

CIS-476 Design Patterns Term Project Presentation

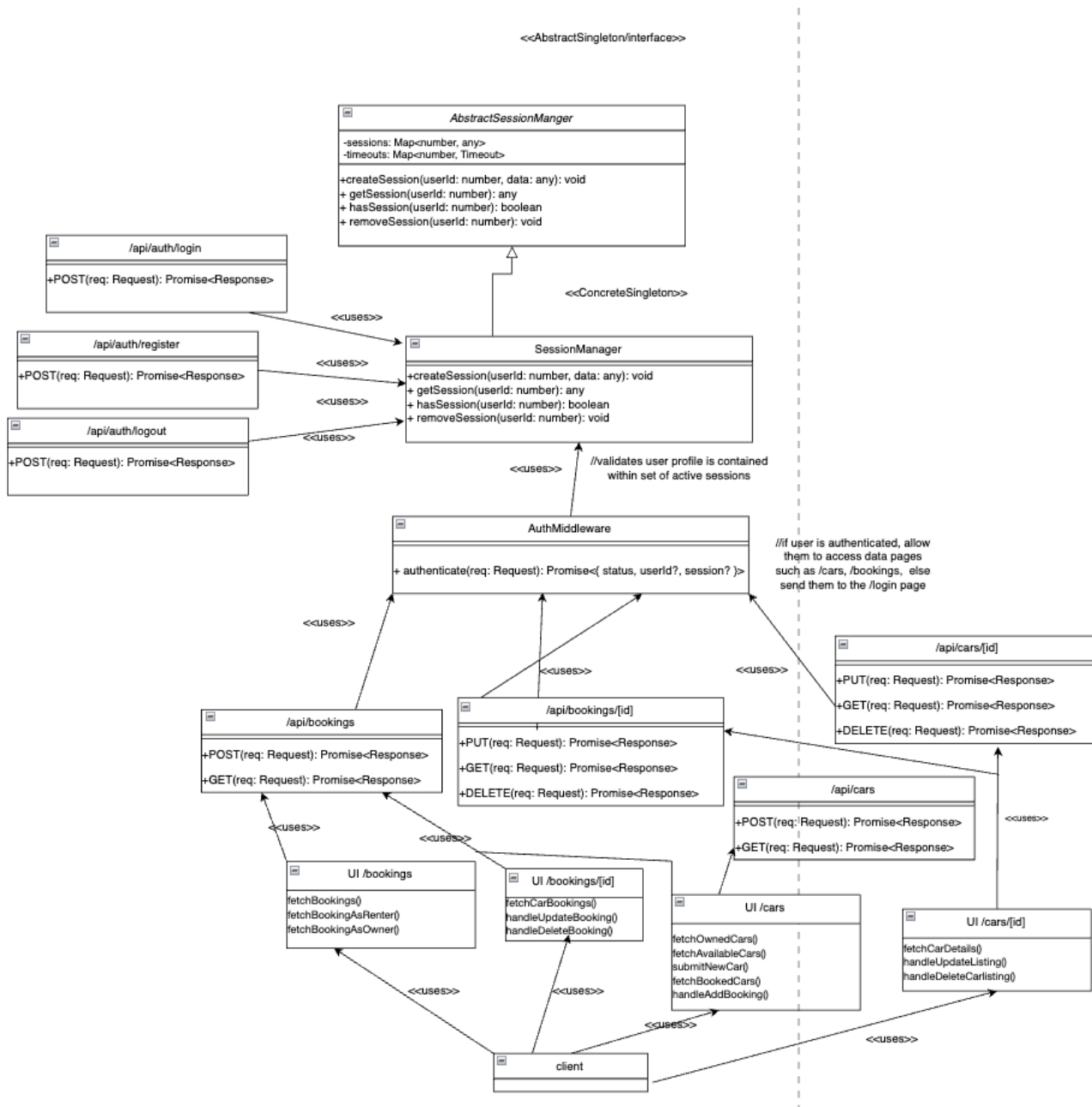
By Joseph Cornell

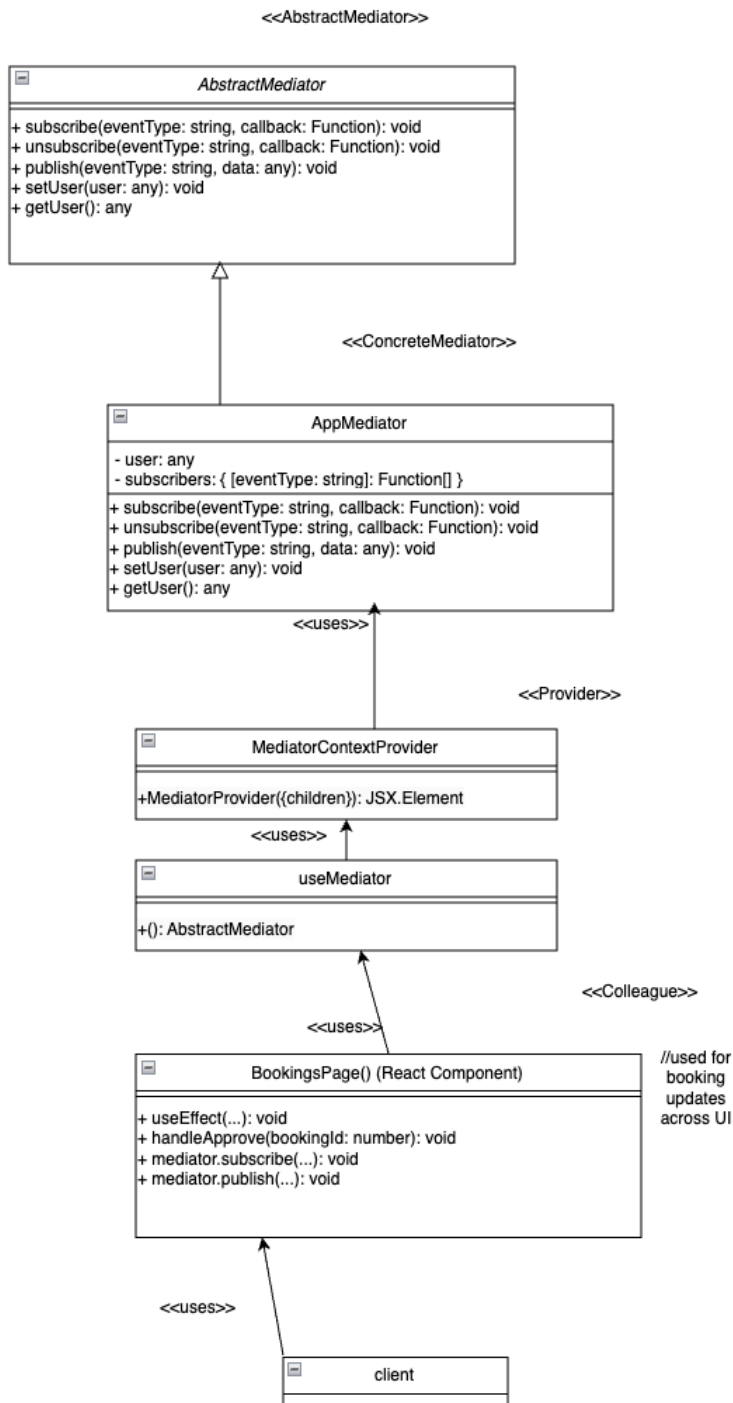
Table of Contents

- 1. Pg3-9 UML Class Diagrams and Descriptions**
- 2. Pg12-14 Database Schema**
- 3. UI Screenshots**

Singleton session manager:

Current user sessions are stored within -sessions, sessions are created upon login after verifying they have an account in the database, sessions are deleted upon logout and timeout ~30 minutes, within register, users are able to create a user account within the sqlite3 db, passwords are encrypted within db using aargon2 node js library, sessions are validated using JWT upon login, then using the userId within request headers/local cookie to to persist the users session throughout their usage.





Mediator

Pattern UI

component

harmonization

Different UI

components use

the mediator

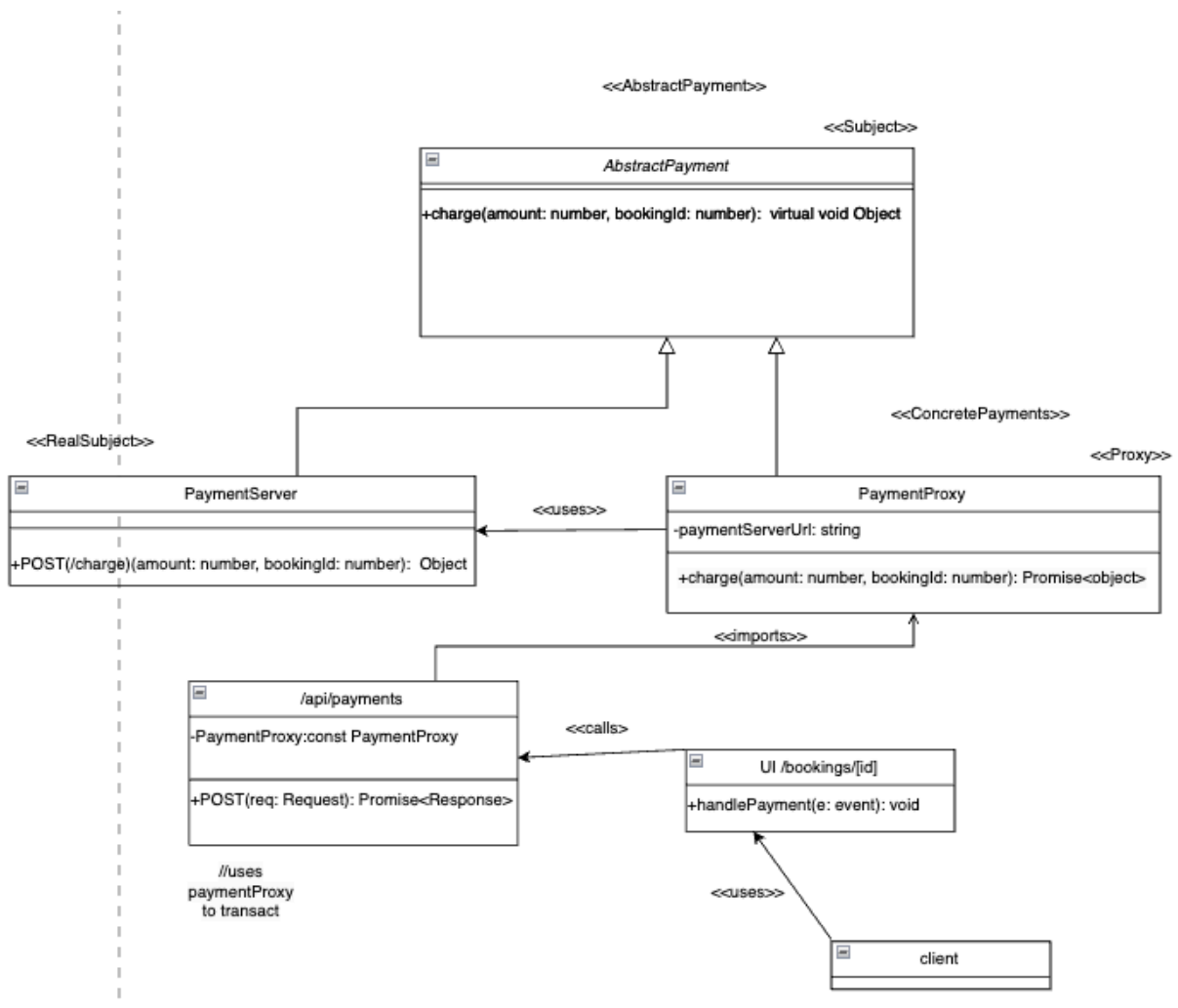
provider to receive

relevant updates

in regards to the

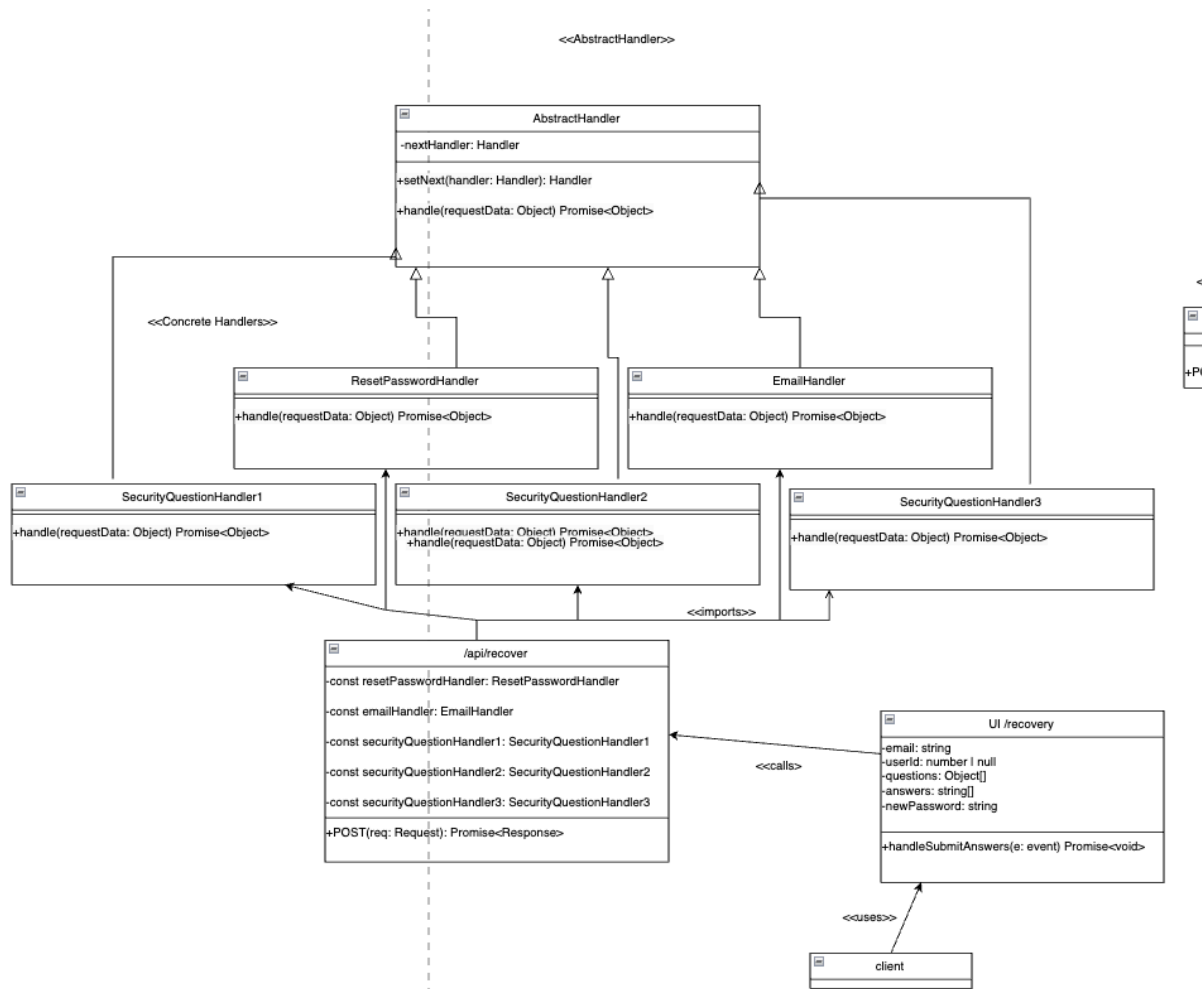
shared state data

across the app



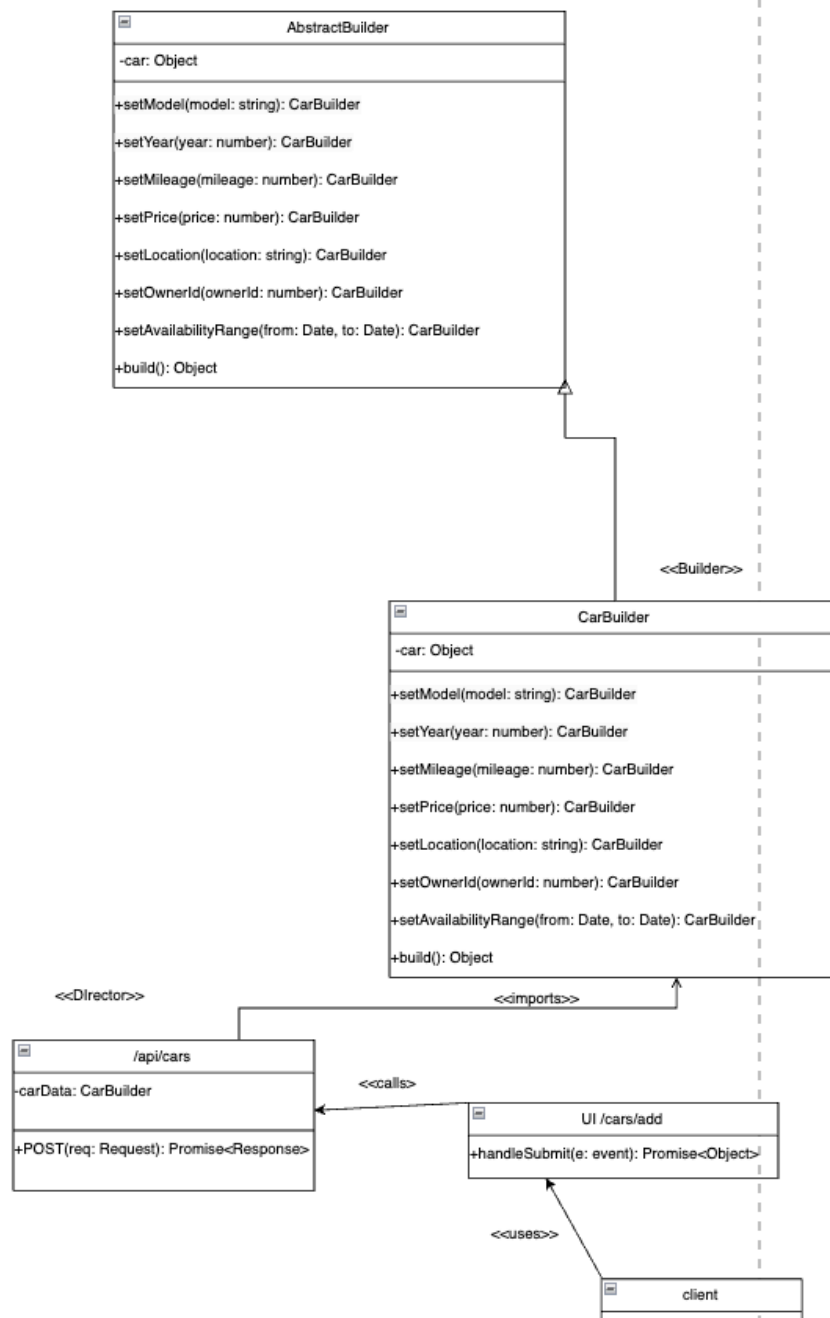
Payment Proxy:

Payment button in UI sends POST request within payment api, within the api, the payment server is called, receives the payment request and payment is confirmed



Chain of Responsibility for account recovery:

Handler methods receive the requests, determine whether they can be passed to the next stop or not based valid input, if valid, they are passed to the next handler, with the resetPassword handler being the last in the link, if valid the password for the account is reset else the request is dealt with gracefully.



Builder Pattern for car listing creation

Different components of the car object are built incrementally to allow variation in build creation and allow us to add these different elements 1 by 1.

Database Schema

```
model User {
  id      Int      @id @default(autoincrement())
  email   String   @unique
  password String
  username String   @unique
  cars    Car[]
  bookings Booking[]
  questions SecurityQuestion[]
  sentMessages Message[] @relation("SentMessages")
  receivedMessages Message[] @relation("ReceivedMessages")
  notifications Notification[]
  bookingObservers BookingObserver[]
}
```

```
model SecurityQuestion {
  id Int @id @default(autoincrement())
  userId Int
  prompt String
  answer String
  user User @relation(fields: [userId], references: [id])
}
```

```
model Car {
  id      Int      @id @default(autoincrement())
  model   String
  year    Int
  mileage Int
  location String
  available Boolean @default(true)
  ownerId Int
  pricePerHour Float
  availableFrom DateTime
  availableTo DateTime
  owner User @relation(fields: [ownerId], references: [id])
  bookings Booking[]
}
```

```
model Booking {
  id      Int      @id @default(autoincrement())
  carId   Int
  renterId Int
  startDate DateTime
  endDate DateTime
  status  String   @default("pending")
}
```

```

cost      Float?
paymentStatus String?

car      Car      @relation(fields: [carId], references: [id])
renter   User     @relation(fields: [renterId], references: [id])
observers BookingObserver[]
}

model Message {
  id      Int      @id @default(autoincrement())
  senderId Int
  receiverId Int
  content String
  timestamp DateTime @default(now())
  sender   User    @relation("SentMessages", fields: [senderId], references: [id])
  receiver User    @relation("ReceivedMessages", fields: [receiverId], references: [id])
}

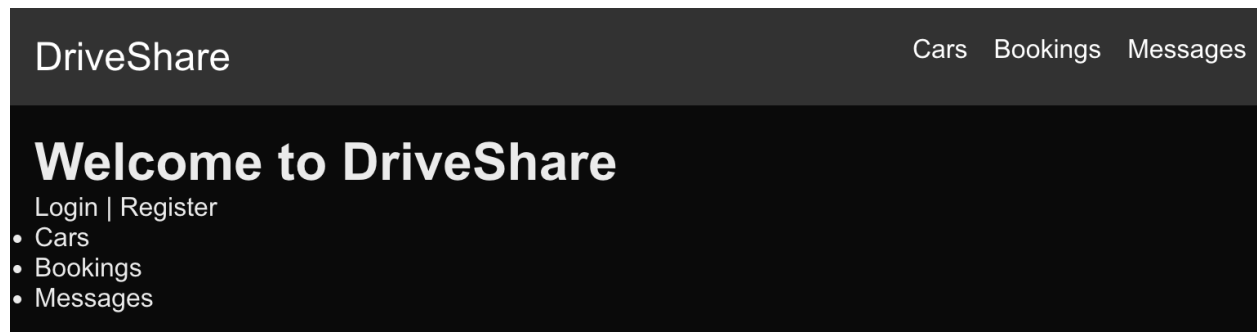
model Notification {
  id      Int      @id @default(autoincrement())
  userId  Int
  content String
  createdAt DateTime @default(now())
  user    User     @relation(fields: [userId], references: [id])
}

model BookingObserver {
  id      Int      @id @default(autoincrement())
  bookingId Int
  userId  Int
  role    String
  createdAt DateTime @default(now())
  booking Booking @relation(fields: [bookingId], references: [id])
  user    User     @relation(fields: [userId], references: [id])
}

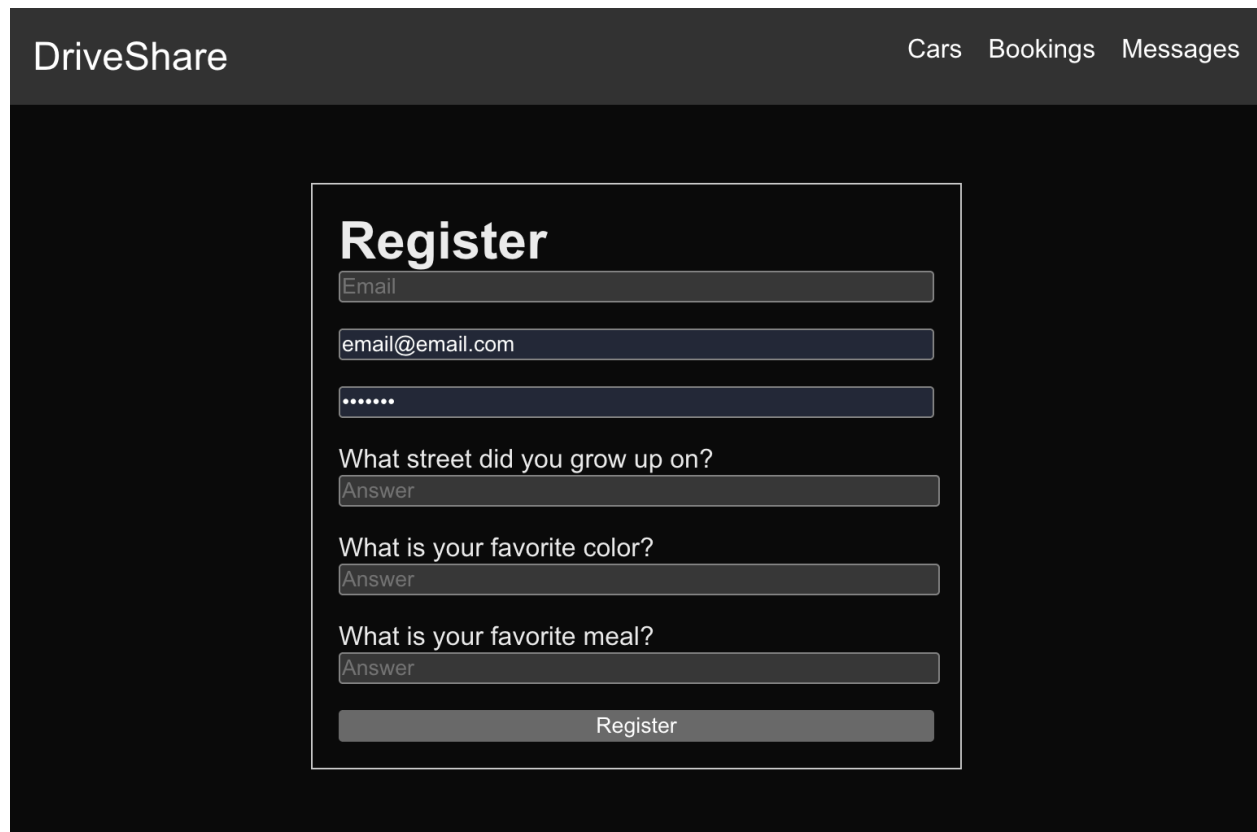
```

UI ScreenShots:

Home Page



Register Page



The image shows the 'Register' page. It has the same dark header as the welcome page, with 'DriveShare' and navigation links 'Cars', 'Bookings', and 'Messages'. The main content area is dark and contains a white-bordered box with the title 'Register' in bold. Inside the box, there are several input fields: an 'Email' field with the placeholder 'email@email.com', a password field with a masked input '.....', and three text input fields with prompts 'What street did you grow up on?', 'What is your favorite color?', and 'What is your favorite meal?'. Each of these three fields has a placeholder 'Answer'. At the bottom of the box is a 'Register' button.

Login page

DriveShare

CarsBookingsMessages

Login

[Need to register?](#)[Forgot password?](#)

Cars Page

DriveShare

CarsBookingsMessages

Cars

My Owned Cars

Ford F-150 - 2020 - Dallas - \$15/hr

My Booked Cars

Booking #1

Car ID: 1

From: 4/9/2025, 6:00:00 PM

To: 4/9/2025, 7:00:00 PM

Status: pending

Available Cars for Rent

Tesla Model 3 - 2021 - Houston - \$25/hr

Search Available Cars

Start Date:

End Date:

Brand:

Search Available Cars

Start Date:

mm/dd/yyyy, --:-- --



End Date:

mm/dd/yyyy, --:-- --



Brand:

Enter brand

Model:

Enter model

Year:

Enter year

Max Mileage:

Enter maximum mileage

Max Price Per Hour:

Enter maximum price per hour

Location:

Enter location

Search



`cars/[id]`

Car Details (#2)

Car Information

Model: Ford F-150

Year: 2020

Mileage: 120000

Location: Dallas

Price per Hour: \$15

Availability: 4/10/2025, 10:00:00 AM to 4/15/2025, 10:00:00 PM

Current Bookings for this Car

No bookings for this car.

Edit Car Listing

Model:

Year:

Mileage:

Location:

Price per Hour:

Available From:

Available To:

Update Listing

Delete Listing

bookings page

My Bookings & Notifications

Your Bookings (as Renter)

- **Booking #1** for Car #1 from 4/9/2025, 6:00:00 PM to 4/9/2025, 7:00:00 PM - Status: pending

[View Details](#)

Bookings for Your Cars (as Owner)

No bookings found for your cars.

Notifications

- Your booking (ID: 1) for car (ID: 1) has an update: "bookingCreated". - 4/9/2025, 10:42:01 PM

bookings/[id]

Booking Details (#1)

Booking Information

Booking ID: 1
Car ID: 1
Renter ID: 2
Cost: 25
Start Date: 4/9/2025, 6:00:00 PM
End Date: 4/9/2025, 7:00:00 PM
Status: pending
Payment Status: unpaid

Car Information

Model: Tesla Model 3
Year: 2021
Mileage: 100000
Location: Houston
Price per Hour: \$25
Availability: 4/9/2025, 6:00:00 PM to 4/9/2025, 9:00:00 PM

[View Car Listing](#)

Edit Booking Details

New Start Date: 04/09/2025, 10:00 PM

New End Date: 04/09/2025, 11:00 PM

[Update Booking](#)

Renter Actions

[Pay Now](#)

Book this Car

Start Date: mm/dd/yyyy, --:-- --

End Date: mm/dd/yyyy, --:-- --

[Book this Car](#)



As owner

Booking Details (#1)


Booking Information


Booking ID: 1
Car ID: 1
Renter ID: 2
Cost: 25
Start Date: 4/9/2025, 6:00:00 PM
End Date: 4/9/2025, 7:00:00 PM
Status: pending
Payment Status: unpaid

Car Information

Model: Tesla Model 3
Year: 2021
Mileage: 100000
Location: Houston
Price per Hour: \$25
Availability: 4/9/2025, 6:00:00 PM to 4/9/2025, 9:00:00 PM
[View Car Listing](#)

Edit Booking Details

New Start Date: 

New End Date: 

Owner Actions

messages /inbox

Inbox & Contacts

Active Conversations

- **User 3**
Hello
4/9/2025, 11:20:38 PM

Owners for Your Bookings

- User 3 +

Bookees of Your Cars

No bookees found.

Messages /chat

Chat with User 3

You: Hello

Type your message
here...

Send