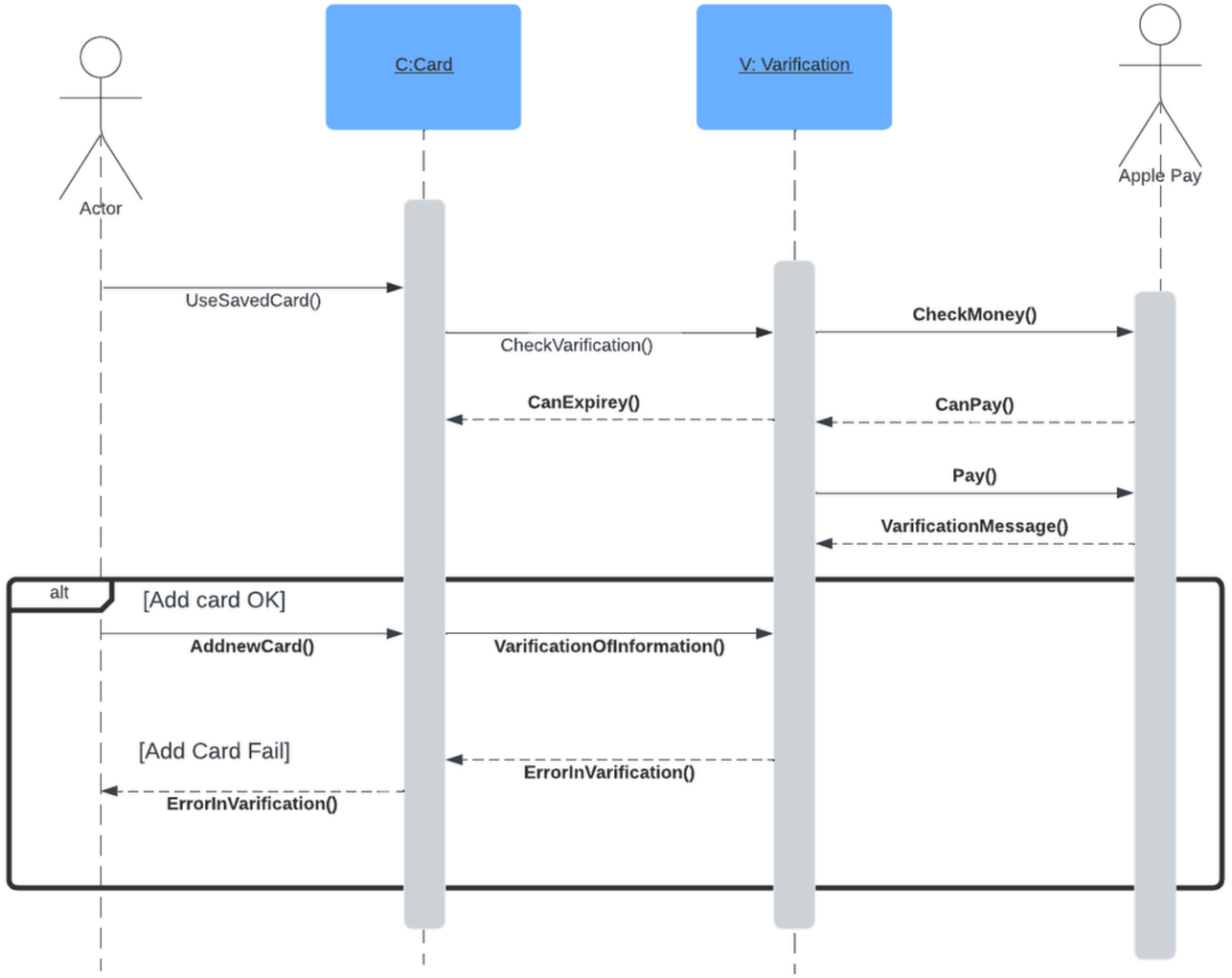
Sequence diagram Booking location



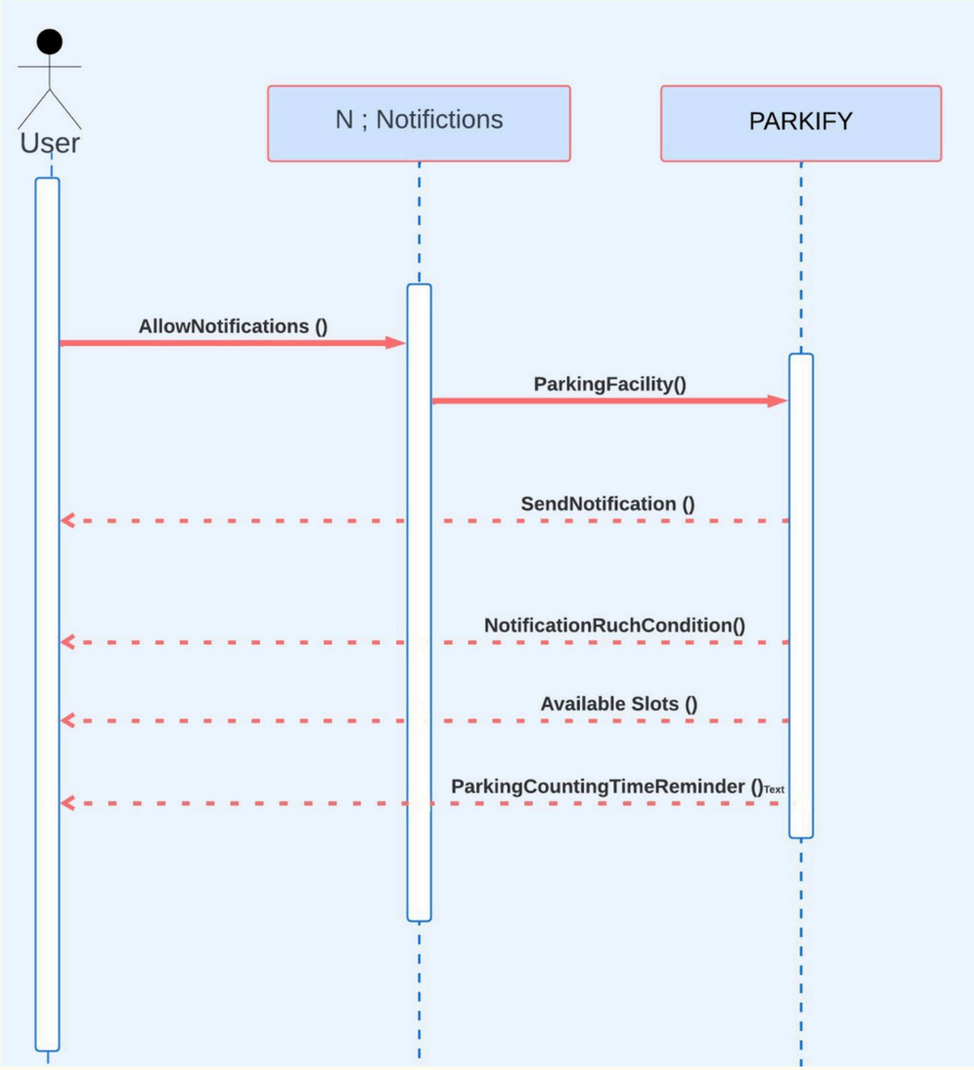
Sequence diagram Special Parking



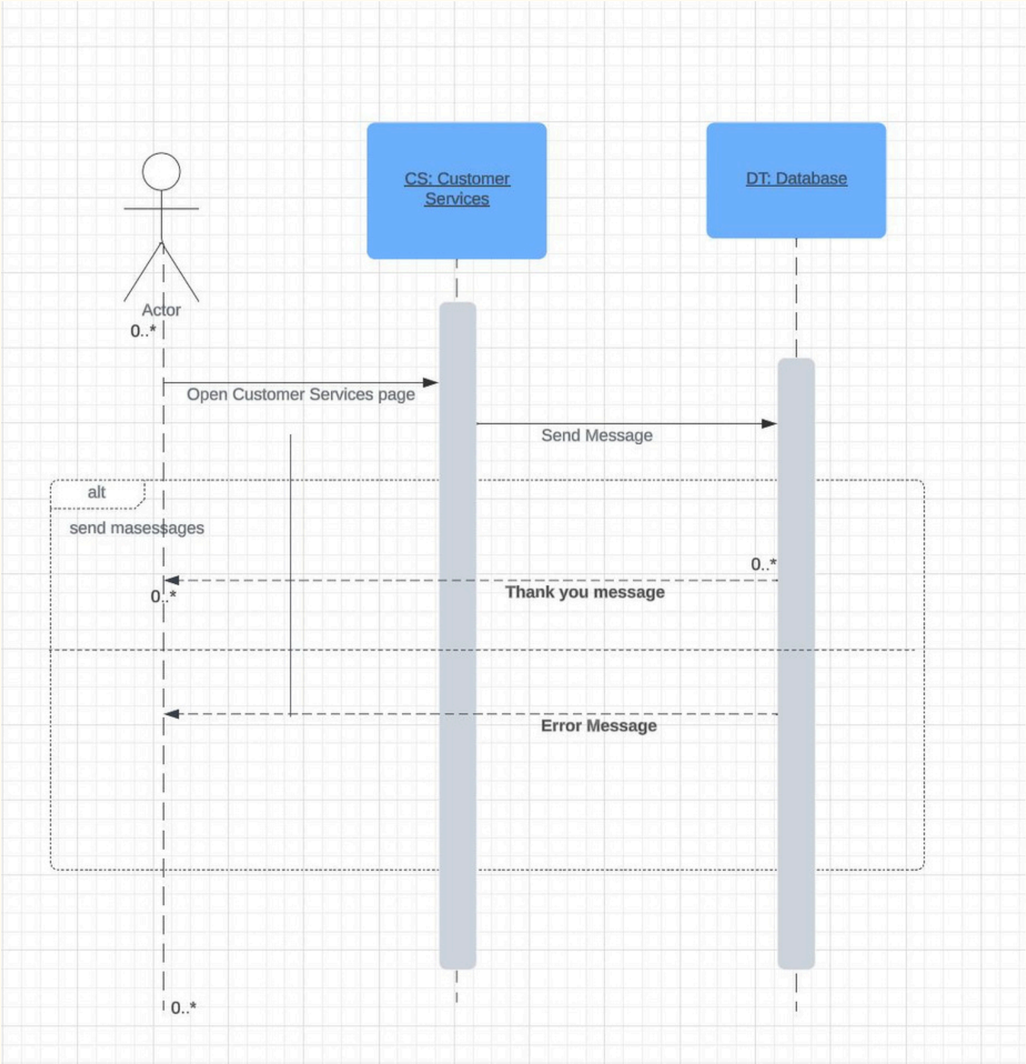
Sequence diagram Payment



Sequence diagram Notification

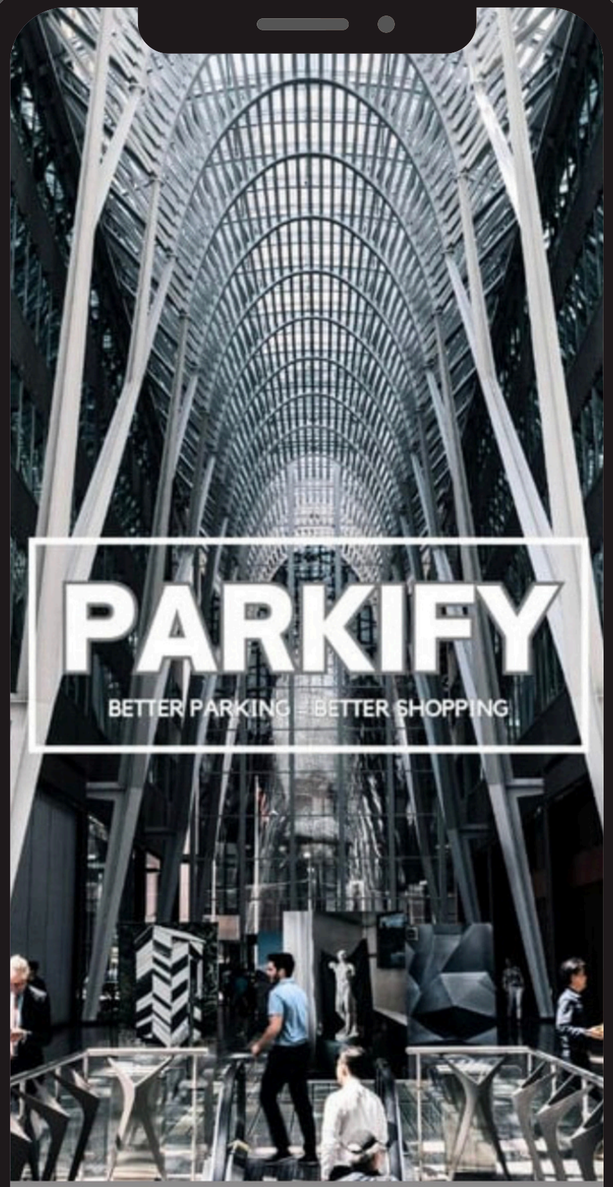


Sequence diagram coustomer services

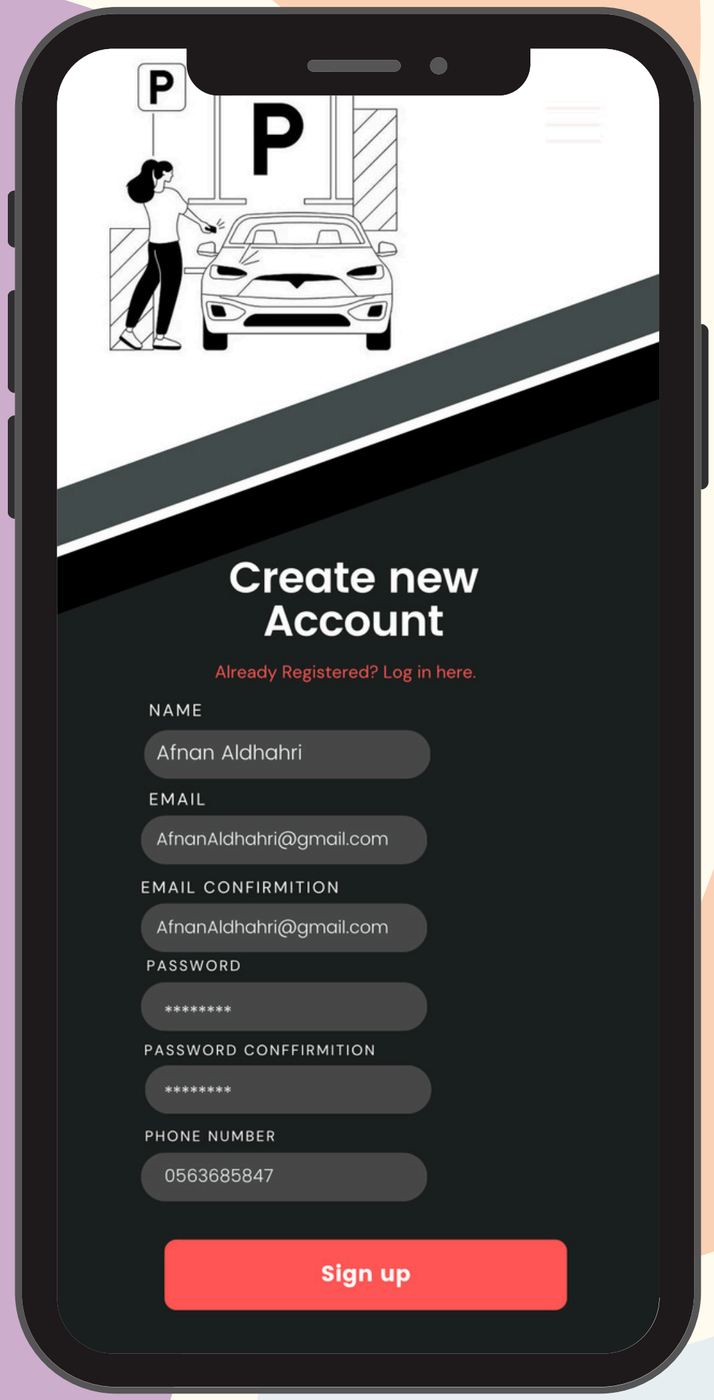


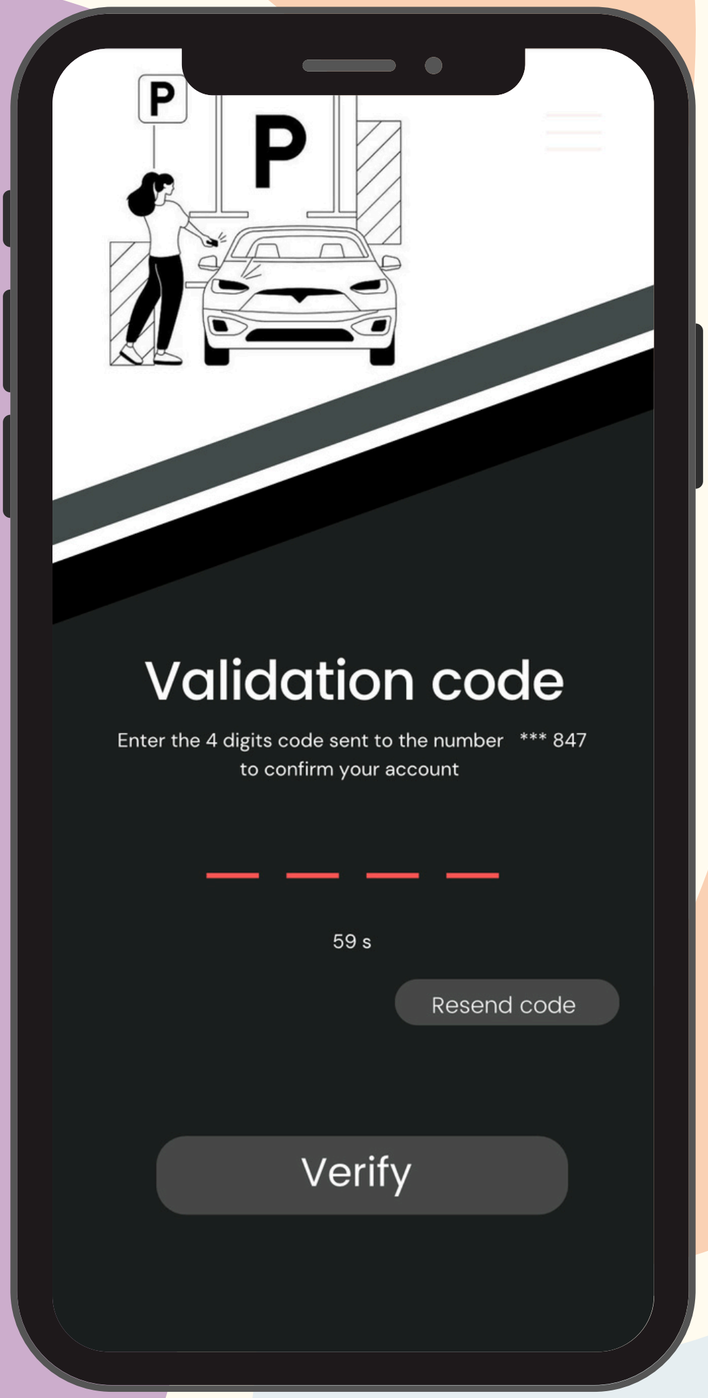
Prototype

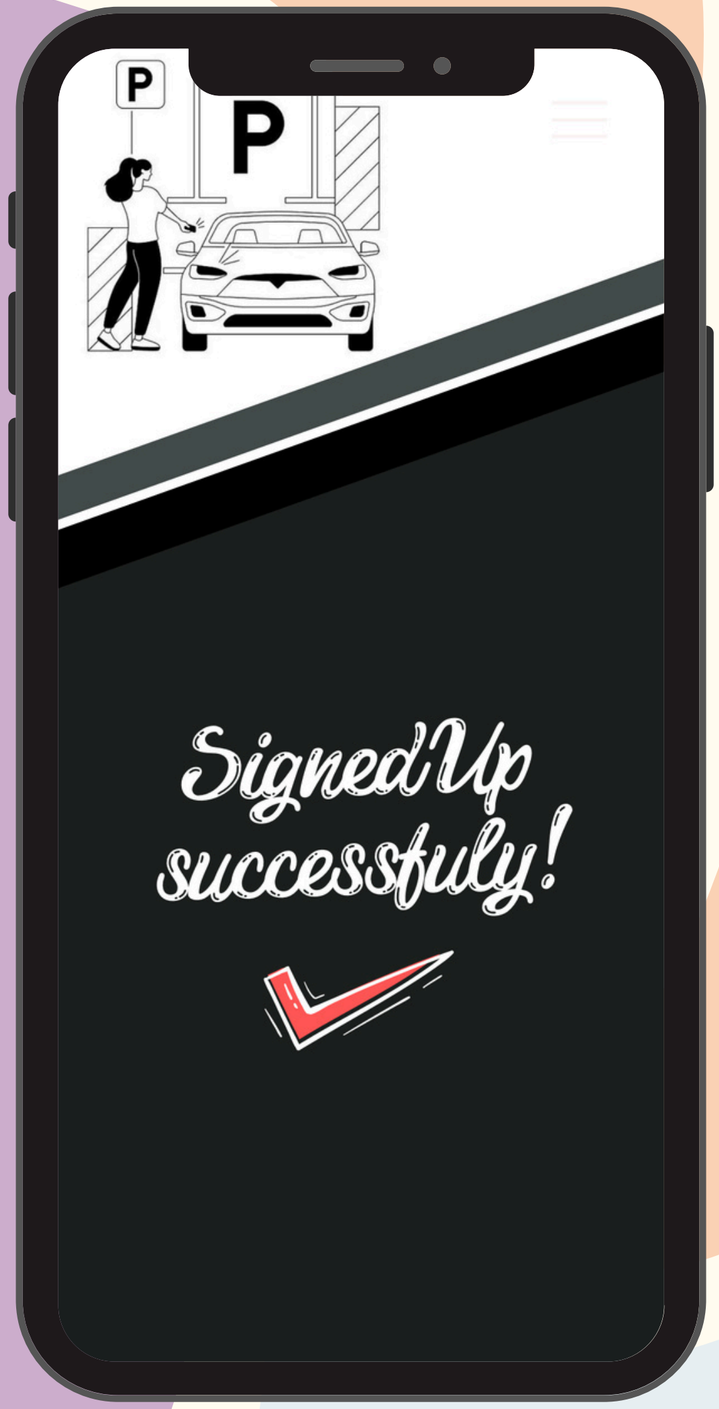
*

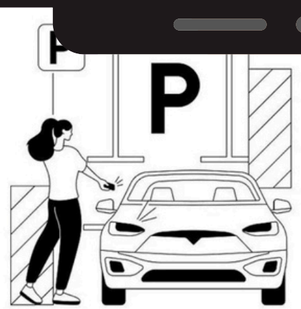


To avoid delay, which always happens when going to
the mall due to crowded parking and lack of organization


P
P
Create new Account
Already Registered? Log in here.
NAME
Afnan Aldhahri
EMAIL
AfnanAldhahri@gmail.com
EMAIL CONFIRMITION
AfnanAldhahri@gmail.com
PASSWORD
****
PASSWORD CONFFIRMITION
****
PHONE NUMBER
0563685847
Sign up


P
P
Validation code
Enter the 4 digits code sent to the number *** 847
to confirm your account
59 s
Resend code
Verify


P
P
SignedUp
successfuly!



Login

Log in to continue.

EMAIL/PHONE NUMBER

AfnanAldhahri@gmail.com

PASSWORD

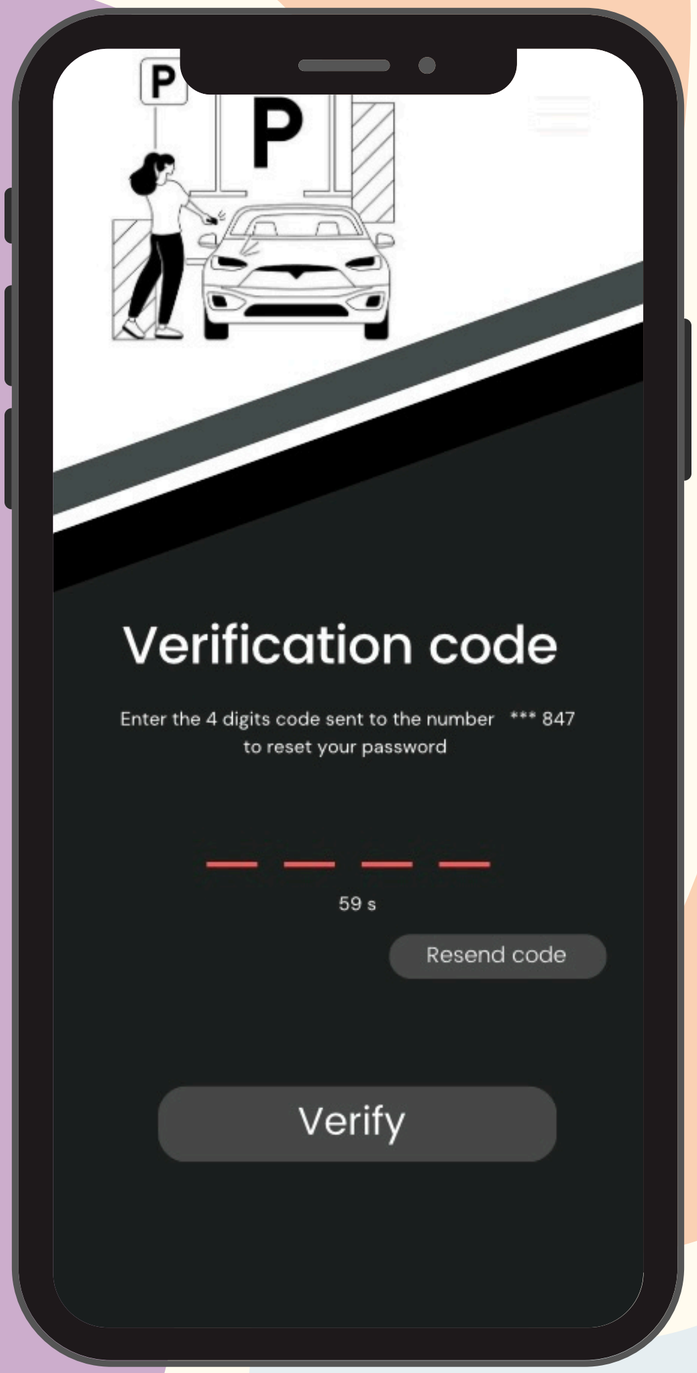
******

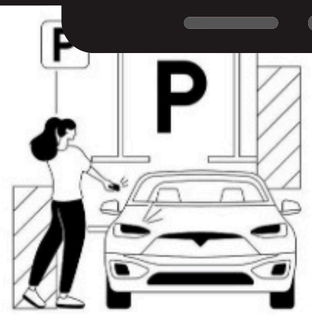
Log in

Keep me signed in ☐

Forgot Password? Rest it

Don't have an account? Sign up


P
P
Verification code
Enter the 4 digits code sent to the number *** 847
to reset your password
59 s
Resend code
Verify



Reset password

NEW PASSWORD

******

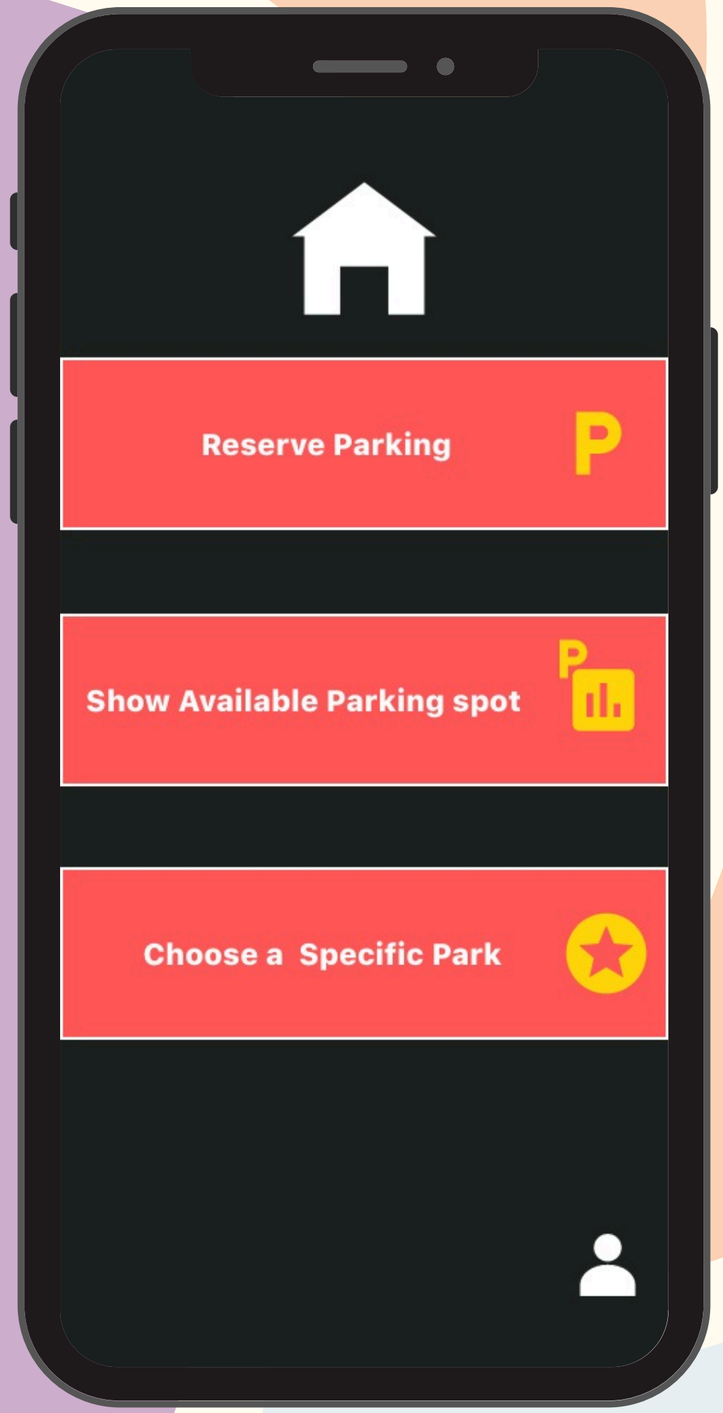
CONFIRM NEW PSSWORD

******

Submit

Reset password successful!

Log in now


Reserve Parking
P
Show Available Parking spot
P
Choose a Specific Park

Lets reserve a parking spot about your car

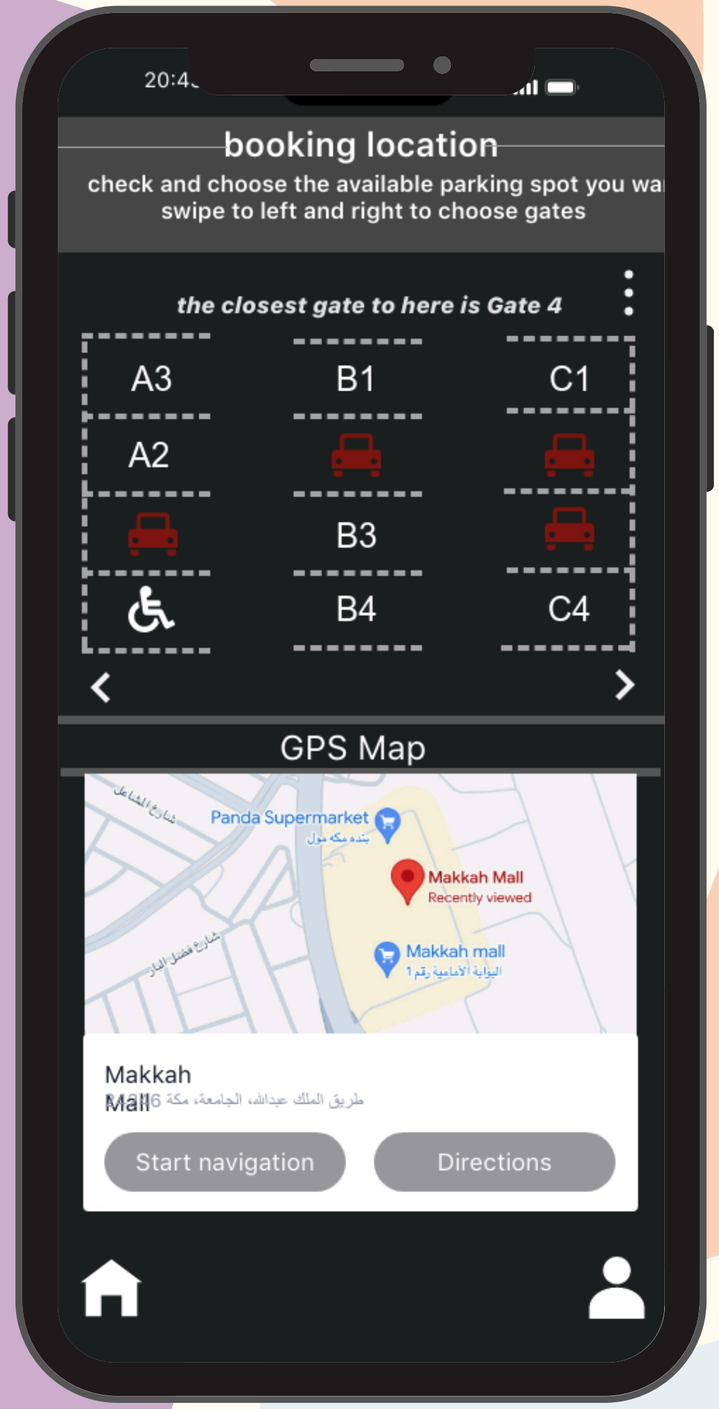
Select Date

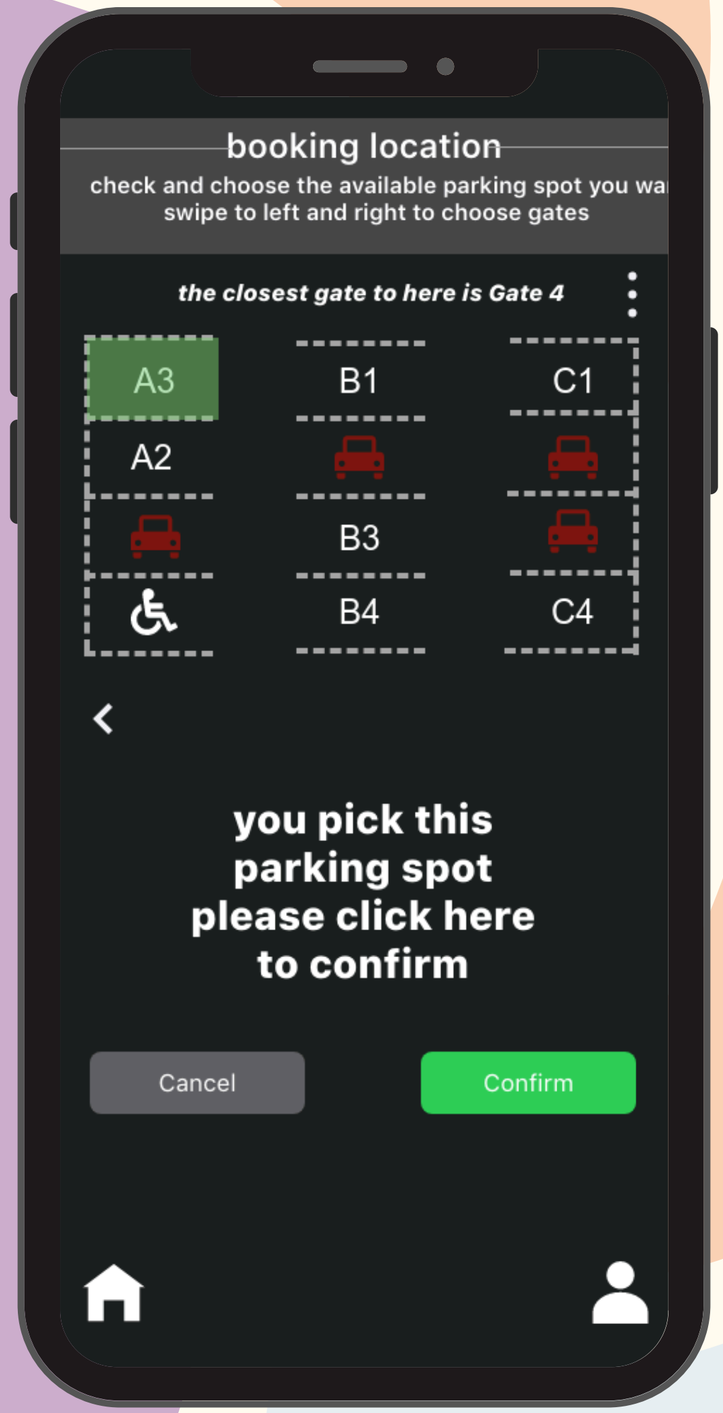
Month Year

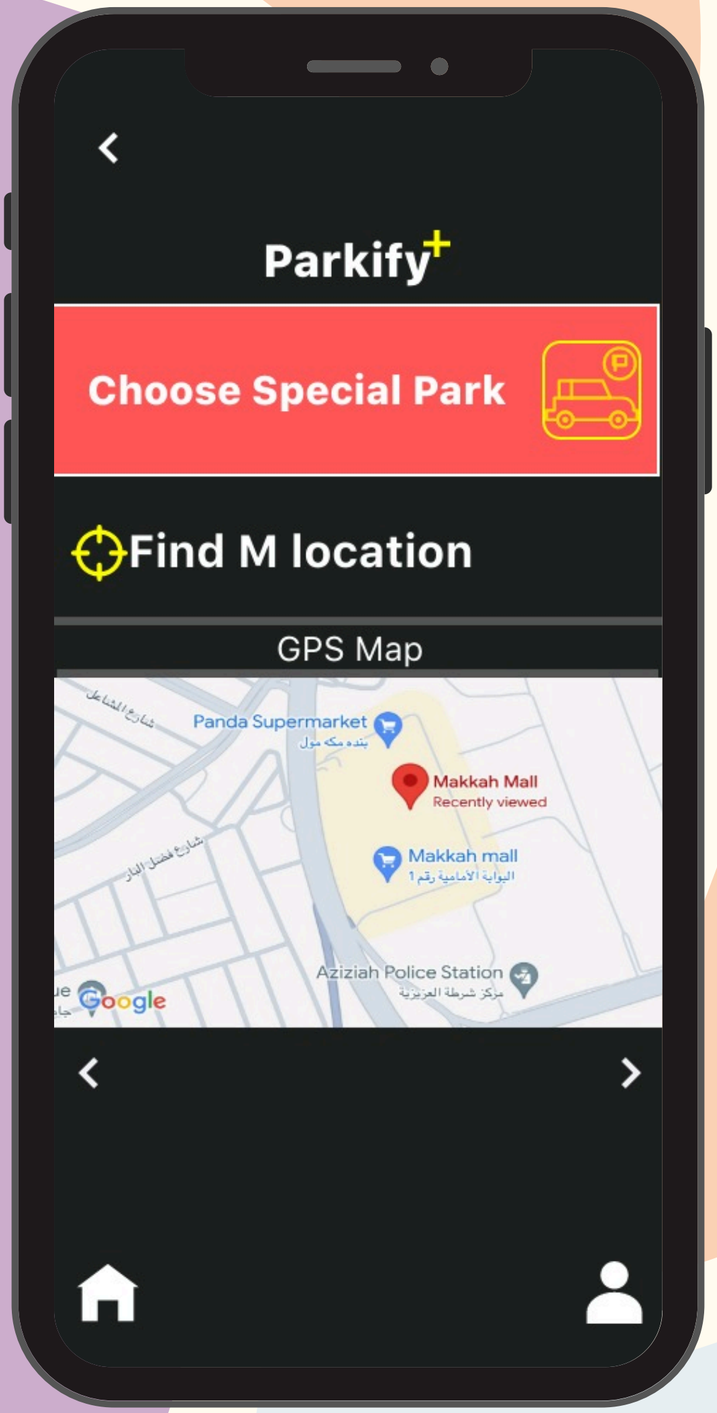
Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30	31	01	02	03	04
05	06	07	08	09	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	01

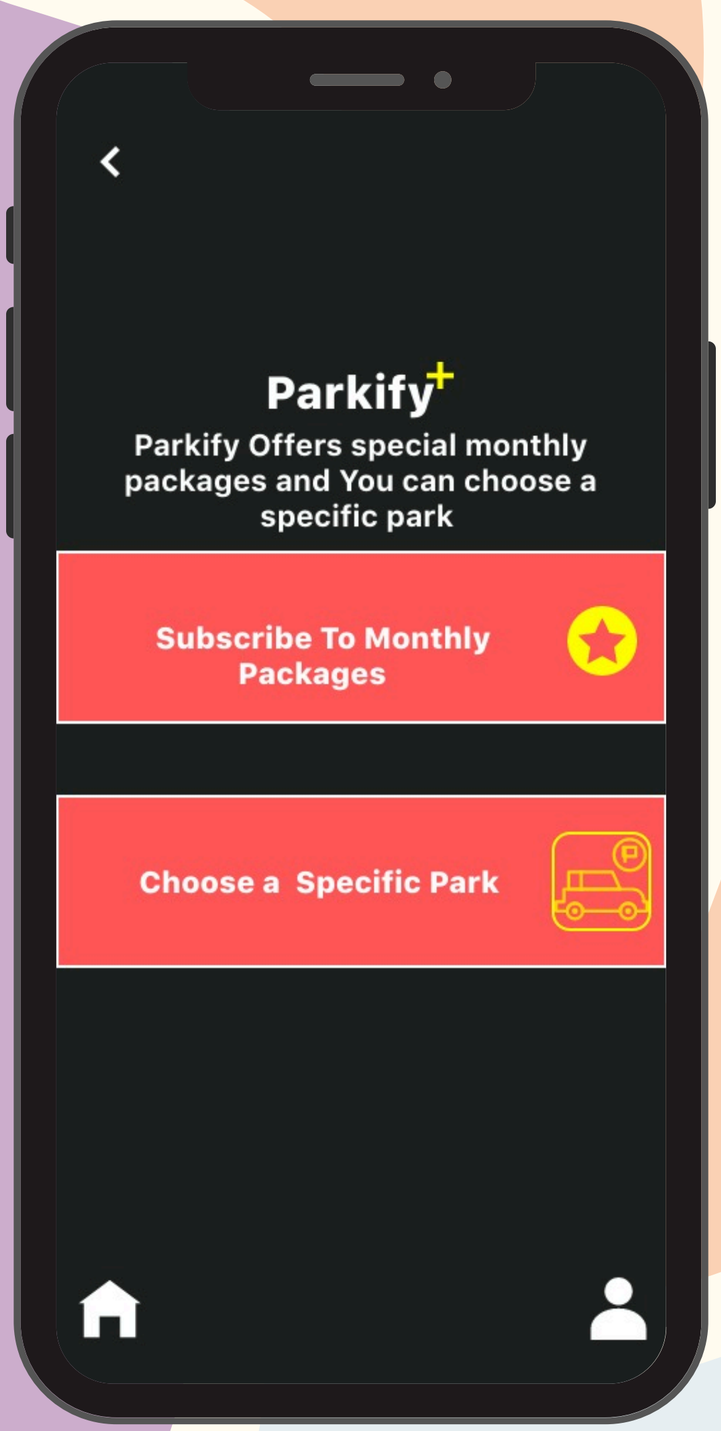
TIME : 8:00PM to 10:30PM

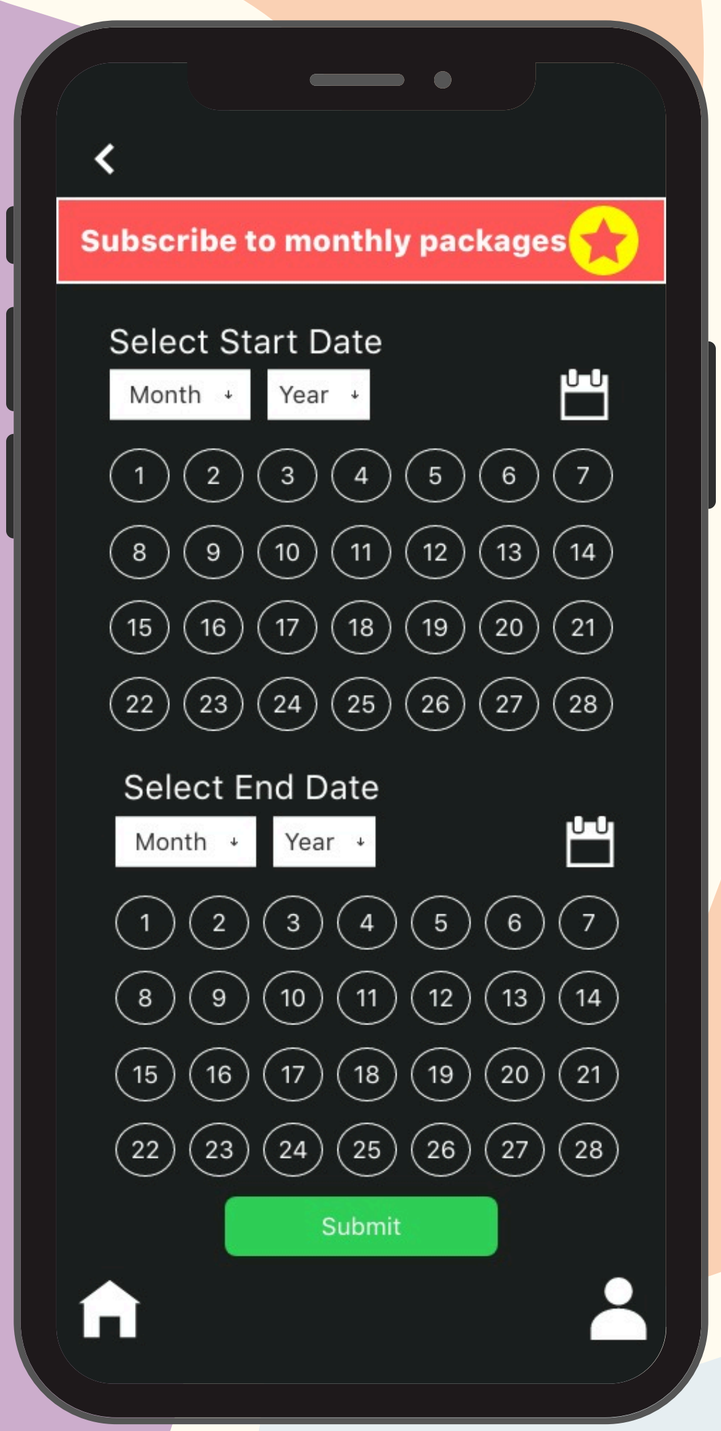
SUBMIT

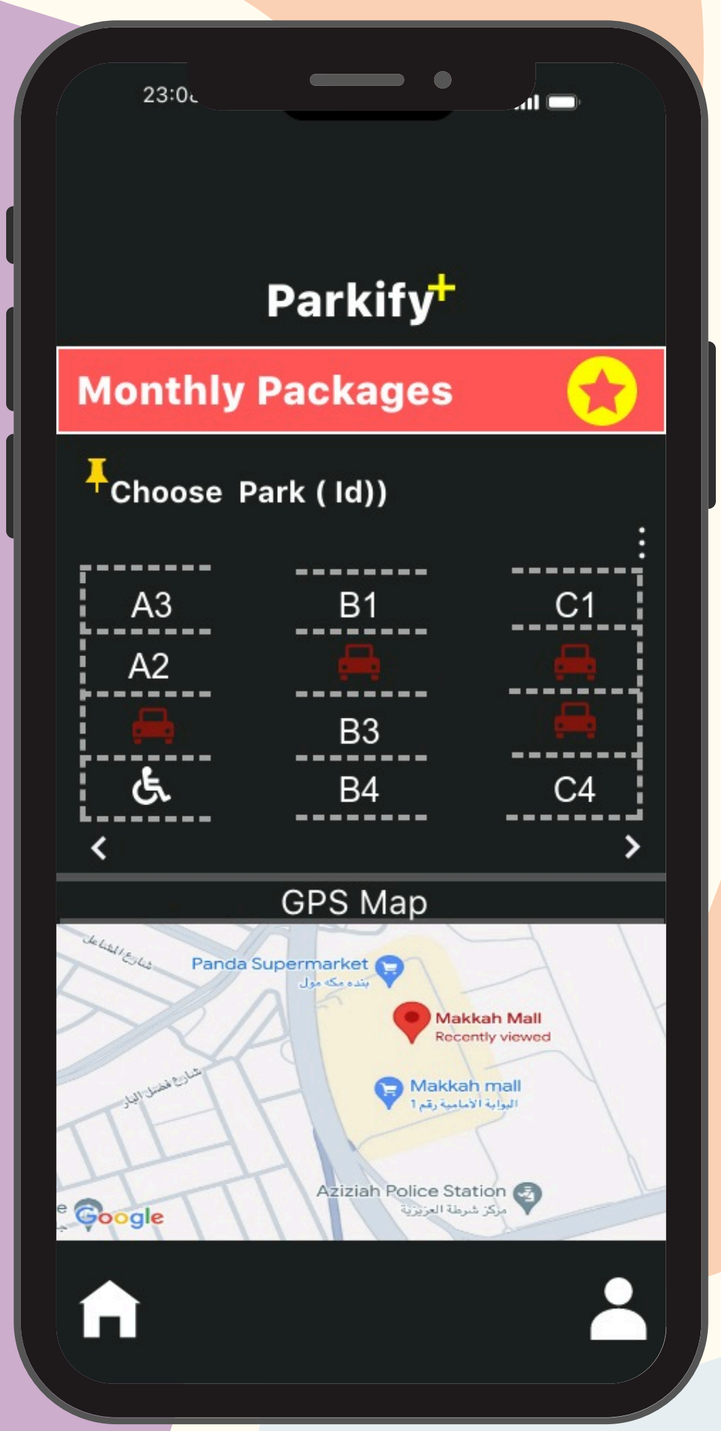

20:4
booking location
check and choose the available parking spot you wa
swipe to left and right to choose gates
the closest gate to here is Gate 4
A3
B1
C1
A2
B3
B4
C4
GPS Map
Panda Supermarket
Makkah Mall
Recently viewed
Makkah mall
Makkah
Mall
Start navigation
Directions

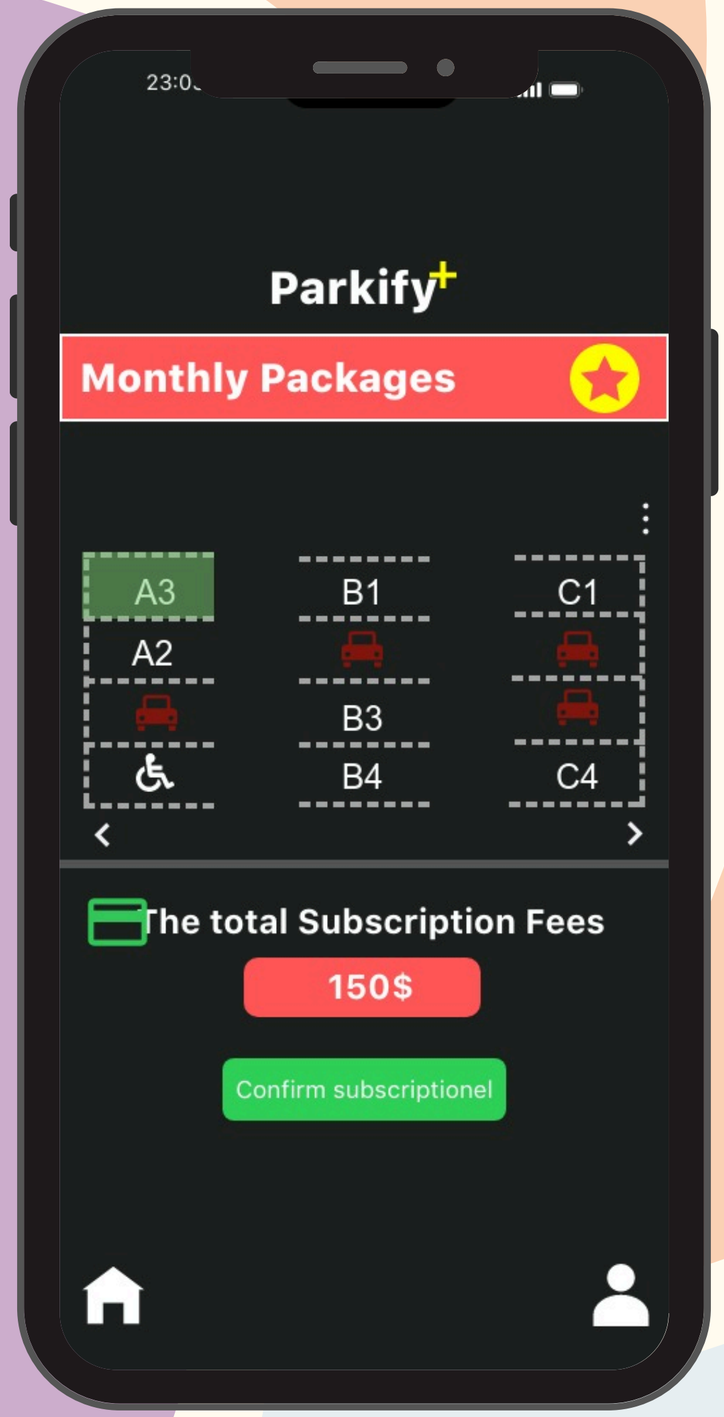

booking location
check and choose the available parking spot you wa
swipe to left and right to choose gates
the closest gate to here is Gate 4
A3
B1
C1
A2
B3
B4
C4
you pick this parking spot please click here to confirm
Cancel
Confirm

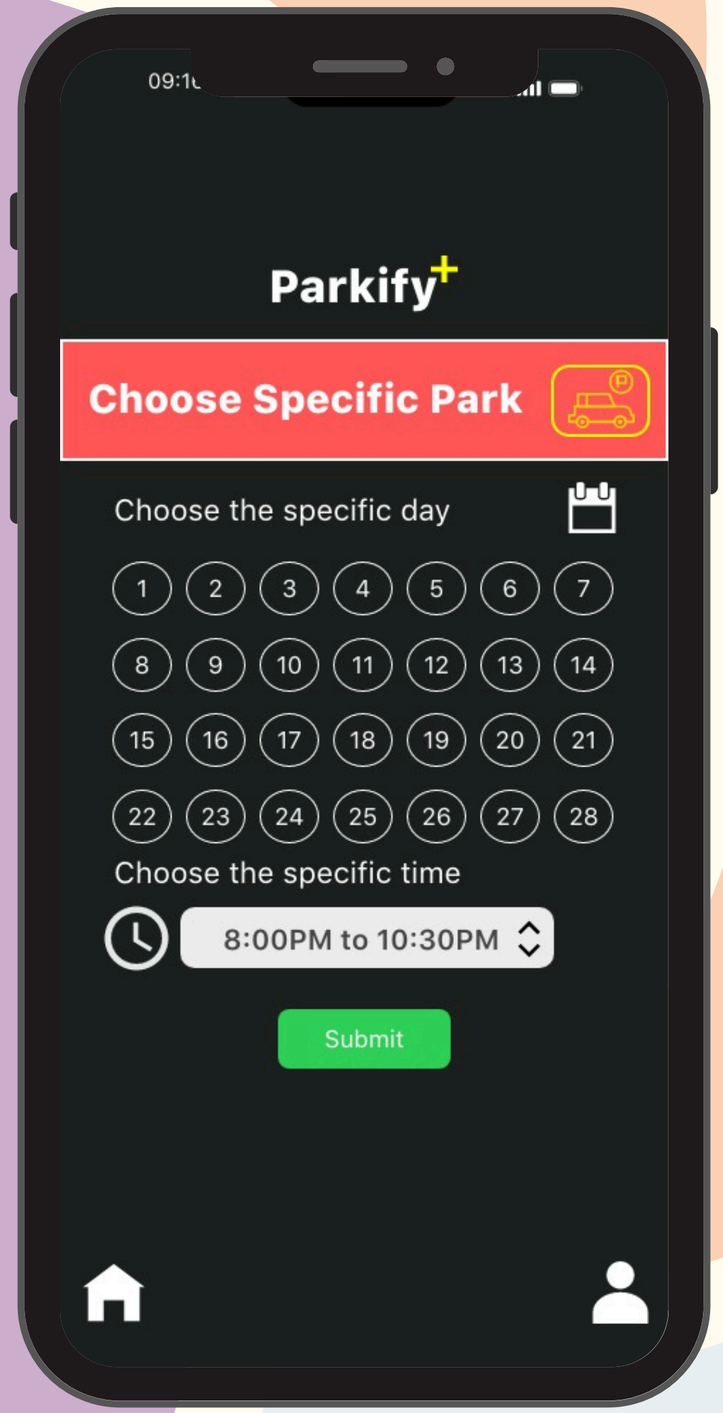

Parkify+
Choose Special Park
Find M location
GPS Map
Panda Supermarket
Makkah Mall
Recently viewed
Makkah mall
Aziziah Police Station
Google

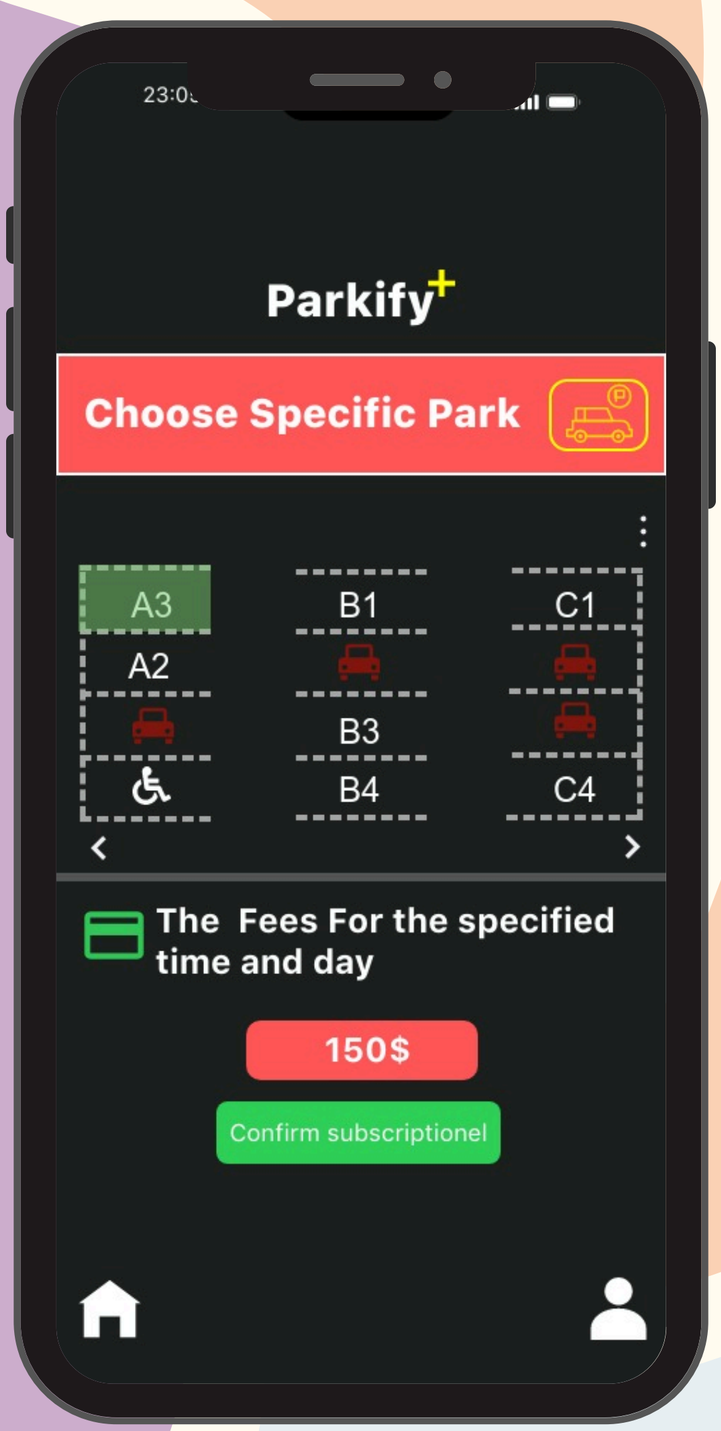

Parkify+
Parkify Offers special monthly packages and You can choose a specific park
Subscribe To Monthly Packages
Choose a Specific Park

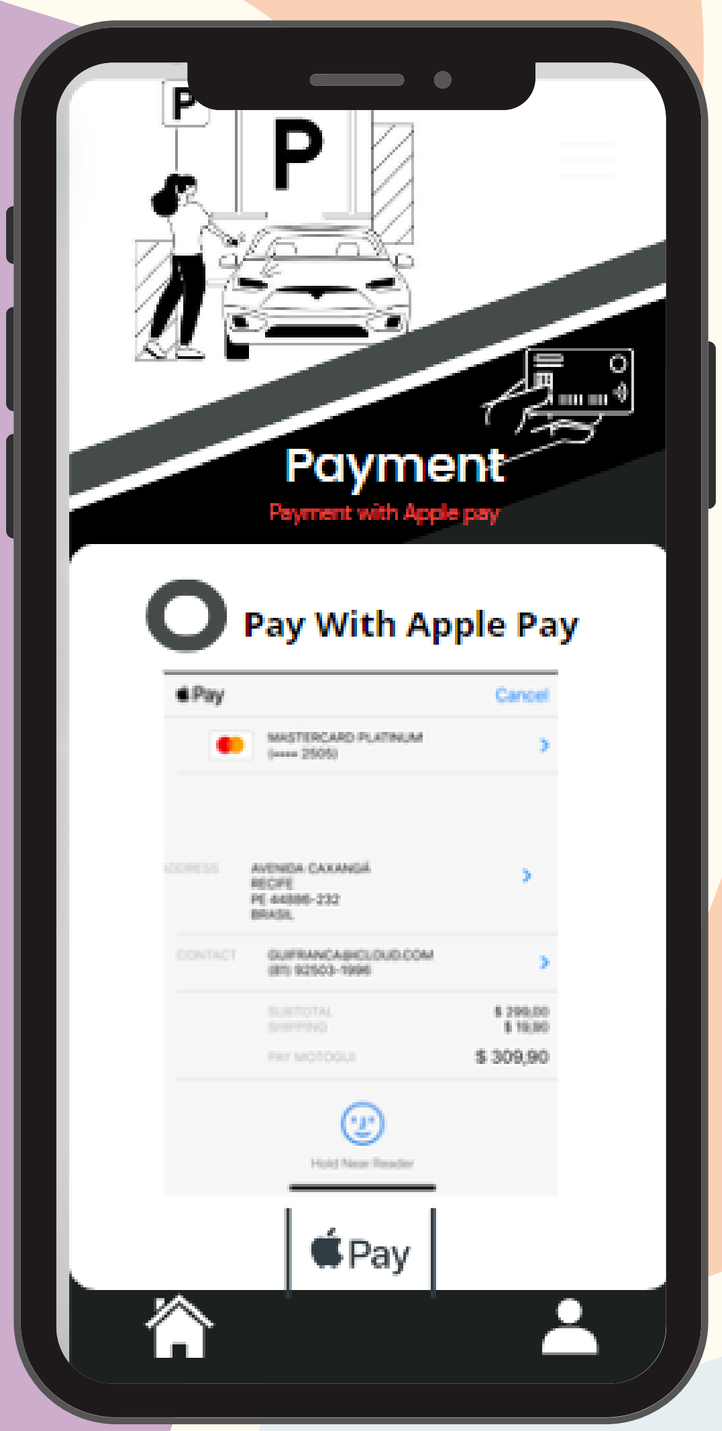

Subscribe to monthly packages
Select Start Date
Month
Year
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
Select End Date
Month
Year
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
Submit


Parkify+
Monthly Packages
Choose Park (Id))
A3
B1
C1
A2
B3
B4
C4
GPS Map
Panda Supermarket
Makkah Mall
Recently viewed
Makkah mall
Aziziah Police Station
Google

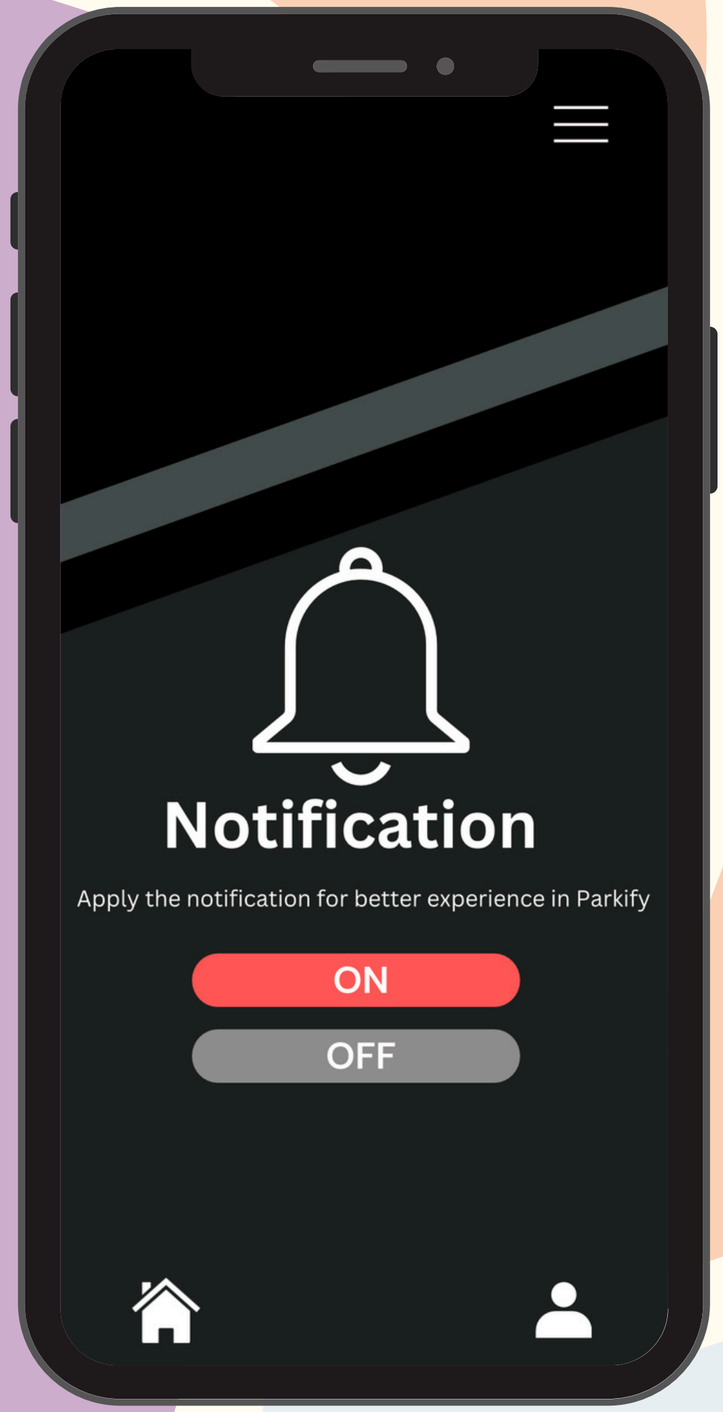

23:0
Parkify+
Monthly Packages
A3
B1
C1
A2
B3
B4
C4
The total Subscription Fees
150$
Confirm subscriptionel

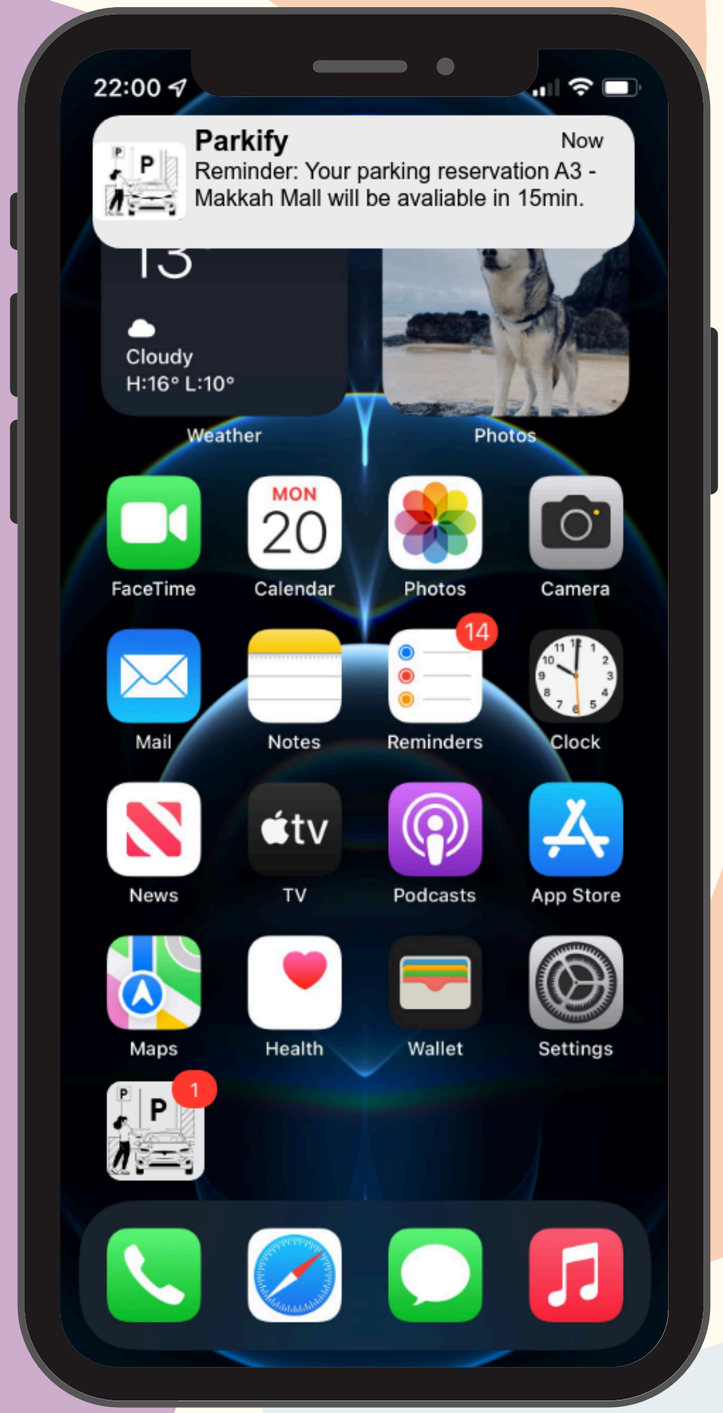

09:1
Parkify+
Choose Specific Park
Choose the specific day
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
Choose the specific time
8:00PM to 10:30PM
Submit

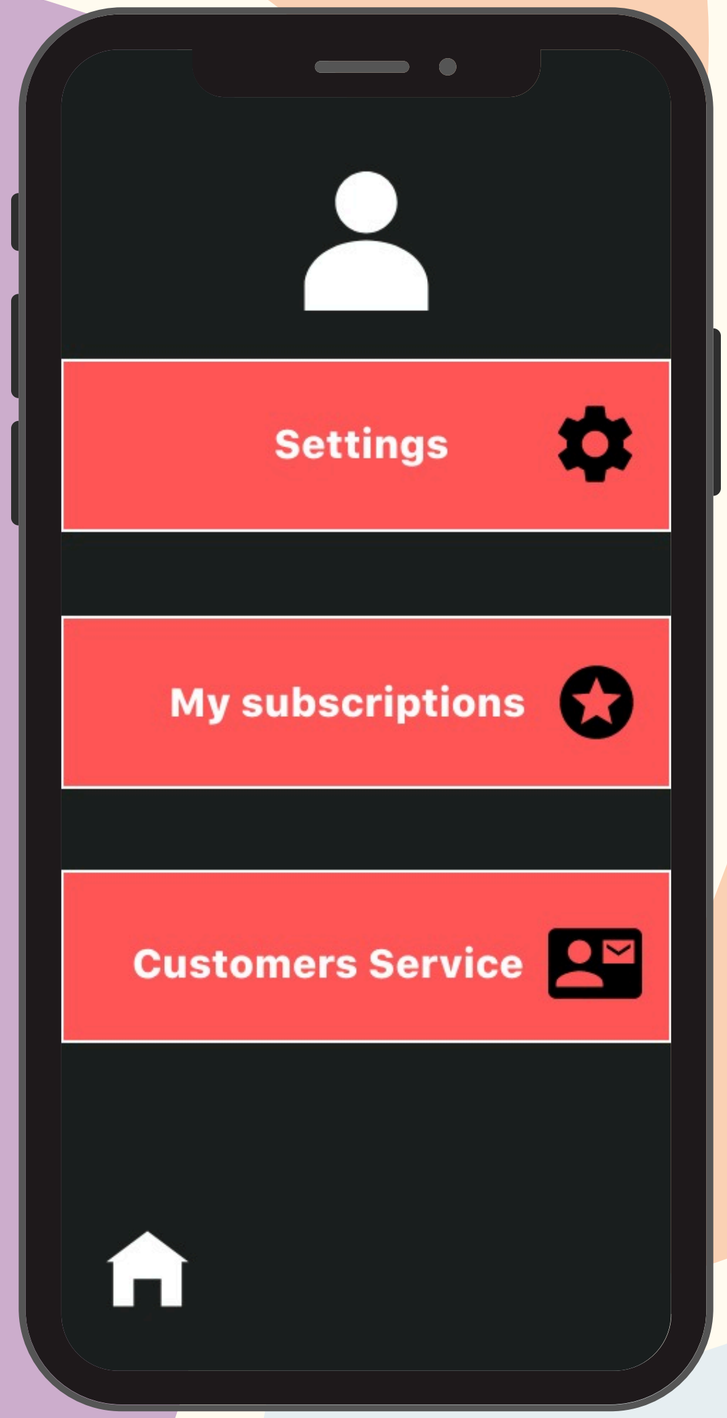

23:0
Parkify+
Choose Specific Park
A3
B1
C1
A2
B3
B4
C4
The Fees For the specified time and day
150$
Confirm subscriptionel


P
Payment
Payment with Apple pay
Pay With Apple Pay
Pay
Cancel
MASTERCARD PLATINUM
AVENIDA CAXANGÁ
RECIFE
BRASIL
$ 309,90
Hold Near Reader
Pay

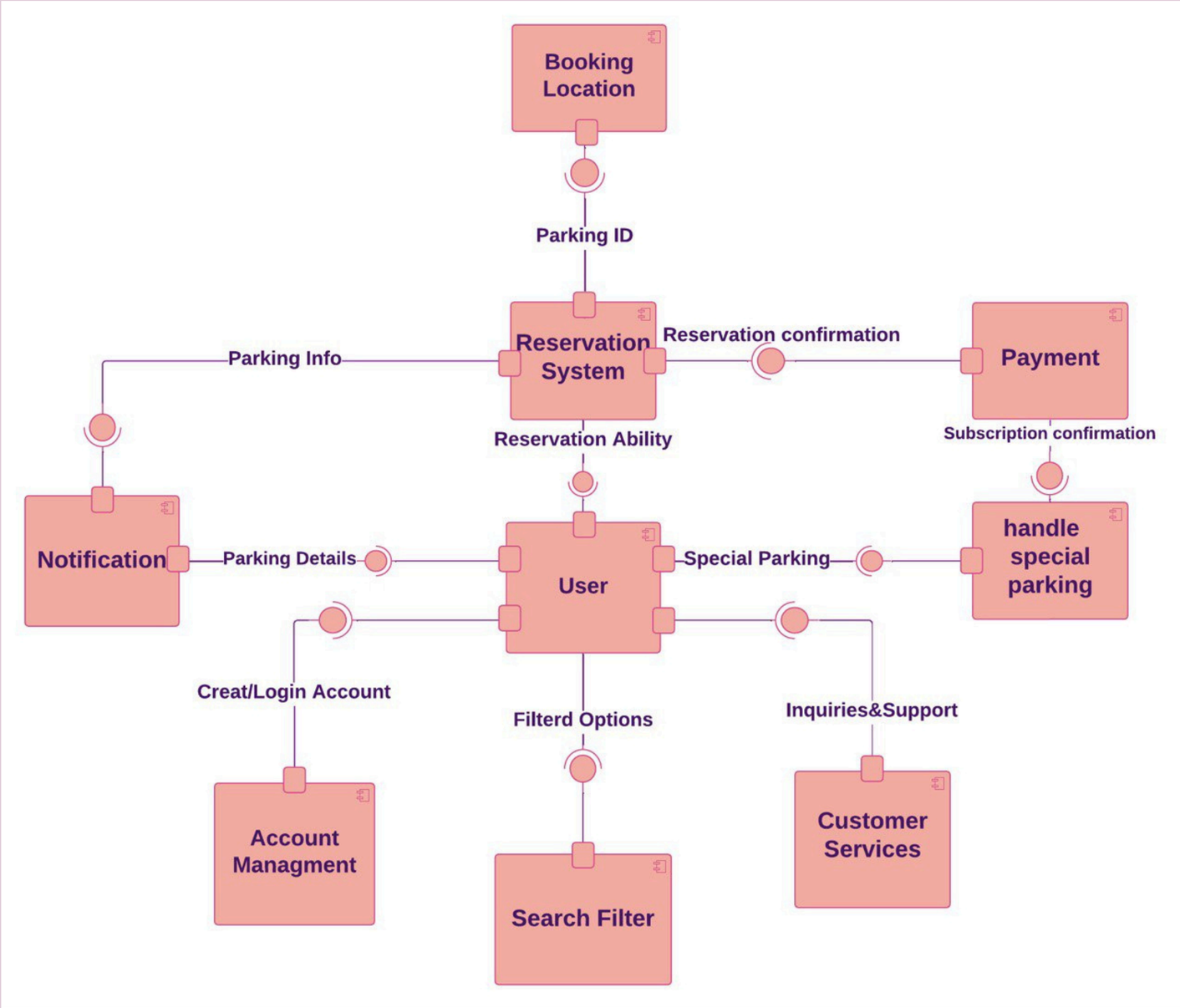
Subscribed successfully!

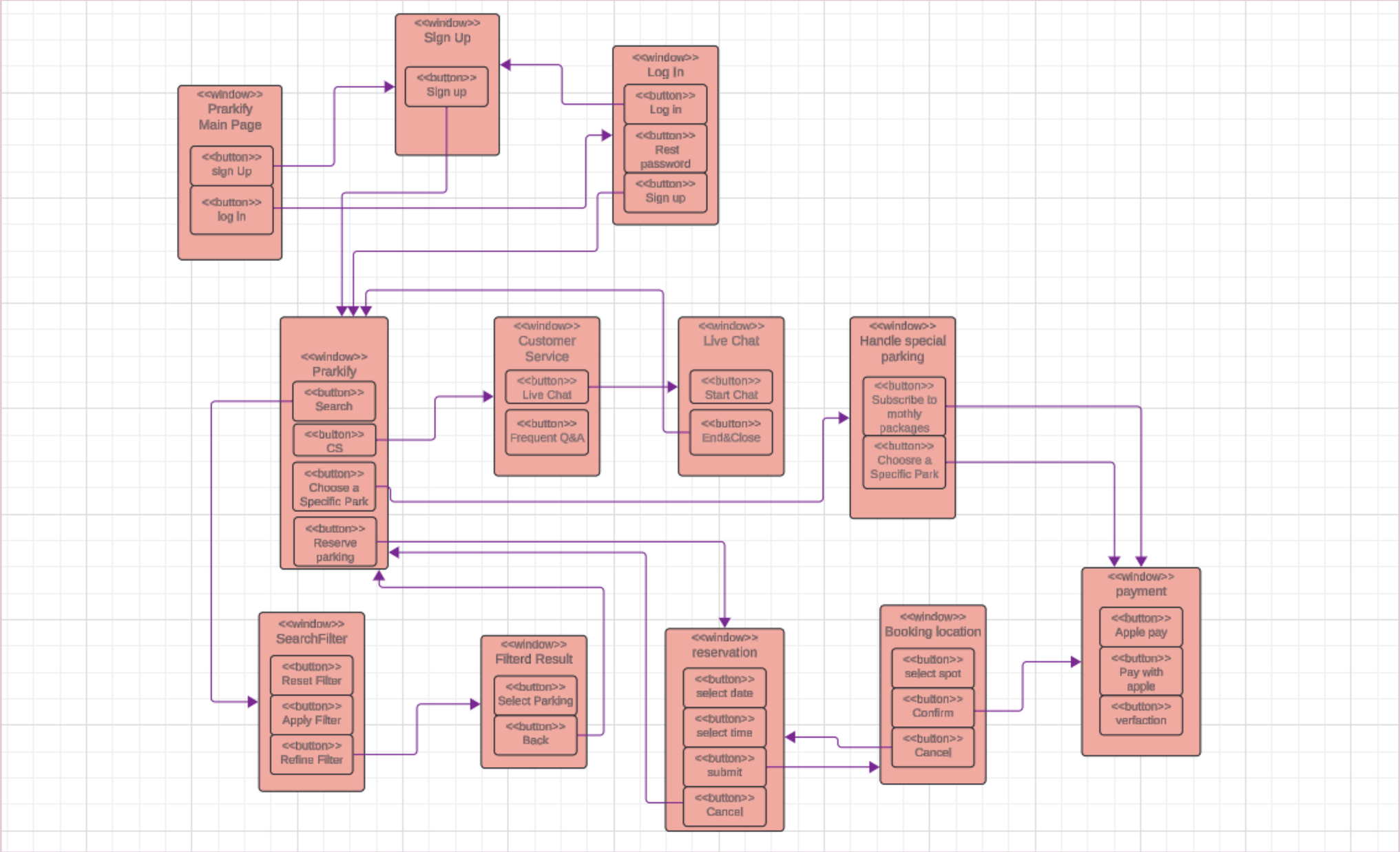

Notification
Apply the notification for better experience in Parkify
ON
OFF

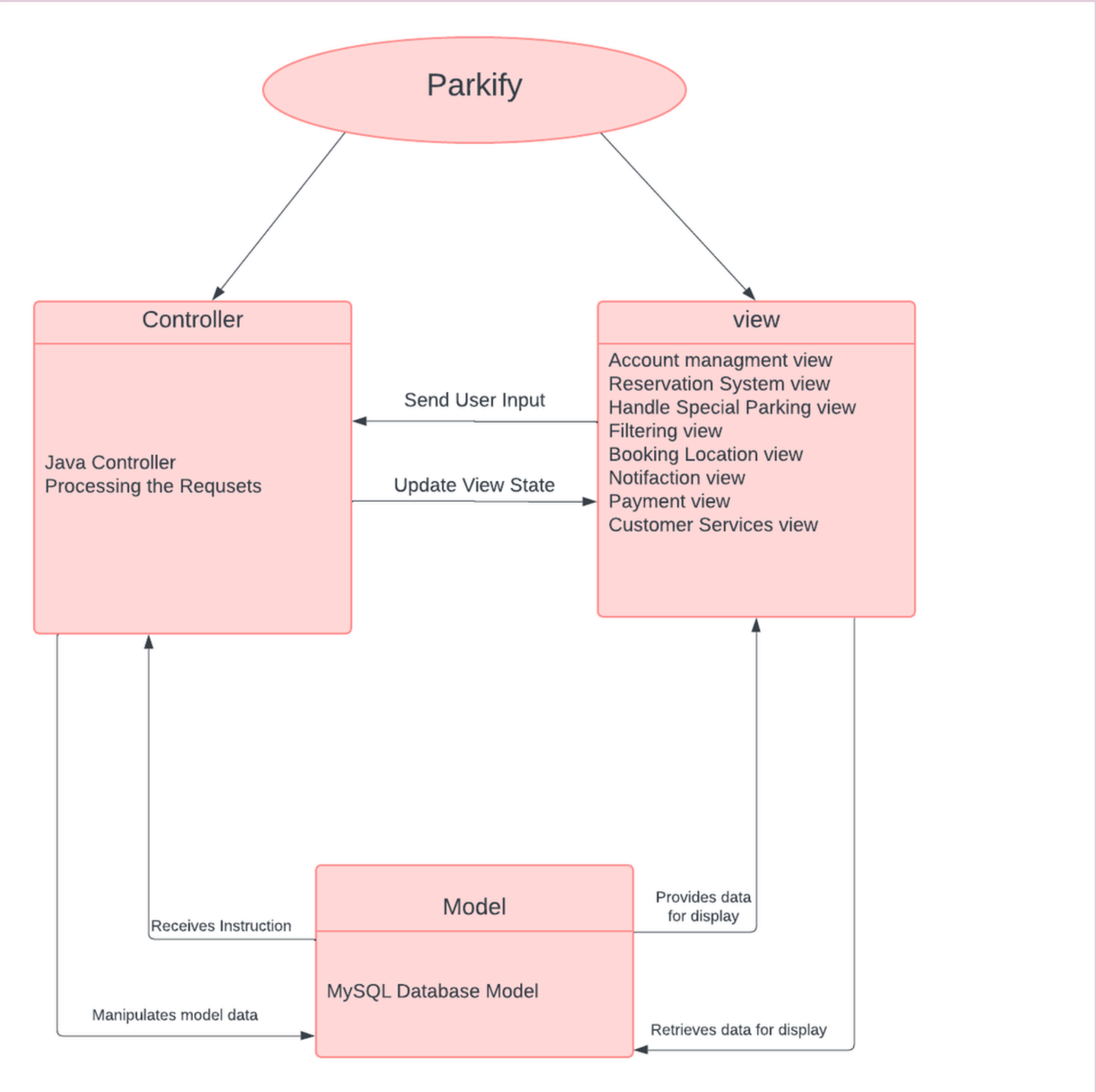

22:00
Parkify
Now
Reminder: Your parking reservation A3 - Makkah Mall will be avaliable in 15min.
Cloudy
H:16° L:10°
Weather
Photos
FaceTime
MON
20
Calendar
Photos
Camera
14
Mail
Notes
Reminders
Clock
News
TV
Podcasts
App Store
Maps
Health
Wallet
Settings
1

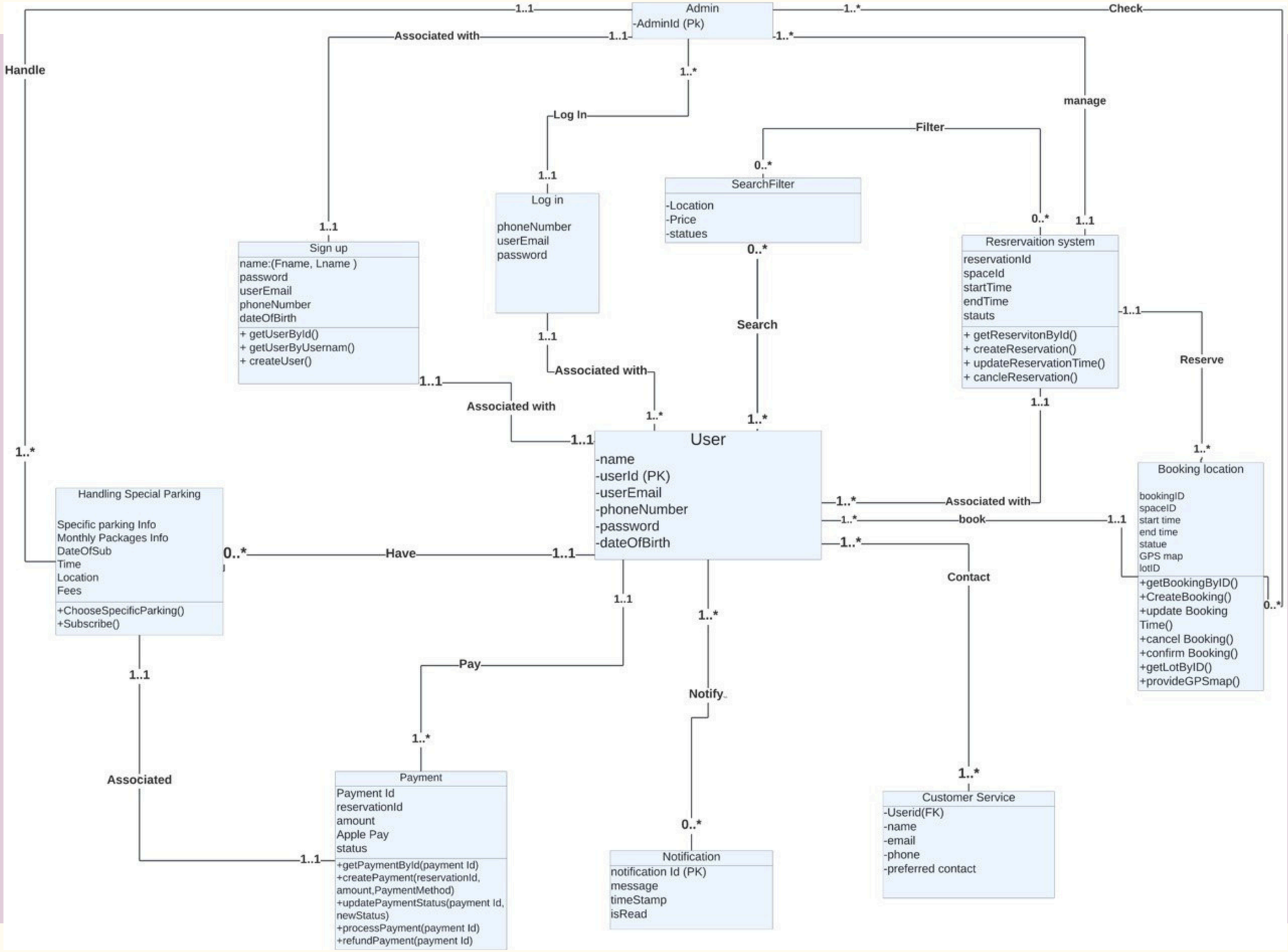

Settings
My subscriptions
Customers Service

Chapter 2 (Design phase)


Booking Location
Parking ID
Reservation System
Reservation confirmation
Payment
Parking Info
Reservation Ability
Subscription confirmation
Notification
Parking Details
User
Special Parking
handle special parking
Creat/Login Account
Filterd Options
Inquiries&Support
Account Managment
Search Filter
Customer Services


<<window>>
Prarkify
Main Page
<<button>>
sign Up
<<button>>
log In
<<window>>
Sign Up
<<button>>
Sign up
<<window>>
Log In
<<button>>
Log in
<<button>>
Rest
password
<<button>>
Sign up
<<window>>
Prarkify
<<button>>
Search
<<button>>
CS
<<button>>
Choose a
Specific Park
<<button>>
Reserve
parking
<<window>>
Customer
Service
<<button>>
Live Chat
<<button>>
Frequent Q&A
<<window>>
Live Chat
<<button>>
Start Chat
<<button>>
End&Close
<<window>>
Handle special
parking
<<button>>
Subscribe to
mothly
packages
<<button>>
Choosre a
Specific Park
<<window>>
SearchFilter
<<button>>
Reset Filter
<<button>>
Apply Filter
<<button>>
Refine Filter
<<window>>
Filterd Result
<<button>>
Select Parking
<<button>>
Back
<<window>>
reservation
<<button>>
select date
<<button>>
select time
<<button>>
submit
<<button>>
Cancel
<<window>>
Booking location
<<button>>
select spot
<<button>>
Confirm
<<button>>
Cancel
<<window>>
payment
<<button>>
Apple pay
<<button>>
Pay with
apple
<<button>>
verfaction


Parkify
Controller
Java Controller
Processing the Requsets
view
Account managment view
Reservation System view
Handle Special Parking view
Filtering view
Booking Location view
Notifaction view
Payment view
Customer Services view
Send User Input
Update View State
Model
MySQL Database Model
Receives Instruction
Manipulates model data
Provides data
for display
Retrieves data for display


Admin
-AdminId (Pk)
Handle
Check
Associated with
Log In
manage
Filter
Log in
phoneNumber
userEmail
password
SearchFilter
-Location
-Price
-statues
Sign up
name:(Fname, Lname)
password
userEmail
phoneNumber
dateOfBirth
+ getUserById()
+ getUserByUsernam()
+ createUser()
Resrervaition system
reservationId
spaceId
startTime
endTime
stauts
+ getReservitonById()
+ createReservation()
+ updateReservationTime()
+ cancleReservation()
Search
Reserve
Associated with
User
-name
-userId (PK)
-userEmail
-phoneNumber
-password
-dateOfBirth
Booking location
bookingID
spaceID
start time
end time
statue
GPS map
lotID
+getBookingByID()
+CreateBooking()
+update Booking Time()
+cancel Booking()
+confirm Booking()
+getLotByID()
+provideGPSmap()
book
Handling Special Parking
Specific parking Info
Monthly Packages Info
DateOfSub
Time
Location
Fees
+ChooseSpecificParking()
+Subscribe()
Have
Contact
Pay
Notify
Associated
Payment
Payment Id
reservationId
amount
Apple Pay
status
+getPaymentById(payment Id)
+createPayment(reservationId, amount,PaymentMethod)
+updatePaymentStatus(payment Id, newStatus)
+processPayment(payment Id)
+refundPayment(payment Id)
Notification
notification Id (PK)
message
timeStamp
isRead
Customer Service
-Userid(FK)
-name
-email
-phone
-preferred contact

Chapter 3 (Implementation)
Databases

Users Database

Users

+ ADD USER

Email	Password	Username	Full Name	Profile Image
aphrodiet.7@gmail.com	[hidden]		Bayan Abdullah	
areen.abdullah007@g...	[hidden]		Areen Abdullah	
babybea57@hotmail.com	[hidden]		Bayan	
ahad@gmail.com	[hidden]		ahad	
manayerbusiness@gm...	[hidden]		manayer alabdali	
s1s@gmail.com	[hidden]		SS	

Nearest gates Database

Nearest Gate

+ ADD NEAREST GATE

	Name	ParkingSpaces	ID	Created	Updated
	1	10 parkingspaces	2	a day ago	a day ago
	2	10 parkingspaces	1	a day ago	a day ago

Nearest Cinema Database

Nearest Cinema

+ ADD NEAREST CINEMA

	Name	ParkingSpaces	ID	Created	Updated
	Vox Cinema	10 parkingspaces	2	a day ago	a day ago
	Muvi Cinema	10 parkingspaces	1	a day ago	a day ago

Payment records Database

Payment record

+ ADD PAYMENT RECORD

Name	Card Number	Expiration Date	CVV	ID
bayan	34577655778	5/18/2028	***	7
Ahad Abuyabs	321651465846531	5/17/2027	***	5
Manayer Alabdali	6565465136545	5/17/2029	***	4

Price types Database

Price Type

+ ADD PRICE TYPE

Name	ParkingSpaces	ID	Created	Updated
Paid	16 parkingspaces	2	a day ago	a day ago
Free	4 parkingspaces	1	a day ago	a day ago

Parking Spaces

ParkingSpaces

+ ADD PARKINGSPACE

Name	Mall	Nearest Cinema	Nearest Gate	Price Type
1R1	RedSea Mall (R)	Vox Cinema	1	Paid
1R2	RedSea Mall (R)	Vox Cinema	1	Paid
1R3	RedSea Mall (R)	Vox Cinema	1	Paid
1R4	RedSea Mall (R)	Vox Cinema	1	Free
1R5	RedSea Mall (R)	Vox Cinema	1	Paid
2R1	RedSea Mall (R)	Vox Cinema	2	Paid

Parking spots are all available

Name	Mall	Availability	Nearest Cinema	Nearest Gate
1R1	RedSea Mall (R)	✓	Vox Cinema	1
1R2	RedSea Mall (R)	✓	Vox Cinema	1
1R3	RedSea Mall (R)	✓	Vox Cinema	1
1R4	RedSea Mall (R)	✓	Vox Cinema	1
1R5	RedSea Mall (R)	✓	Vox Cinema	1
2R1	RedSea Mall (R)	✓	Vox Cinema	2

Parking spaces availability

Availability changed because some parking are reserved

ParkingSpaces

+ ADD PARKINGSPACE

Availability	Reservations	ID	Created	Updated
	1 reservation	80	a day ago	a minute ago
✓		79	a day ago	21 minutes ago
	1 reservation	78	a day ago	a minute ago
✓		77	a day ago	18 minutes ago
	1 reservation	76	a day ago	a minute ago
✓		75	a day ago	17 minutes ago

Reservations:
before canceling (2R3)

Reservation

+ ADD RESERVATION

Name	User	ParkingSpace	ID	Created
2R3	aphrodiet.7@gmail.com	2R3	43	an hour ago
1R3	ahad@gmail.com	1R3	42	3 hours ago
1R1	babybea57@hotmail.com	1R1	41	5 hours ago
1R5	Aurora Cruz@gmai.com	1R5	40	5 hours ago
1A4	ss@gmail.com	1A4	39	5 hours ago
2R4	manayerbusiness@gm...	2R4	36	5 hours ago

The reservations:
after canceling the (2R3) reservation

Reservation

+ ADD RESERVATION

Name	User	ParkingSpace	ID	Created
1R3	ahad@gmail.com	1R3	42	3 hours ago
1R1	babybea57@hotmail.com	1R1	41	5 hours ago
1R5	Aurora Cruz@gmai.com	1R5	40	5 hours ago
1A4	ss@gmail.com	1A4	39	5 hours ago
2R4	manayerbusiness@gm...	2R4	36	5 hours ago
1A5	s1s@gmail.com	1A5	33	5 hours ago

The availability after reservation

ParkingSpaces

+ ADD PARKINGSPACE

Name	Mall	Nearest Cinema	Nearest Gate	Price Type
2R3	RedSea Mall (R)	Vox Cinema	2	Paid

ParkingSpaces

+ ADD PARKINGSPACE

Nearest Cinema	Nearest Gate	Price Type	Availability	Reservations
Vox Cinema	2	Paid		1 reservation

The availability After canceling the reservation:

the parking space become Available

ParkingSpaces

+ ADD PARKINGSPACE

Name	Mall	Nearest Cinema	Nearest Gate	Price Type
2R3	RedSea Mall (R)	Vox Cinema	2	Paid

ParkingSpaces

+ ADD PARKINGSPACE

Availability	Reservations	ID	Created	Updated
✓		73	a day ago	21 minutes ago

OPTIMIZING ACCESS SPEED

To optimize our data model design and enhance data access efficiency, we'll employ clustering techniques aimed at consolidating records frequently accessed together from disparate tables. For instance, when retrieving parking data for specific zones or areas, this approach minimizes redundant storage by combining associated records.

Chapter 4 (Testing)

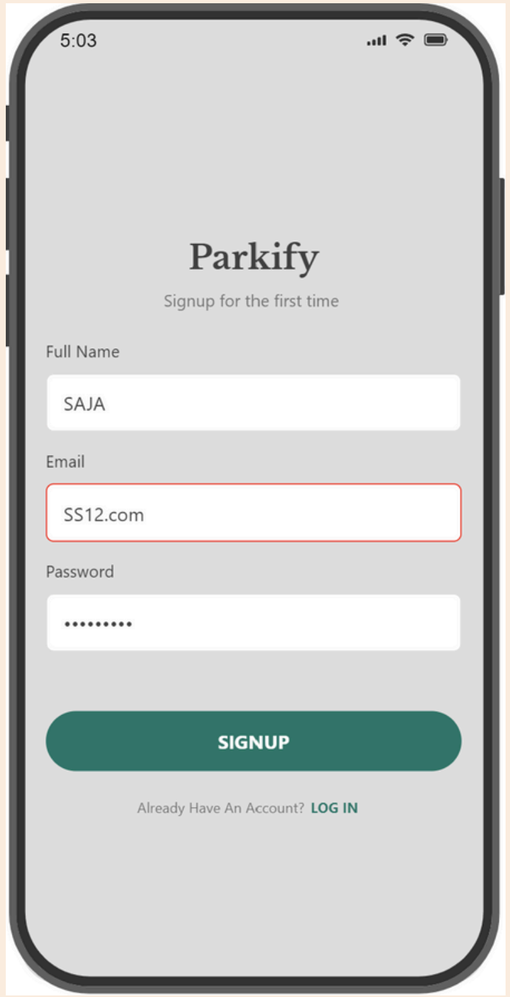
SIGNUP TEST CASE

	Test case	Input	Expected output	Actual output	Remark
1	Test case for entering correct information	Correct information	Main page displayed	Main page displayed	Pass
2	Test case for entering incorrect format	incorrect format (e.g Email)	App doesn’t process to main page	App doesn’t process to main page	Pass
3	Test case for ignoring field	field ignored	Message “This field is required” displayed	Message “This field is required” displayed	Pass

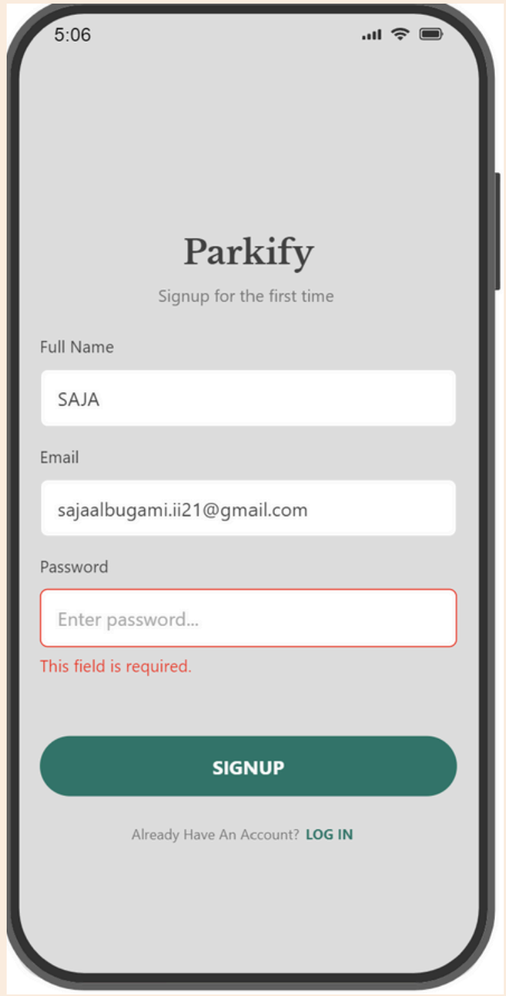
SIGNUP TEST CASES

Test case 1

Test case 2



Test case 3



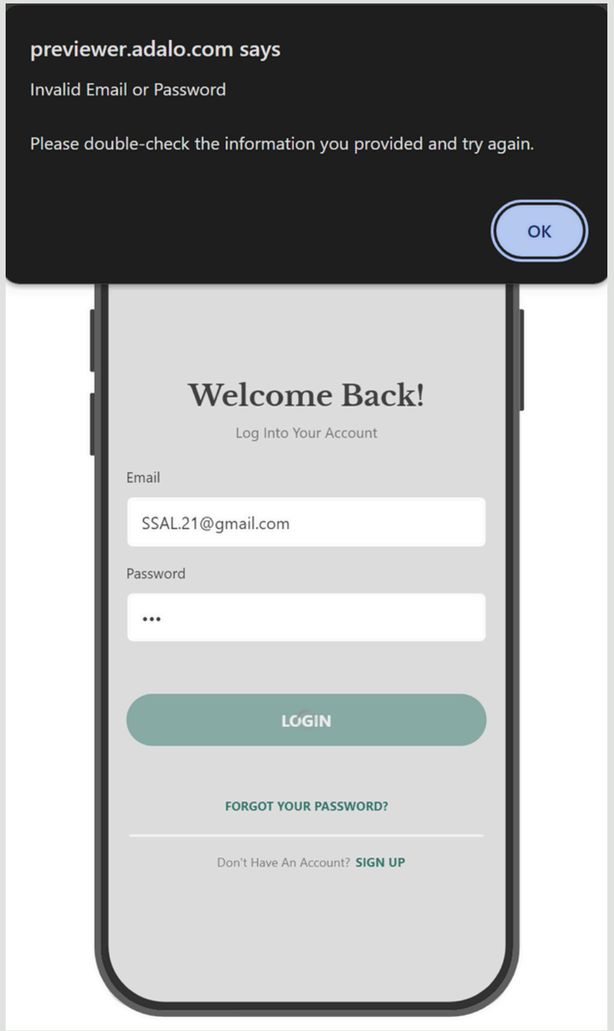
LOGIN TEST CASE

	Test case	Input	Expected output	Actual output	Remark
1	Test case for entering correct information	Correct information	Main page displayed	Main page displayed	Pass
2	Test case for entering incorrect information	Incorrect information	message “Invalid emaill or password” displyed	message “Invalid emaill or password” displyed	Pass
3	Test case for forgetting password	Click “Forgot password”	Password rest page displayed	Password rest page displayed	Pass

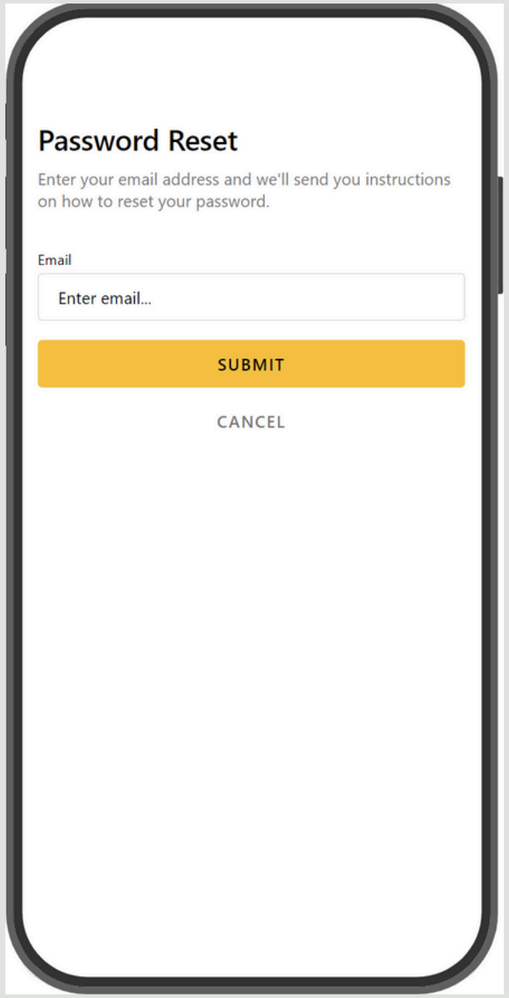
LOGIN TEST CASES

Test case 1

Test case 2



Test case 3

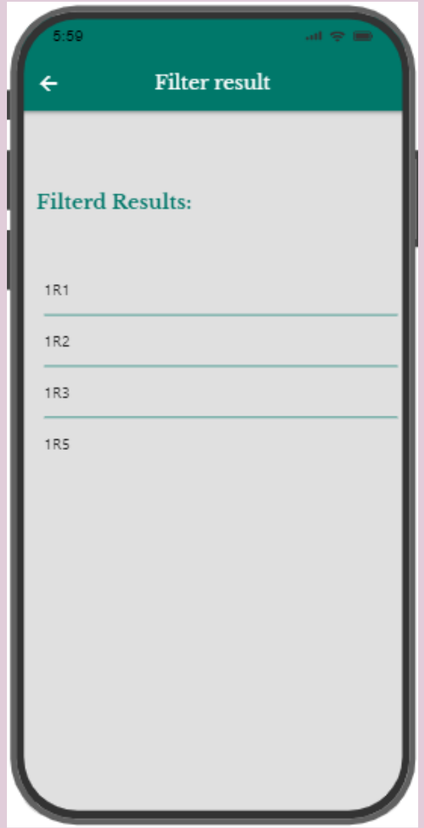


SEARCH&FILTER TEST CASE

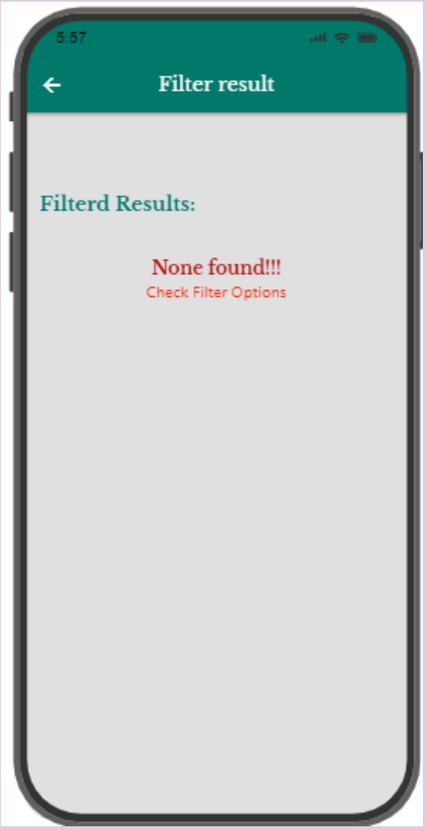
	Test case	Input	Expected output	Actual output	Remark
1	Test Case 1: Correct Filter	correct filter options	Filterd Results: List of parking spaces based on the filter	Filterd Results: List of parking spaces based on the filter	pass
2	Test Case 2: Incorrect Filter	incorrect filter options	Filterd Results: none Found	Filterd Results: none Found!!! check your filter options	pass

SEARCH&FILTER TEST CASES

Test case 1



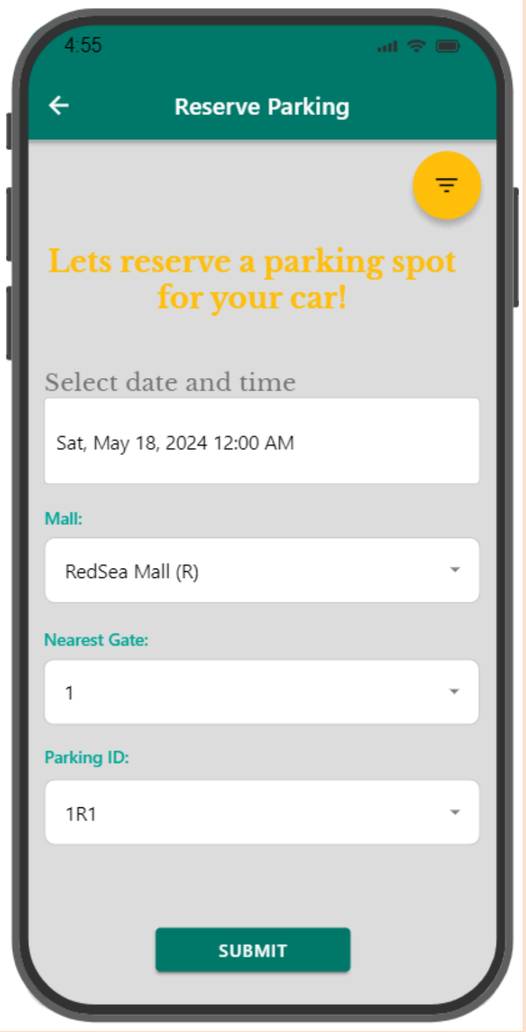
Test case 2

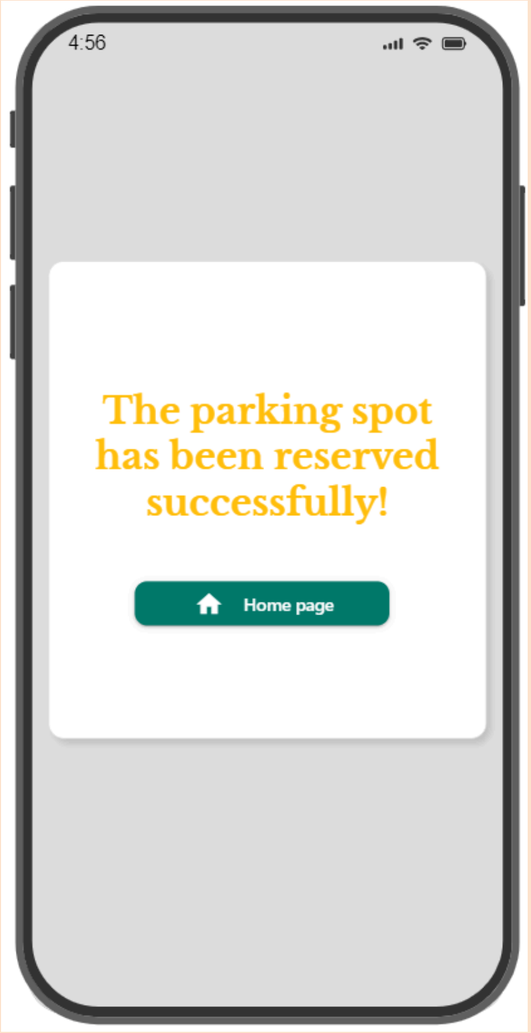


RESERVATION TEST CASE

	Test case	Input	Expected output	Actual output	Remark
1	Test case of data submission and completion of storage in the database	enter submit button	completion of storage in the database	completion of storage in the database	Pass

RESERVATION TEST CASE



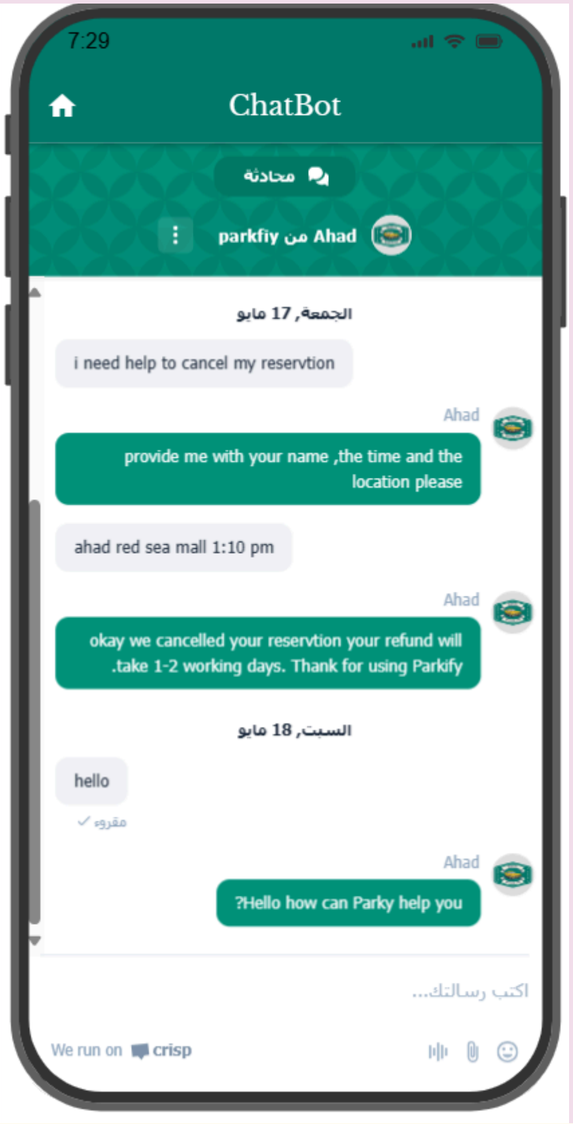


COUSTOMER SERVICES TEST CASE

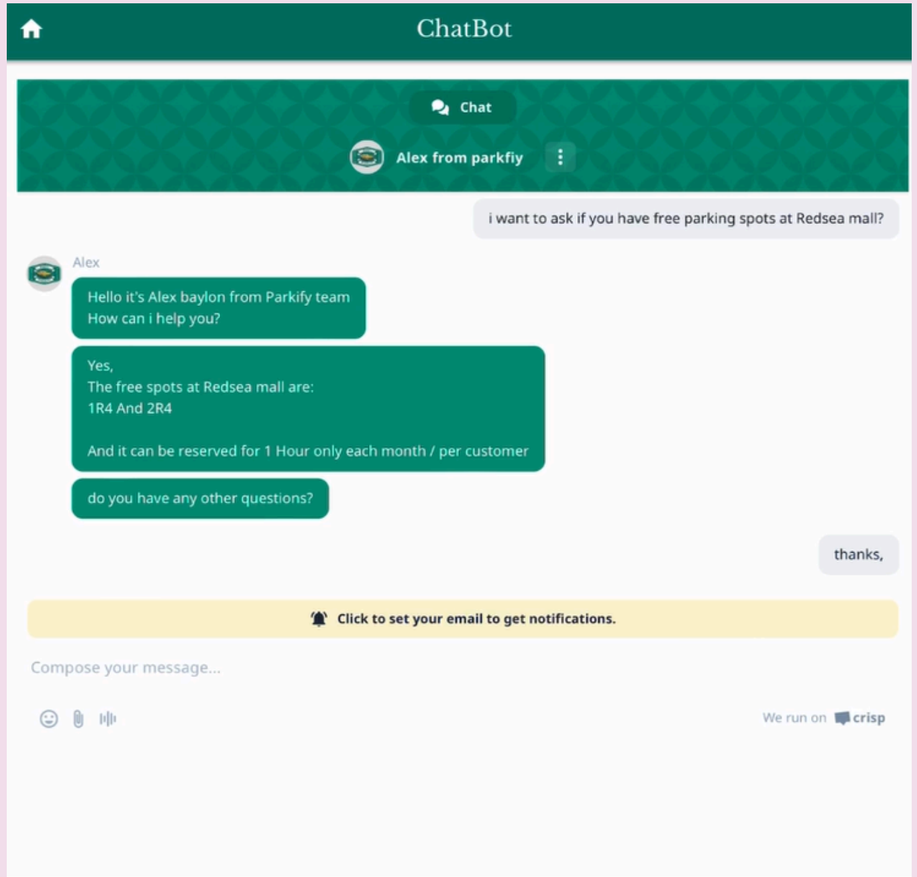
	Test case	Input	Expected output	Actual output	Remark
1	Test case of effective response of the chat bot.	enter a question.	A fast response to the Question asked by the coustomer.	A response in matter of 1 to 60 min Max.	Pass

COUSTOMER SERVICES TEST CASE

Question about Cancellation



Question about Pricing

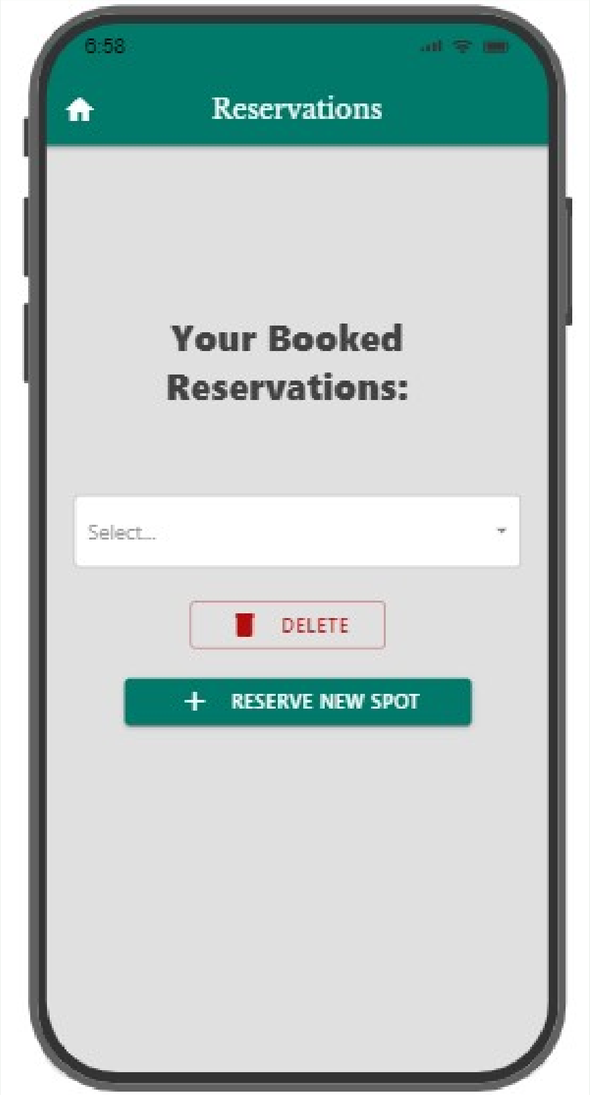


CANCEL RESERVATION TEST CASE

	Test case	Input	Expected output	Actual output	Remark
1	Confirmation: The app should prompt the user for confirmation before canceling the reservation.	enter Confirmation button	Are you sure you want to cancel? Yes or No	unavailable	failure
2	Success message: After successful cancellation, the app should display a confirmation message to the user.	Display a cancellation confirmation message	Your reservation has been successfully cancelled!	!!!!!!!!!!! Your Reservation for (Reservation name) is Canceled !!!!!!!!!!!	Pass

CANCEL RESERVATION TEST CASE

Test case 1



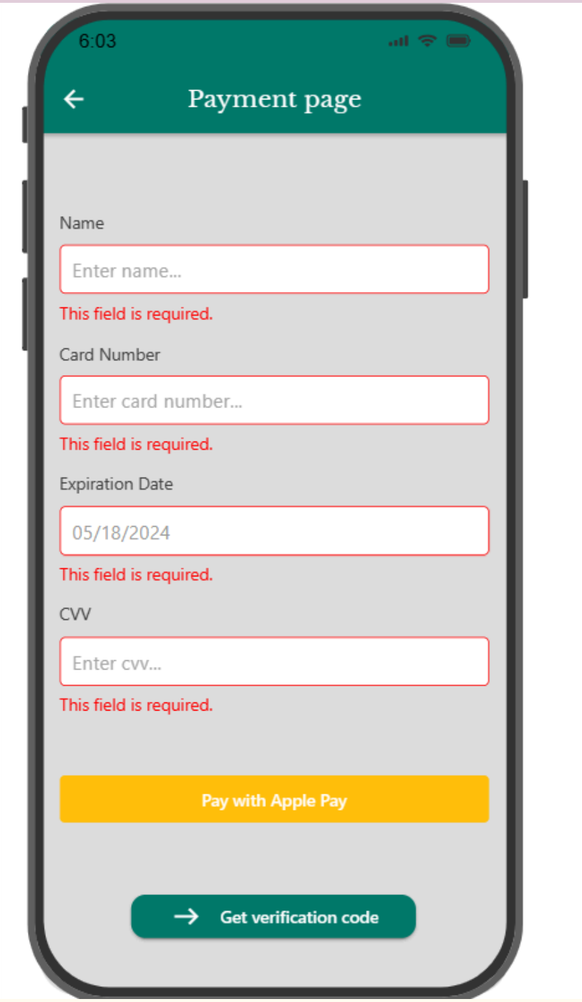
Test case 2

PAYMENT TEST CASE

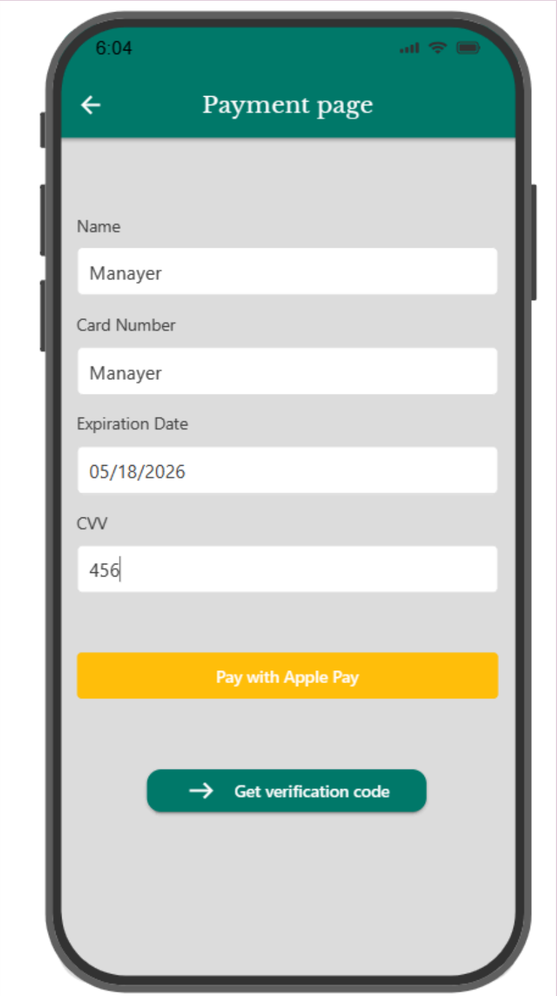
	Test case	Input	Expected output	Actual output	Remark
1	A test case for ignore basic box	ignore basic box	this field is required	this field is required	Pass
2	A test case for enter text in a box that receives numbers only	try to enter text	nothing is received	nothing is received	Pass
3	A test case for enter text in a name box	the name written on the card	the name	the name	Pass

PAYMENT TEST CASES

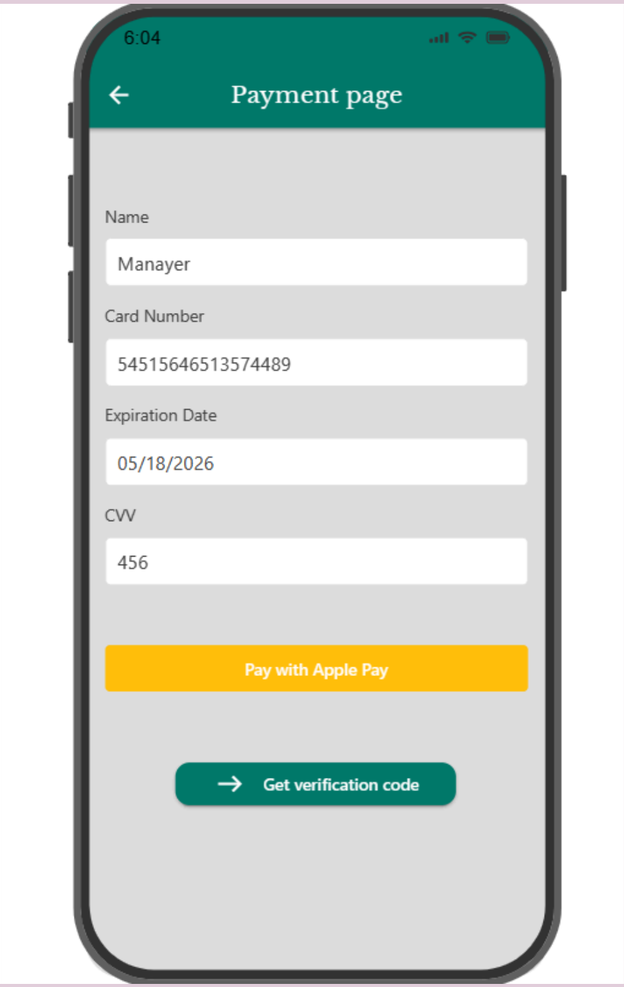
Test case 1



Test case 2



Test case 3



Chapter 5 (Evolution)

SearchFilter Evolution

Search Filter Page

Before Evolution

```html
<!DOCTYPE html>
<html>
<head>
    <title>Parkify - Find Parking</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <h1>Find Parking</h1>

    <form>
        <label for="mall">Mall:</label>
        <select id="mall">
            <option value="mall1">Mall of Arabia</option>
            <option value="mall2">Red Sea Mall</option>
        </select><br><br>

        <label for="price">Price:</label>
        <input type="text" id="price" placeholder="5-10 SAR"><br><br>

        <button type="button" onclick="searchParking()">Search</button>
    </form>
</body>
</html>
```

After Evolution

```html
<!DOCTYPE html>
<html>
<head>
    <title>Parkify - Find Parking</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <h1>Find Parking</h1>

    <form>
        <label for="mall">Mall:</label>
        <select id="mall" onchange="updateGatesAndCinemas()">
            <option value="mall1">Mall of Arabia</option>
            <option value="mall2">Red Sea Mall</option>
        </select><br><br>

        <label for="gate">Gate:</label>
        <select id="gate">
            </select><br><br>

        <label for="cinema">Cinema:</label>
        <select id="cinema">
            </select><br><br>

        <label for="price">Price:</label>
        <input type="text" id="price" placeholder="5-10 SAR"><br><br>

        <button type="button" onclick="searchParking()">Search</button>
    </form>
    <script src="script.js"></script>
</body>
</html>
```

SearchFilter Evolution

Filter Results Page

Before Evolution

```html
<!DOCTYPE html>
<html>
<head>
    <title>Parkify - Parking Results</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <h1>Parking Results</h1>

    <div id="results">
        </div>
</body>
</html>
```

After Evolution

```html
<!DOCTYPE html>
<html>
<head>
    <title>Parkify - Parking Results</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <h1>Parking Results</h1>

    <div id="results">
         </div>

    <div id="map-container">
        </div>

    <script src="script.js"></script> </body>
</html>
```

Reservation Evolution

Before Evolution

```html
<!DOCTYPE html>
<html>
<head>
    <title>ParkPro - Parking Reservation</title>
    <link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
    <div class="container">
        <div class="parkpro-screen">
            <h1>Welcome to ParkPro - Parking Reservation</h1>
            <div class="reservation-form">
                <label for="date">Date:</label>
                <input type="date" id="date">
                <label for="time">Time:</label>
                <input type="time" id="time">
                <label for="location">Location:</label>
                <select id="location">
                    <option value="location1">Location 1</option>
                    <option value="location2">Location 2</option>
                    <option value="location3">Location 3</option>
                </select>
                <button id="reserve-btn">Reserve</button>
                <button id="cancel-btn">Cancel</button>
                <button id="details-btn">View Details</button>
            </div>
        </div>
    </div>
</body>
</html>
```

After Evolution

```html
<!DOCTYPE html>
<html>
<head>
    <title>ParkPro - Parking Reservation</title>
    <link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
    <div class="container">
        <div class="parkpro-screen">
            <h1>Welcome to ParkPro - Parking Reservation</h1>
            <form id="reservation-form">
                <div class="form-group">
                    <label for="location">Location:</label>
                    <select id="location" required>
                        <option value="">Select Location</option>
                        <option value="location1">Location 1</option>
                        <option value="location2">Location 2</option>
                        <option value="location3">Location 3</option>
                    </select>
                </div>
                <div class="form-group">
                    <label for="date">Date:</label>
                    <input type="date" id="date" required>
                </div>
                <div class="form-group">
                    <label for="time">Time:</label>
                    <input type="time" id="time" required>
                </div>
                <div class="btn-group">
                    <button type="submit" id="reserve-btn">Reserve</button>
                    <button type="button" id="cancel-btn">Cancel</button>
                </div>
            </form>
        </div>
    </div>
</body>
</html>
```

Payment Evolution

Before Evolution

```html
<!DOCTYPE html>
<html>
<head>
    <title>Payment Page</title>
    <link rel="stylesheet" href="styles.css">
    <script src="script.js"></script>
</head>
<body>
    <div class="payment-form">
        <h2>Payment Details</h2>
        <form action="submit_payment.php" method="POST">
            <label for="card-number">Card Number:</label>
            <input type="text" id="card-number" name="card-number" required><br><br>
            <label for="expiration-date">Expiration Date:</label>
            <input type="text" id="expiration-date" name="expiration-date" required><br><br>
            <label for="cvv">CVV Code:</label>
            <input type="text" id="cvv" name="cvv" required><br><br>
            <input type="submit" value="Submit Payment">
        </form>
    </div>
</body>
</html>
```

After Evolution

welcome page.html > html

```html
<!DOCTYPE html>
<html>
<head>
    <title>Payment Page</title>
    <style>
        /* Add CSS styles here */
        .payment-form {
            border: 2px solid yellow;
            padding: 20px;
            background: white;
        }
    </style>
</head>
<body>
    <div class="payment-form">
        <h2>Payment Details</h2>
        <form action="submit_payment.php" method="POST">
            <label for="card-number">Card Number:</label>
            <input type="text" id="card-number" name="card-number" required><br><br>
            <label for="expiration-date">Expiration Date:</label>
            <input type="text" id="expiration-date" name="expiration-date" required><br><br>
            <label for="cvv">CVV Code:</label>
            <input type="text" id="cvv" name="cvv" required><br><br>
            <input type="submit" value="Submit Payment">
        </form>
    </div>
</body>
</html>
```

coustomer services

Before Evolution

Evolution

After Evolution

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Customer Services Interface</title>

</head>
<body>
        <h1>Customer Services</h1>
        <p>Welcome to our customer services interface. Please fill out the form below and we will get back to you shortly.</p>
        <form action="#" method="POST">
            <label for="name">Name:</label>
            <input type="text" id="name" name="name" required><br><br>
            <label for="email">Email:</label>
            <input type="email" id="email" name="email" required><br><br>
            <label for="message">Message:</label><br>
            <textarea id="message" name="message" rows="4" required></textarea><br><br>
            <button type="submit">Submit</button>
        </form>
</body>
</html>
```

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Customer Services Interface</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #ffe6e6; /* Pink */
            margin: 0;
            padding: 0;
        }
        .container {
            max-width: 800px;
            margin: 20px auto;
            background-color: #ffffcc; /* Yellow */
            padding: 20px;
            border-radius: 8px;
            box-shadow: 0 0 20px rgba(0,0,0,0.1);
            display: flex;
            flex-direction: column;
            align-items: center;
        }
        h1 {
            color: #ff4d4d; /* Red */
            margin-bottom: 20px;
        }
        p {
            color: #666;
            margin-bottom: 20px;
        }
        form {
            width: 100%;
            max-width: 500px;
        }
        label {
            display: block;
            margin-bottom: 10px;
        }
        input, textarea {
            width: 100%;
            padding: 10px;
            margin-bottom: 20px;
            border: 1px solid #ccc;
            border-radius: 5px;
            box-sizing: border-box;
        }
        button {
            padding: 10px 20px;
            background-color: #ff3333; /* Red */
            color: #fff;
            border: none;
            border-radius: 5px;
            cursor: pointer;
        }
```

```html
            color: #fff;
            border: none;
            border-radius: 5px;
            cursor: pointer;
        }
        button:hover {
            background-color: #cc0000; /* Darker Red */
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>Customer Services</h1>
        <p>We're here to assist you! Please fill out the form below and we'll get back to you as soon as possible.</p>
        <form action="#" method="POST">
            <label for="name">Name:</label>
            <input type="text" id="name" name="name" required><br>
            <label for="email">Email:</label>
            <input type="email" id="email" name="email" required><br>
            <label for="message">Message:</label><br>
            <textarea id="message" name="message" rows="4" required></textarea><br>
            <button type="submit">Submit</button>
        </form>
    </div>
</body>
```

Cancellation of reservation evolution

Before Evolution

```html
<!DOCTYPE html>
<html>
<head>
  <title>Reservations</title>
</head>
<body>
  <h1>Reservations</h1>
  <p>Your Booked Reservations:</p>
  <select id="reservationsList">
    <option value="">Select...</option>
  </select>
  <button id="deleteButton">DELETE</button>
  <button id="addReservationButton">+ RESERVE NEW SPOT</button>

  <div id="successMessage" style="display: none;">
    Reservation deleted successfully!
  </div>
</body>
<script>
  const reservationList = document.getElementById('reservationsList');
  const deleteButton = document.getElementById('deleteButton');
  const successMessage = document.getElementById('successMessage');

  // Simulate some initial reservations (replace with actual data fetching)
  reservationList.add(new Option('John Doe - 2024-05-20 10:00 AM', 'reservation1'));
  reservationList.add(new Option('Jane Smith - 2024-05-22 2:00 PM', 'reservation2'))
```

Cancellation of reservation evolution

Before Evolution

```
reservationList.add(new Option('Jane Smith - 2024-05-22 2:00 PM', 'reservation2'));

// Functionality to handle deleting a reservation
deleteButton.addEventListener('click', () => {
  const selectedReservation = reservationList.value;

  // Check if a reservation is selected
  if (selectedReservation) {
    reservationList.remove(reservationList.selectedIndex);

    // Show success message and hide after a short time
    successMessage.style.display = 'block';
    setTimeout(() => {
      successMessage.style.display = 'none';
    }, 2000); // Hide after 2 seconds (adjust as needed)
  } else {
    alert('Please select a reservation to delete.');
  }
});

// Add functionality to handle adding new reservations (replace with form handling)
// ... (your existing code to handle adding reservations)
</script>
</html>
```

Cancellation of reservation evolution

After Evolution

```html
<!DOCTYPE html>
<html>
<head>
  <title>Reservations</title>
  <style>
    .reservation {
      border: 1px solid #ddd;
      padding: 10px;
      margin-bottom: 10px;
      display: flex; /* Allow buttons to sit next to details */
      justify-content: space-between; /* Align buttons to the right */
    }
  </style>
</head>
<body>
  <h1>Reservations</h1>
  <div id="reservations">
  </div>
  <button id="addReservation">Add New Reservation</button>
</body>
<script>
  const reservationList = document.getElementById('reservations');
  const addReservationButton = document.getElementById('addReservation');

  // Function to display a reservation with details and delete button
  function displayReservation(name, date, time) {
```

Cancellation of reservation evolution

After Evolution

```
  reservation.innerHTML = `<b>Name:</b> ${name} - <b>Date:</b> ${date} - <b>Time:</b> ${time}`

  // Create and add the delete button
  const deleteButton = document.createElement('button');
  deleteButton.textContent = "Delete";
  deleteButton.addEventListener('click', () => {
    // Confirmation prompt before deletion
    if (confirm("Are you sure you want to delete this reservation?")) {
      reservationList.removeChild(reservation);
      // Success message after deletion
      alert("Reservation deleted successfully!");
    }
  });
  reservation.appendChild(deleteButton);

  reservationList.appendChild(reservation);
}

// Simulate some initial reservations (replace with actual data fetching)
displayReservation('John Doe', '2024-05-20', '10:00 AM');
displayReservation('Jane Smith', '2024-05-22', '2:00 PM');

// Add functionality to handle adding new reservations (replace with form handling)
addReservationButton.addEventListener('click', () => {
  // Logic to prompt user for details and call displayReservation
});
```

Signup Evolution

Before Evolution

```html
<!DOCTYPE html>
<html>

<head>
    <title>Parkify - Sign Up</title>
</head>

<body>
    <h1>Sign Up for Parkify</h1>
    <form>
        <label for="name">Name:</label>
        <input type="text" id="name" name="name" required>
        <br><br>
        <label for="email">Email:</label>
        <input type="email" id="email" name="email" required>
        <br><br>
        <label for="password">Password:</label>
        <input type="password" id="password" name="password" required>
        <br><br>
        <input type="submit" value="Sign Up">
    </form>
    <p>Already have an account? <a href="Login.html">Log in</a></p>
</body>

</html>
```

After Evolution

```html
<!DOCTYPE html>
<html>

<head>
    <title>Parkify - Sign Up</title>
</head>

<body>
    <h1>Sign Up for Parkify</h1>
    <form action="login.php" method="POST">
        <label for="name">Name:</label>
        <input type="text" id="name" name="name" required>
        <br><br>
        <label for="email">Email:</label>
        <input type="email" id="email" name="email" required>
        <br><br>
        <label for="password">Password:</label>
        <input type="password" id="password" name="password" required
            pattern="^(?=.*\d)(?=.*[a-z])(?=.*[A-Z])(?=.*[!@#$%^&*]).{8,}$"
            title="Password must contain at least 8 characters,
            including at least one lowercase letter, one uppercase letter, one number,
            and one special character (!@#$%^&*).">
        <br><br>
        <input type="submit" value="Sign Up">
    </form>
    <p>Already have an account? <a href="login.html">Log in</a></p>
</body>

</html>
```

Login Evolution

```html
<!DOCTYPE html>
<html>
<head>
    <title>Parkify - Login</title>
</head>
<body>
    <h1>Welcome Back to Parkify</h1>
    <form action="login.php" method="POST">
        <label for="email">Email:</label>
        <input type="email" id="email" name="email" required>
        <br><br>
        <label for="password">Password:</label>
        <input type="password" id="password" name="password" required>
        <br><br>
        <input type="submit" value="Log In">
    </form>
    <p>Forgot your password? <a href="reset.html">Reset it</a></p>
    <p>Don't have an account? <a href="Signup.html">Sign up</a></p>
</body>
</html>
```

the following code doesn’t need evoluation because:

- Structure: The HTML code follows the correct structure, including the necessary HTML tags.
- Form Validation: The form includes the "required" attribute for each input field, ensuring that the user must fill in both the email and password fields before submitting the form.
- Accessibility: The code includes associated <label> elements for the input fields, which is good for accessibility purposes. The "for" attribute in each <label> element correctly associates it with the corresponding input field using the "id" attribute.