PROJECT REPORT

CAR RENTAL APPLICATION

420-DA4-AS C9_ANDROID MOBILE DEV._ONLINE

GROUP MEMBERS

Thai Duy Roan

Huynh Tu Anh Chau

Ingride Neslie Youadeu Noumbibou

Guided by Professor Pargol Poshtareh

04 December 2024

Overview

The Car Rental Management App is designed to transform the car rental industry by streamlining operations and improving customer satisfaction. This centralized platform efficiently manages a few aspects of car rental services, from car inventory management to booking tracking, payment processing, and report generation. The app optimizes rental workflows and provides real-time insights for better decision-making. The primary purpose of developing this app is to create a seamless, user-friendly experience for both customers and rental agencies. It provides personalized services that cater to the evolving needs of users.

Key features:

Customer's Side

1. **User Authentication**
    - **Login**: Securely access your account using Google Email or Email and password based authentication(Firebase SDK Authentication)
    - **Register**: Create a new account using Password-Based Account (Firebase SDK Authentication)
2. **Car Search and View Available Cars**
    - **Search Cars**: Easily find the perfect car based on your search for a car name and any other available details.
    - **Browse Information**: Explore detailed information about cars, including:
        - Brand
        - Model
        - Pricing
        - Availability
        - Location
        - Seats
        - Car description
3. **Car Rental Process**

- **Rent a Car**: Select your desired car for rental.
- **Set Dates**: Choose valid start and end dates for your rental period.

4. **Payment Options**
    1. **Flexible Payment Methods**: Select your preferred payment option (e.g., card, Klarna).
    2. **Make Payments**: Complete your transaction and see the total fee.
    3. **Payment Information format**:
        - Choose the Visa card and enter a valid Card Number as an example: 4242 4242 4242 4242 .
        - Enter the month and year in future
        - Enter the card CVC which has to be three numbers
        - Enter the Zip Code and have to meet the Canadian format. An example of a valid Canadian format: H7Y8U9.
        - Then you can click on the pay button.
5. **Post-Rental Management**
    - **View Contracts**: Navigated to your rental contract(s) anytime after booking.
6. **Account Management**
    - **Account Details**: View your personal information which are: name, gmail, Date of account creation and Set your avatar.
    - **Logout**: Securely exit your account.

Admin's Side

1. **User Authentication**
    - **Login**: Securely access your account using Google Email or Email and password based authentication(Firebase SDK Authentication)
2. **Car Management**
    - **Add New Cars**:
        - Enter details such as car model, brand, seating capacity, location, price, and upload photos(optional).

- **View All Cars**:
  - Browse the complete list of cars available in the system.
- **Edit Car Details**:
  - Update information like pricing, brand, model, availability, location, image or car's description.
- **Update Car Detail**:
  - Modify existing records to keep car details accurate and up-to-date.

3. **Contract Oversight**
   - **View Customer's Contracts**:
     - Access and edit the status of rental contracts associated with user bookings. From completed to canceled and active.
4. **Account Management**
   - **Account Control Functions**: View all users and block a user.
   - **Logout**: Securely exit your account.

Technologies Used

1. **Frontend**
   - **Android Studio Koala (2024.1.1 Patch 2)**: IDE used for developing the Android App.
   - **XML 1.0**: For designing the user interface (UI) components.
   - **Java 8**: Processes the business logic and user interactions.
   - **Gradle**: Build automation tool for managing dependencies.
2. **Backend**
   - **Authentication:** Integrating **email/password** based Firebase SDK authentication and **Google Sign-in**, into the Car Rental Management App provides a secure and user-friendly way for customers to log in.

- **Secure Session Management**

  Firebase automatically manages session states, enabling customers to stay logged in across sessions without compromising security.

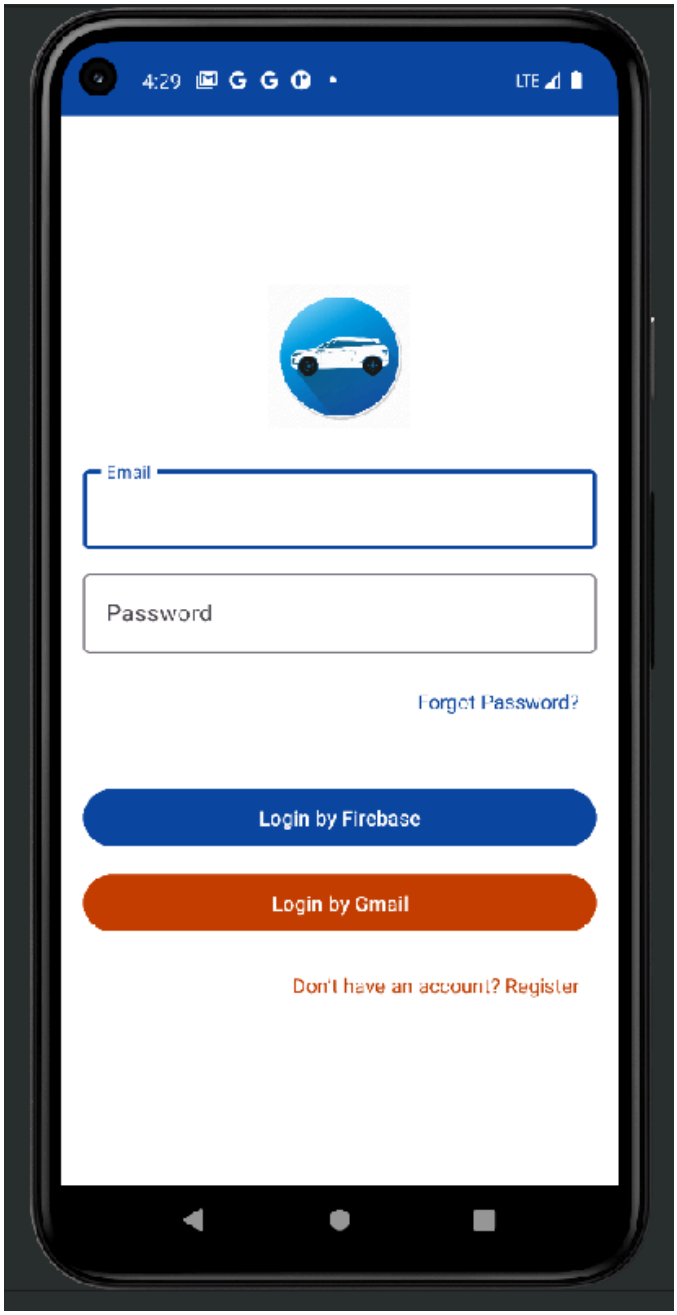
- **Firebase Storage**

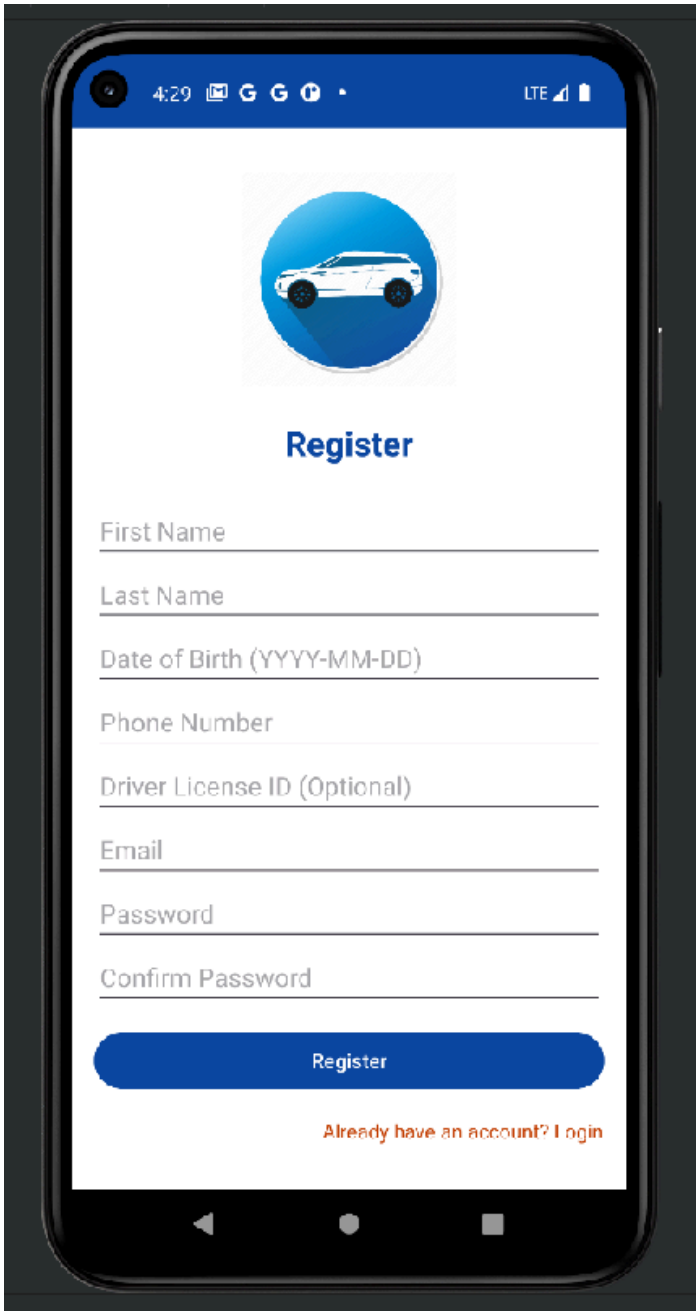
  Utilized Firebase Storage to securely store and manage high-resolution images of cars, ensuring efficient retrieval and scalability integration with the system's database for an enhanced user experience.

3. **APIs**
   - **Google Login API**: Establishes user sign-in and account management.
   - **Stripe API**: Secure online payment processing.
   - **Google Location API**: Provide valid and dynamic location services, including car's pickup location.
   - **Google Calendar API**: For integrating booking schedules and reminders into customers gmail calendar.
4. **Platforms**
   - **Firebase Console**: For monitoring app performance, usage analytics, and push notifications.
   - **GitHub**: Allows version control and updates, and collaboration.
5. **Testing**
   - **Emulators**: Used Pixel 5 API 30 for app testing.

User Interface

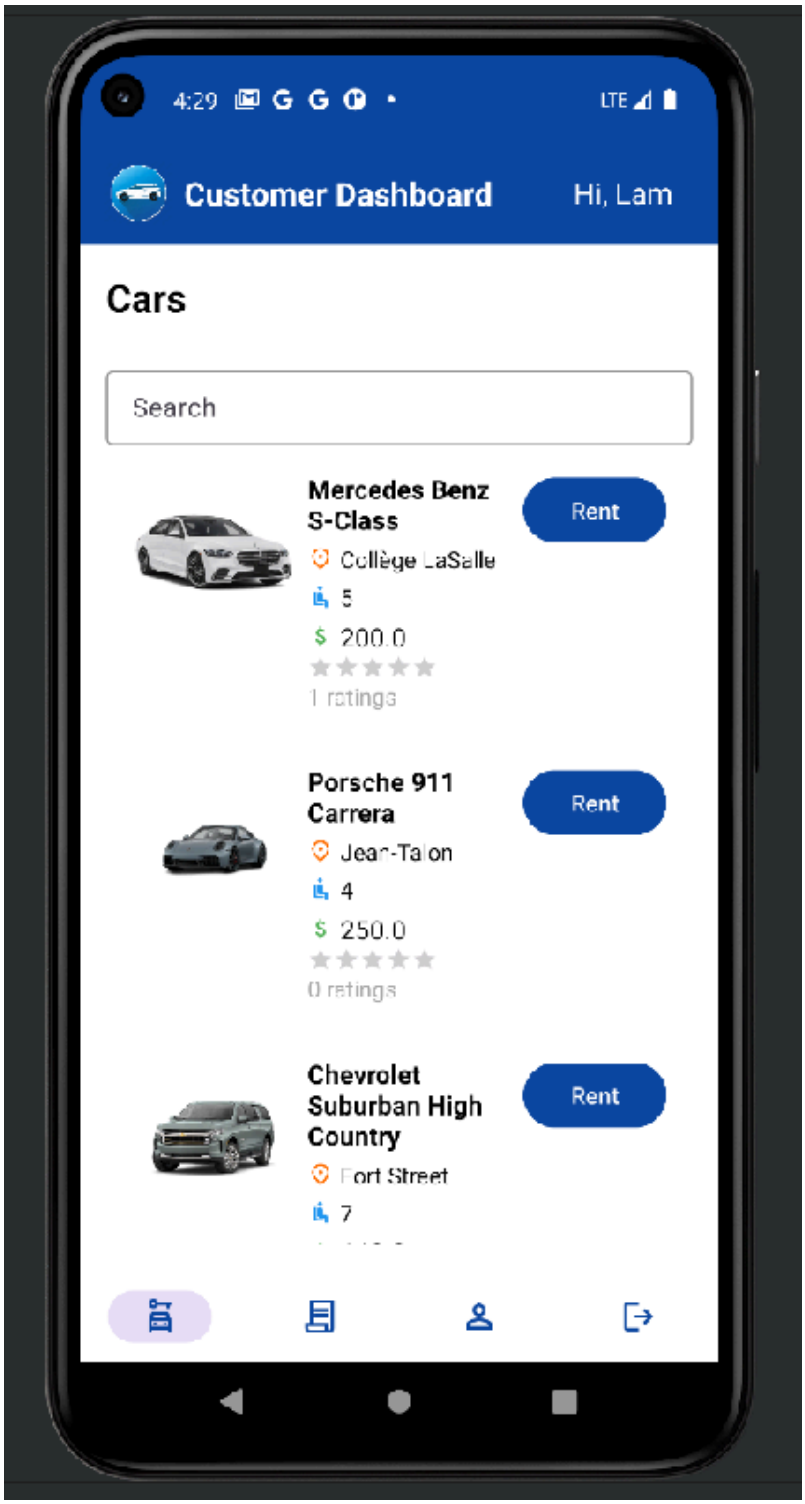
Login and Register UI :



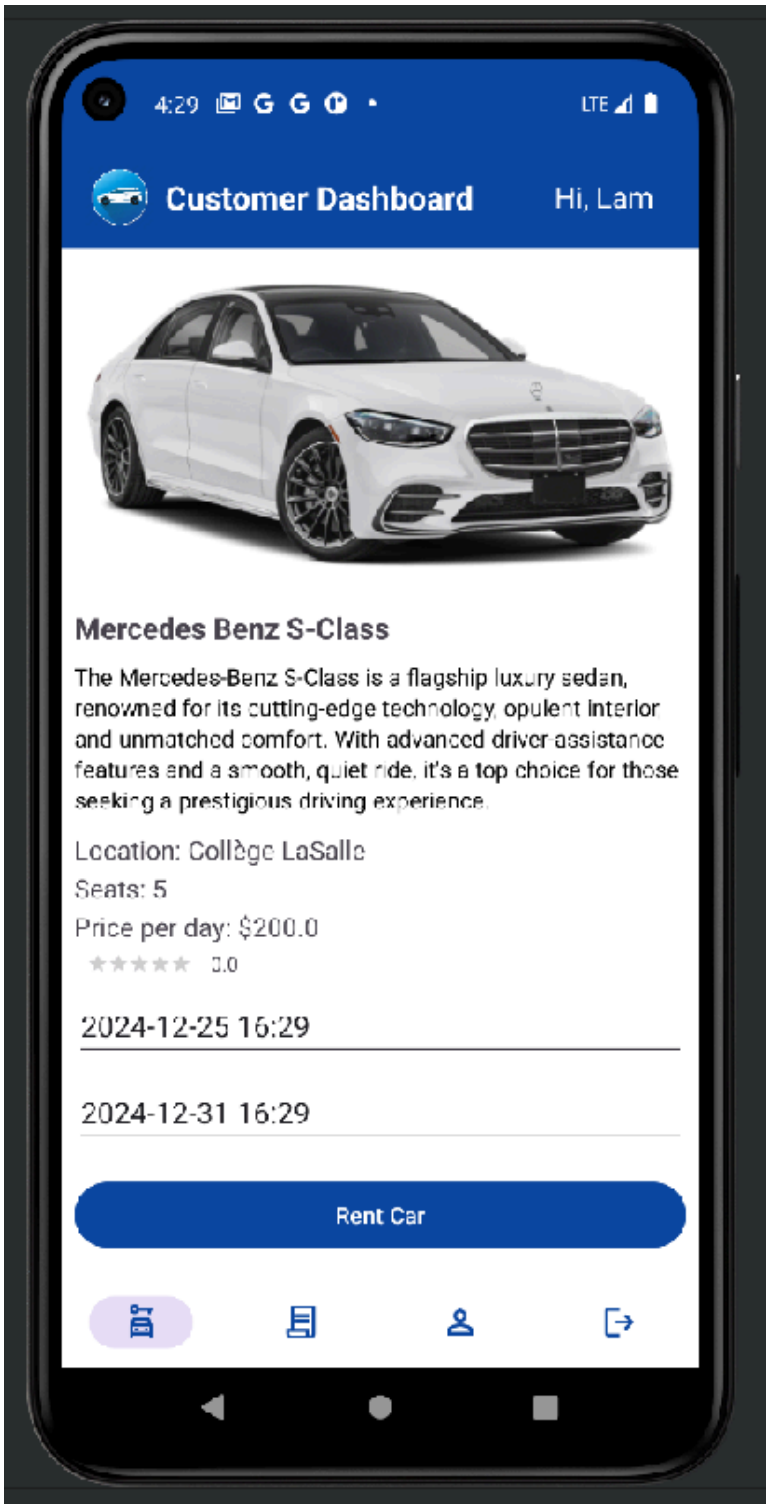


Customer Dashboard:

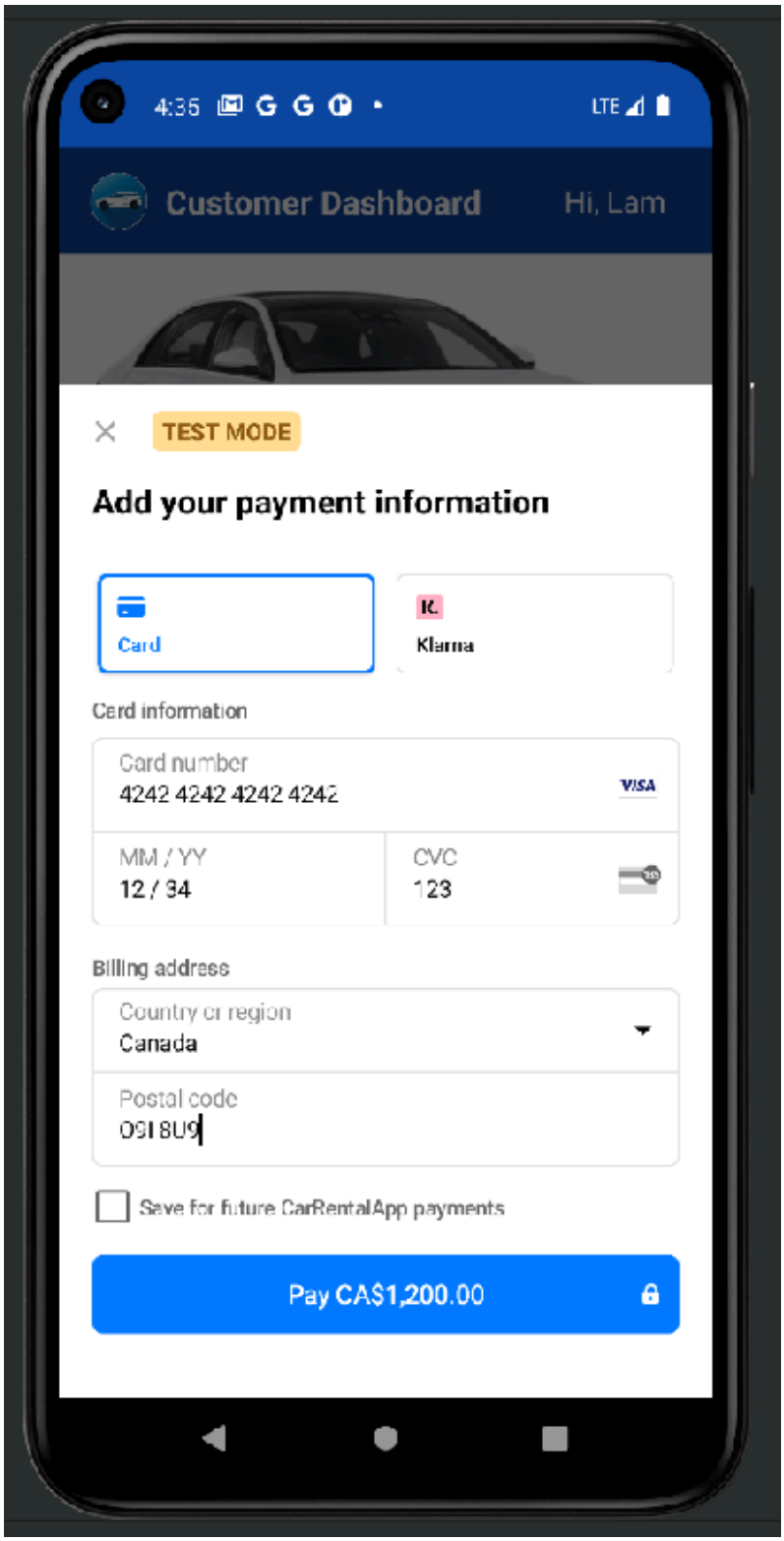
Main UI:



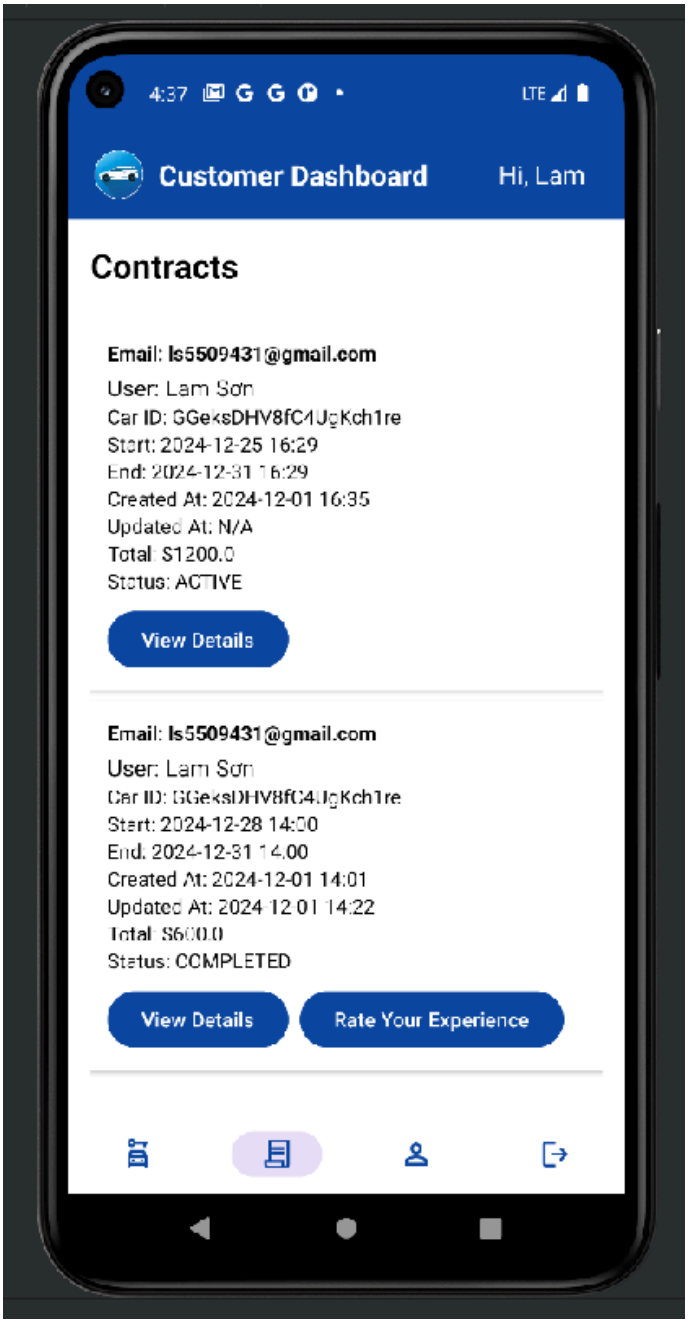
Rent Car UI:

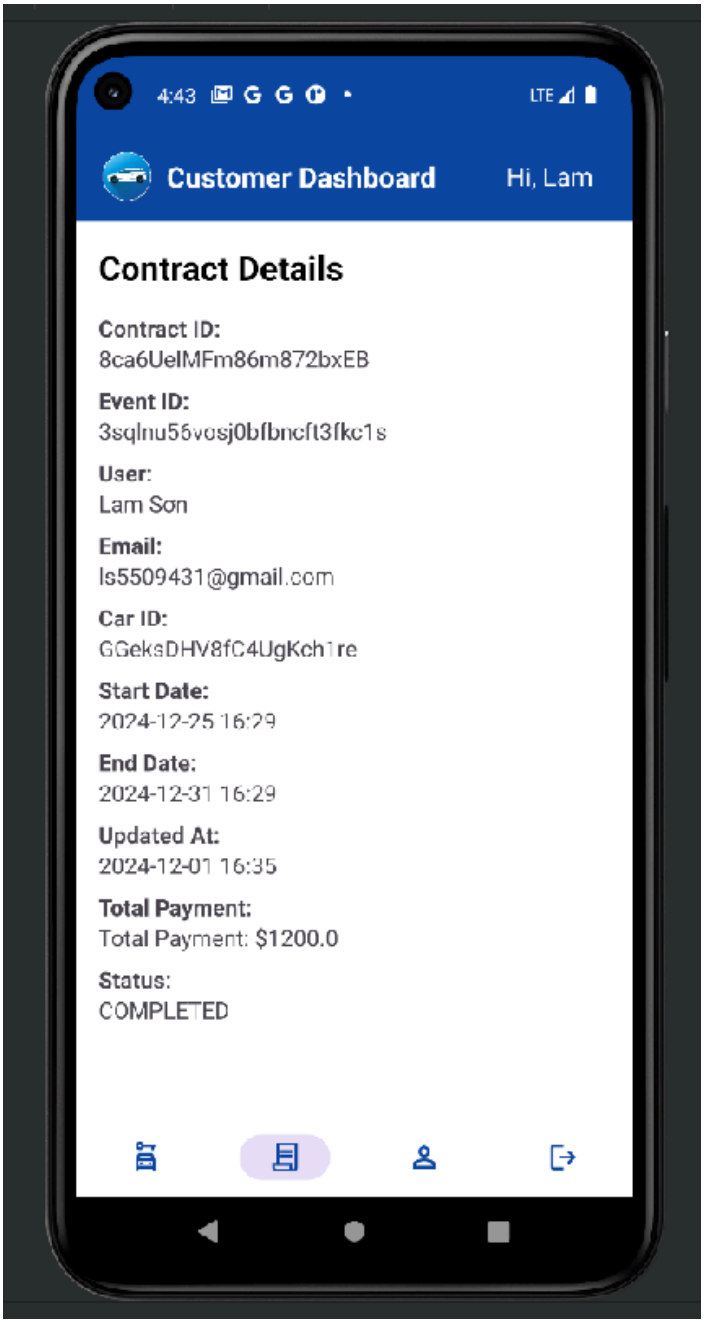

4:29
LTE
Customer Dashboard
Hi, Lam
Mercedes Benz S-Class
The Mercedes-Benz S-Class is a flagship luxury sedan, renowned for its cutting-edge technology, opulent interior and unmatched comfort. With advanced driver assistance features and a smooth, quiet ride, it's a top choice for those seeking a prestigious driving experience.
Location: Collège LaSalle
Seats: 5
Price per day: $200.0
3.0
2024-12-25 16:29
2024-12-31 16:29
Rent Car

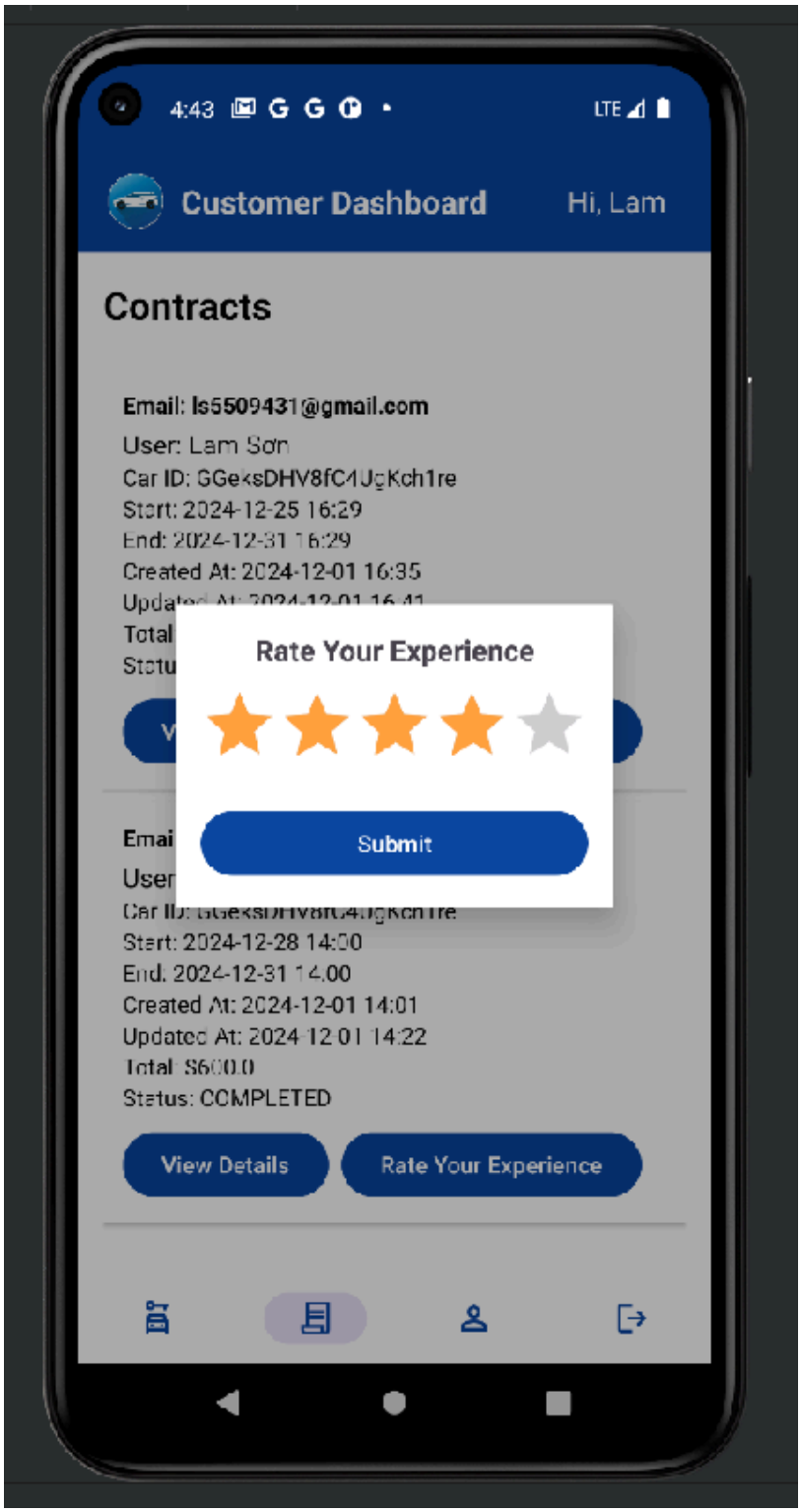
Payment UI:


4:36
LTE
Customer Dashboard
Hi, Lam
TEST MODE
Add your payment information
Card
Klarna
Card information
Card number
4242 4242 4242 4242
VISA
MM / YY
12 / 34
CVC
123
Billing address
Country or region
Canada
Postal code
09I 8U9
Save for future CarRentalApp payments
Pay CA$1,200.00

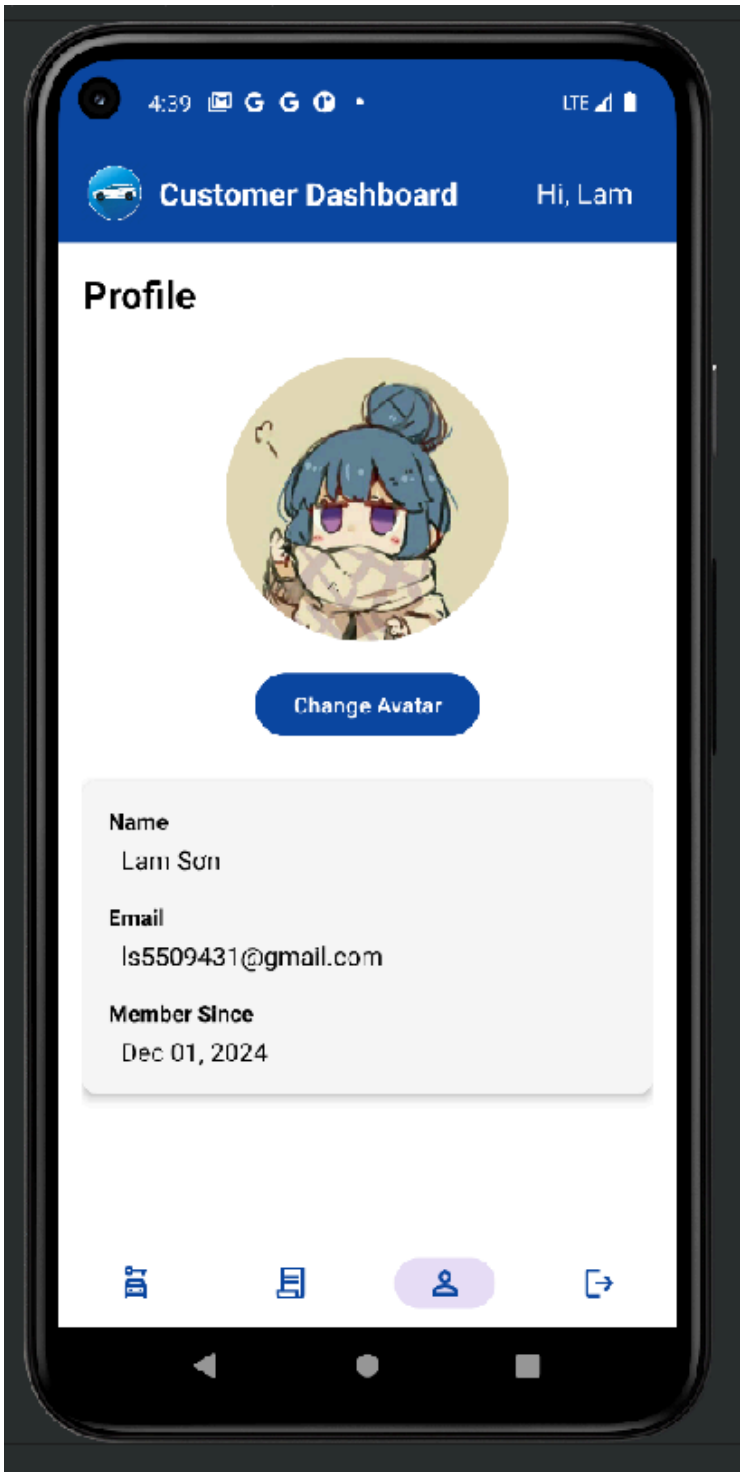
Contracts UI:





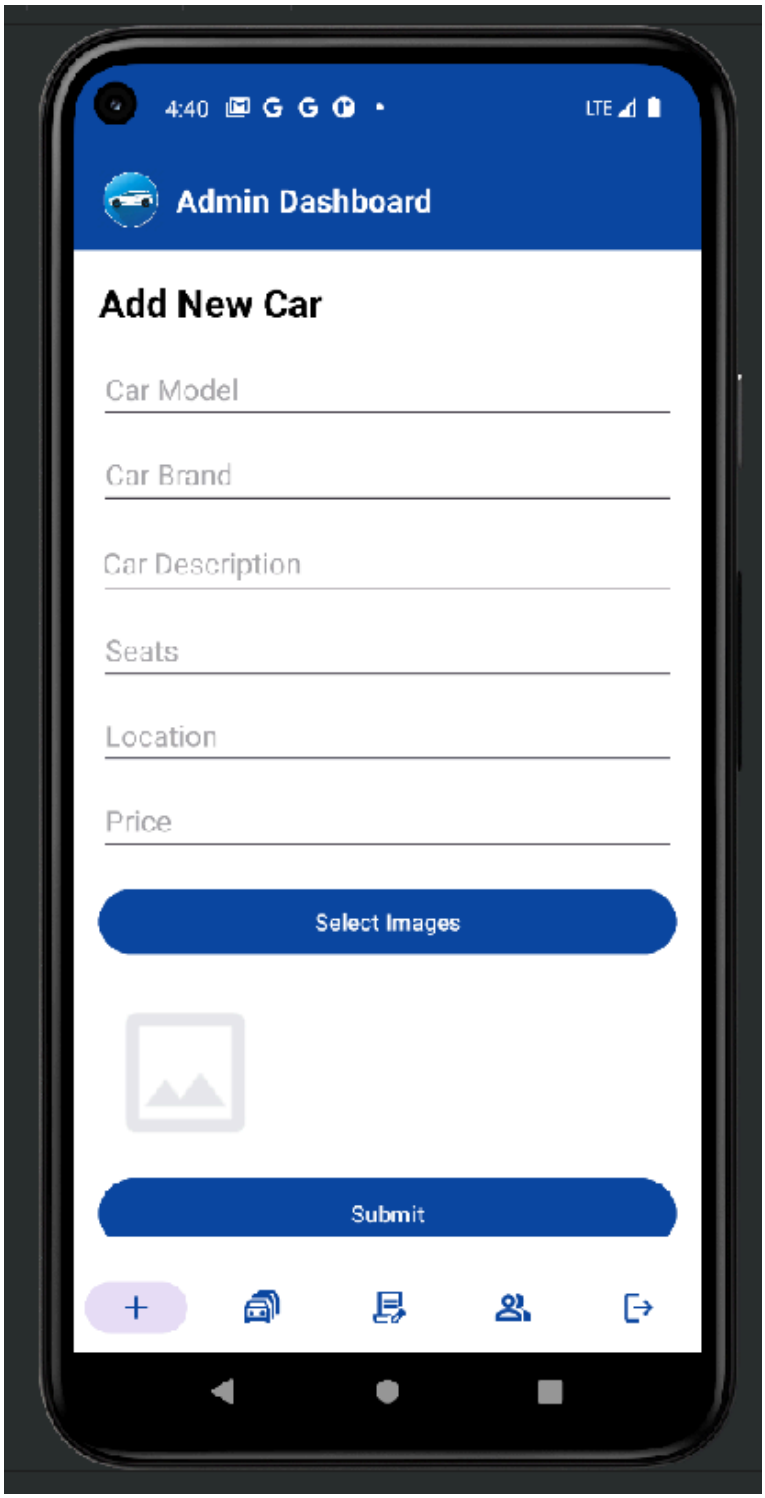

4:43
LTE
Customer Dashboard
Hi, Lam
Contracts
Email: ls5509431@gmail.com
User: Lam Son
Car ID: GGeksDHV8fC4UgKch1re
Start: 2024-12-25 16:29
End: 2024-12-31 16:29
Created At: 2024-12-01 16:35
Rate Your Experience
Submit
Start: 2024-12-28 14:00
End: 2024-12-31 14:00
Created At: 2024-12-01 14:01
Updated At: 2024-12-01 14:22
Total: $600.0
Status: COMPLETED
View Details
Rate Your Experience

Profile UI:

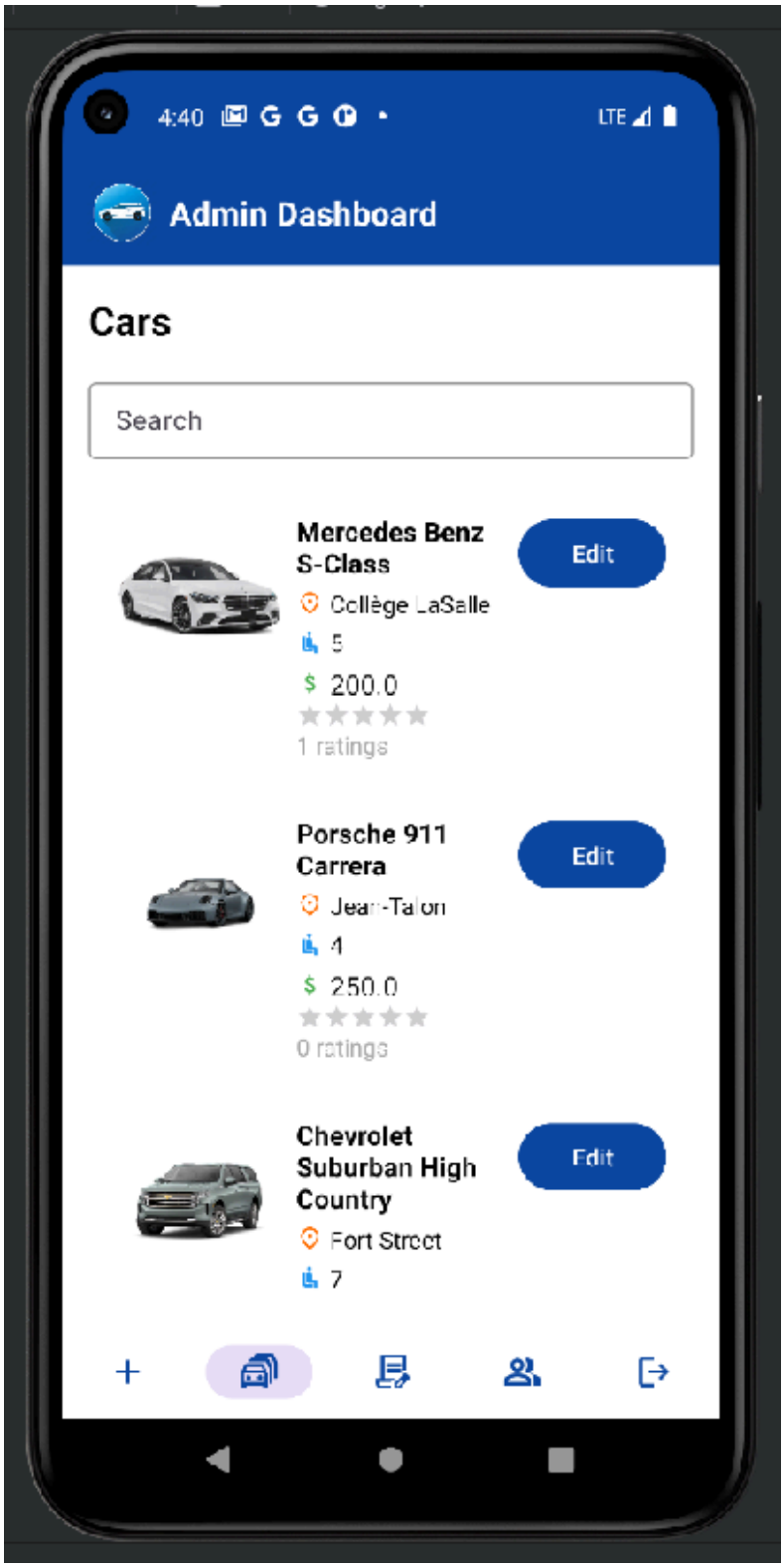

4:39
LTE
Customer Dashboard
Hi, Lam
Profile
Change Avatar
Name
Lam Sơn
Email
ls5509431@gmail.com
Member Since
Dec 01, 2024

Admin Dashboard:

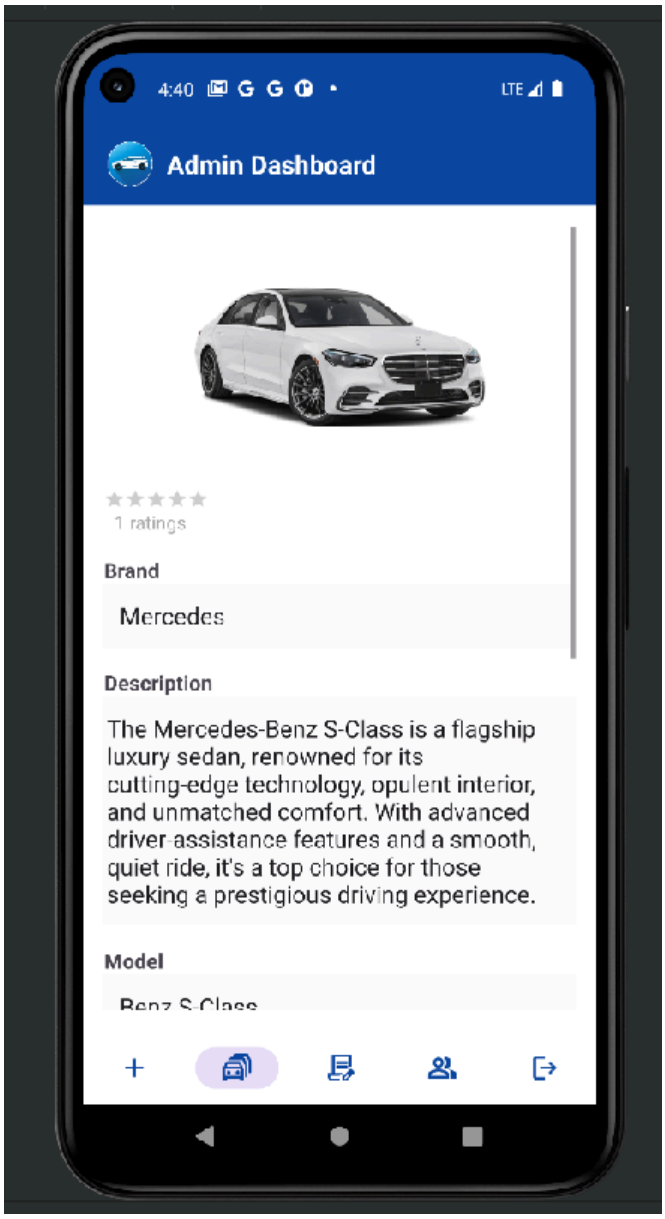
Add Car UI:

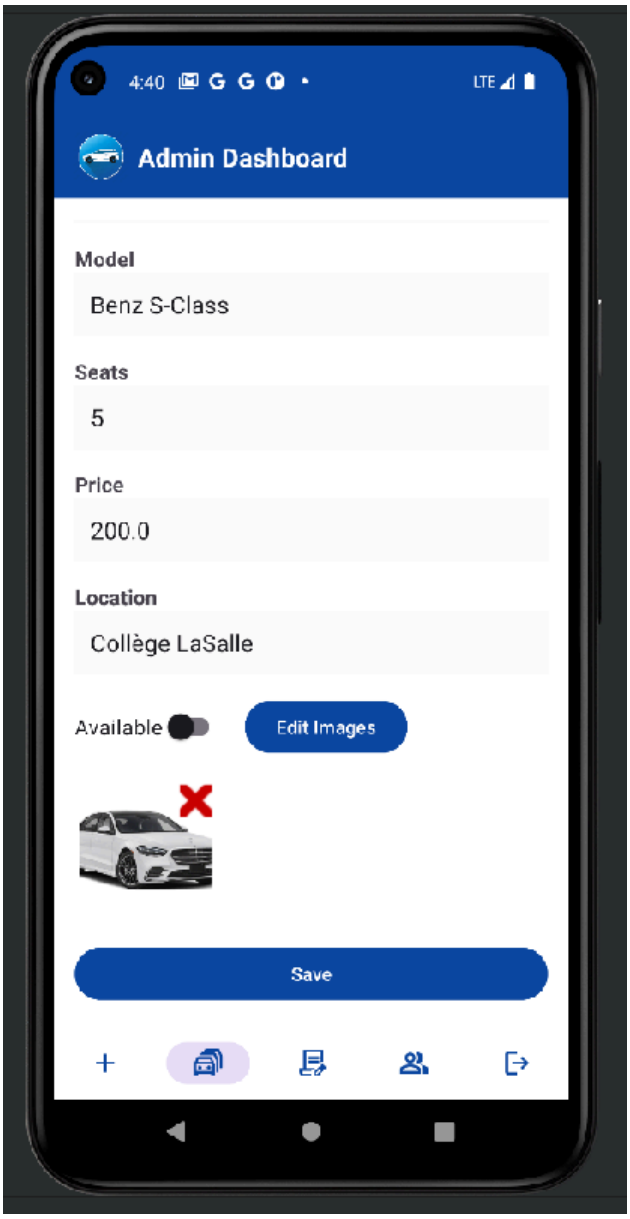


View Car List UI:

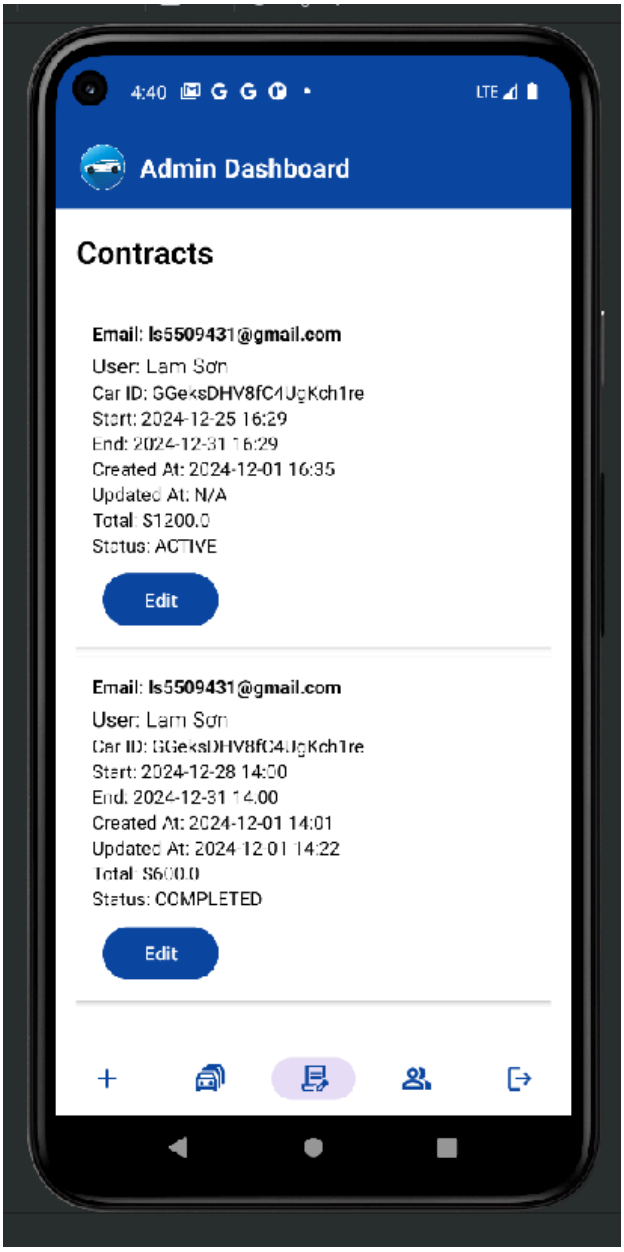


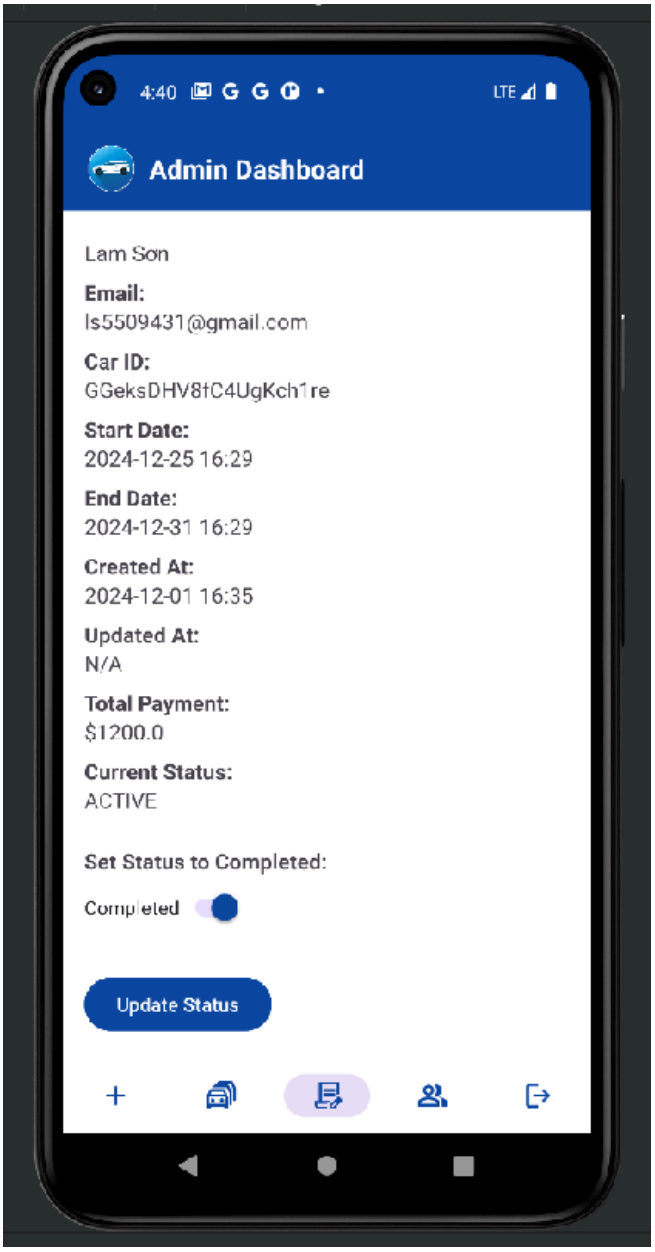
Edit Car UI:

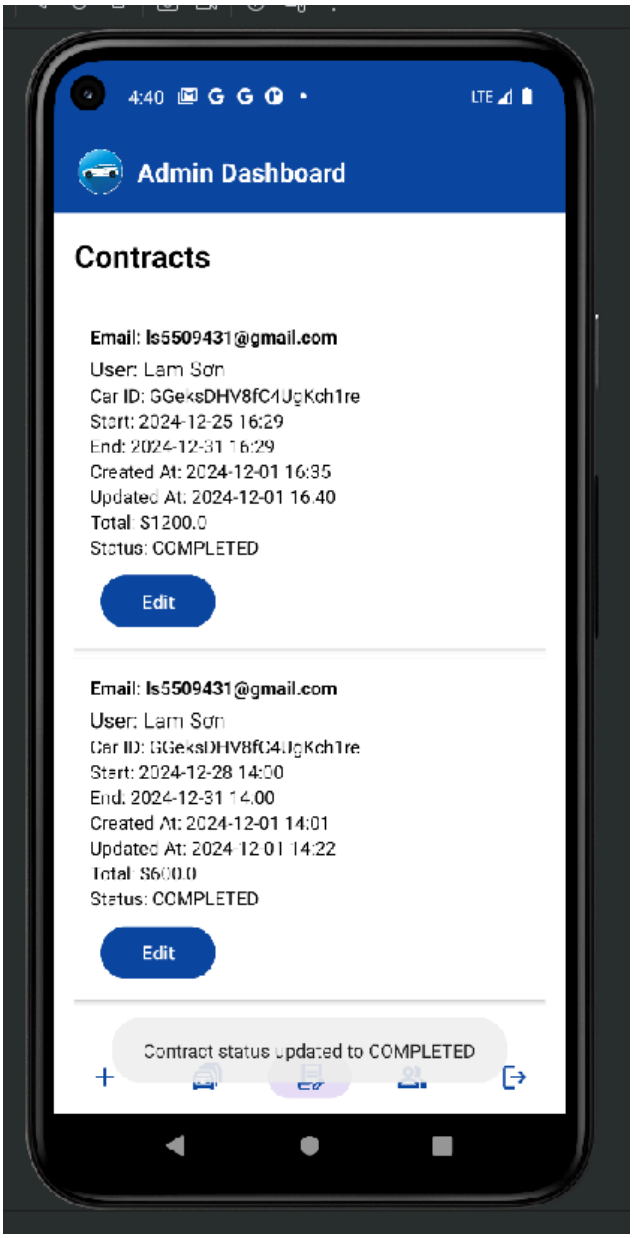

4:40
LTE
Admin Dashboard
1 ratings
Brand
Mercedes
Description
The Mercedes-Benz S-Class is a flagship luxury sedan, renowned for its cutting-edge technology, opulent interior, and unmatched comfort. With advanced driver-assistance features and a smooth, quiet ride, it's a top choice for those seeking a prestigious driving experience.
Model

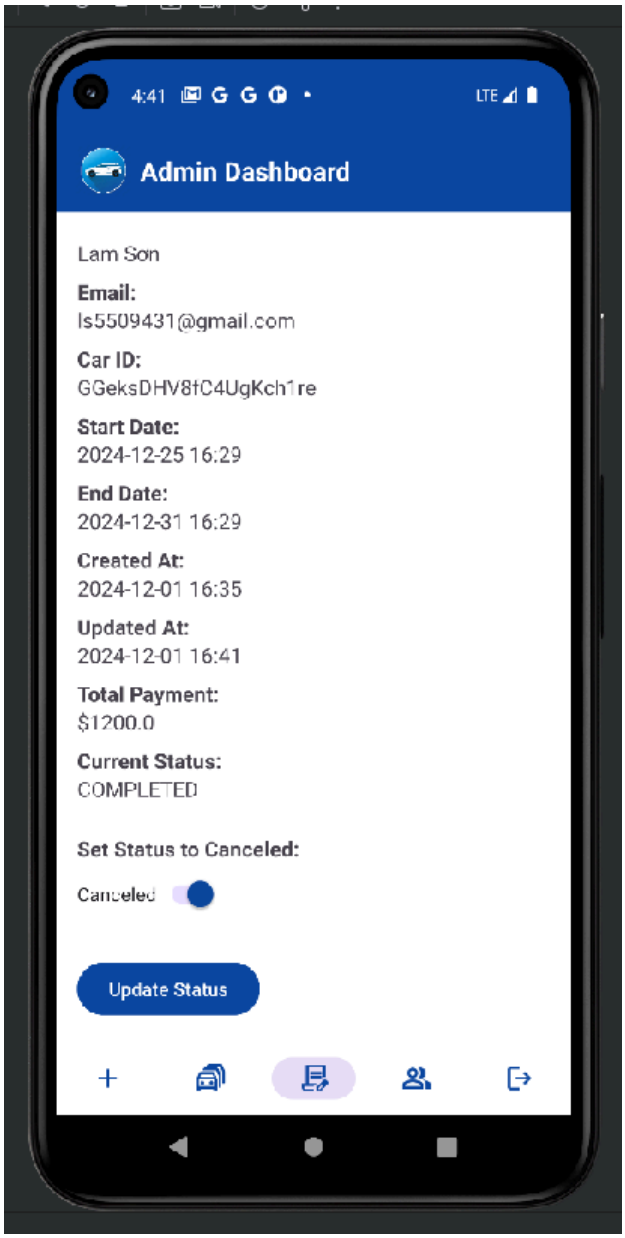

4:40
LTE
Admin Dashboard
Model
Benz S-Class
Seats
5
Price
200.0
Location
Collège LaSalle
Available
Edit Images
Save

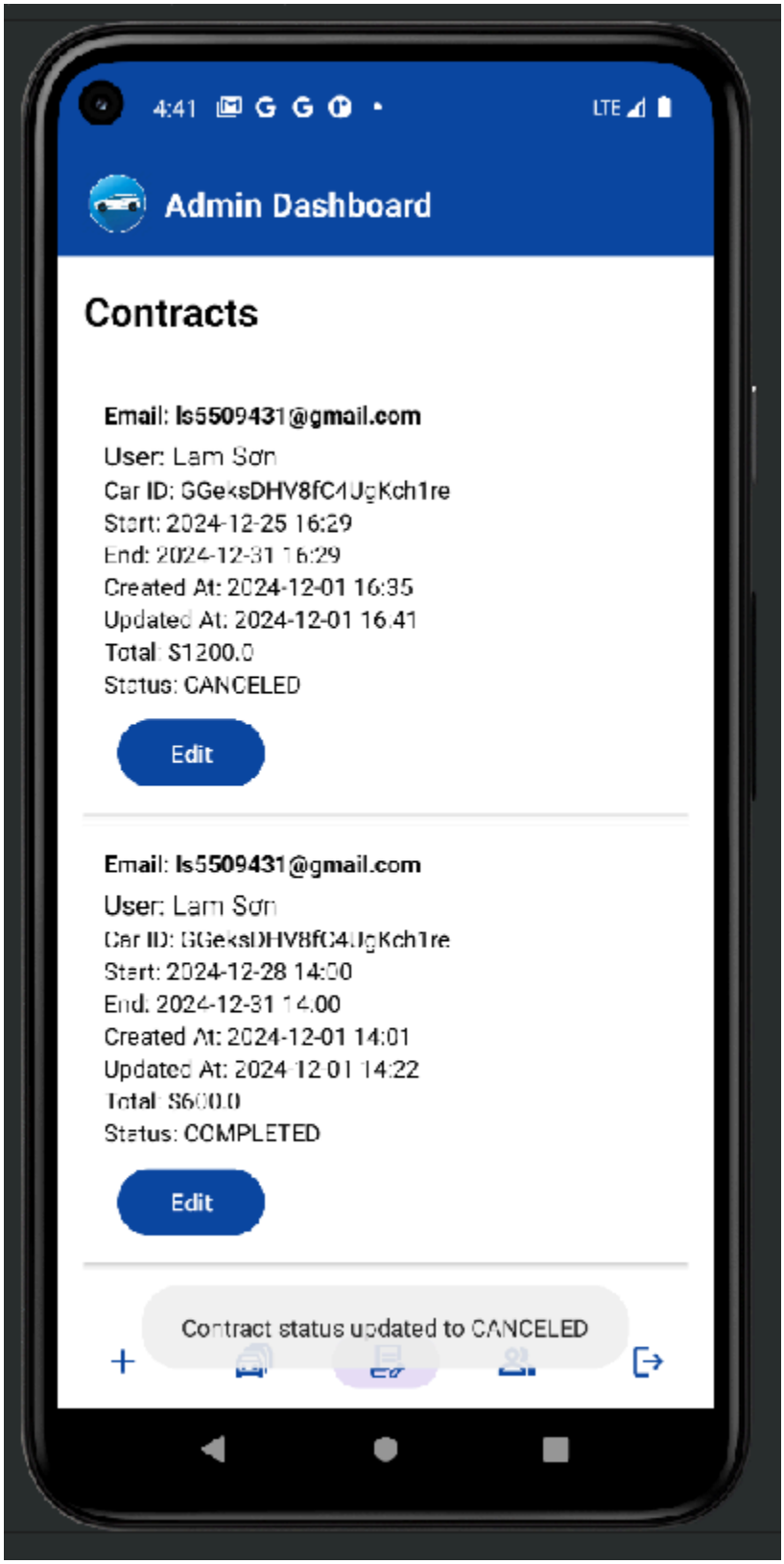
View Contracts & Edit Contract UI:



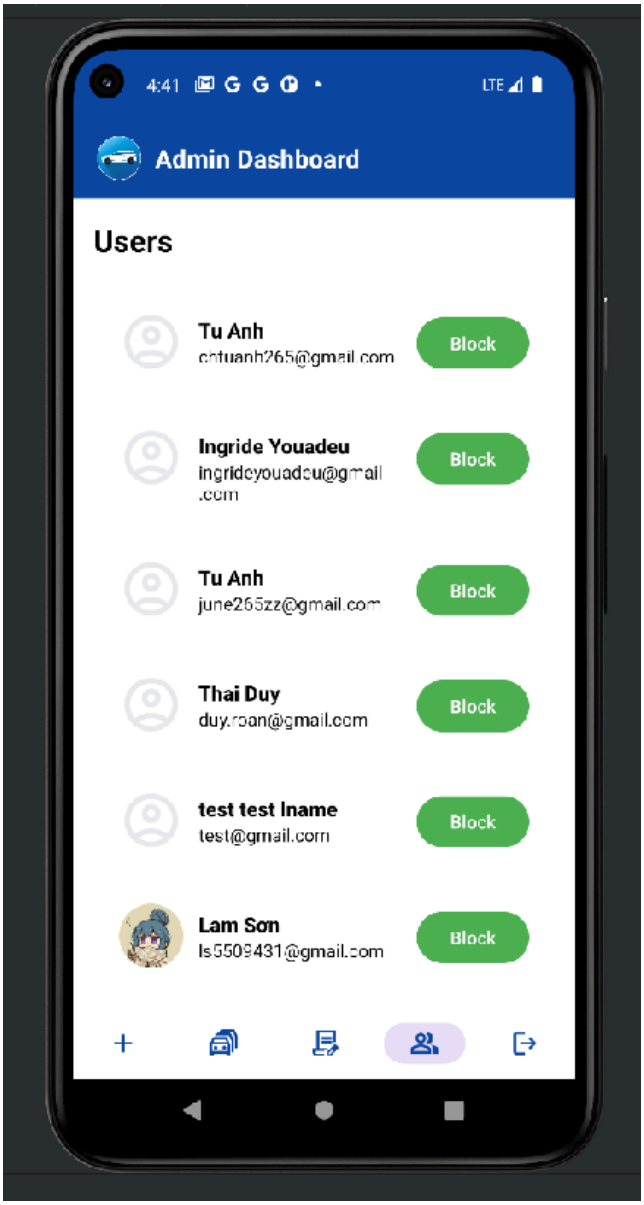


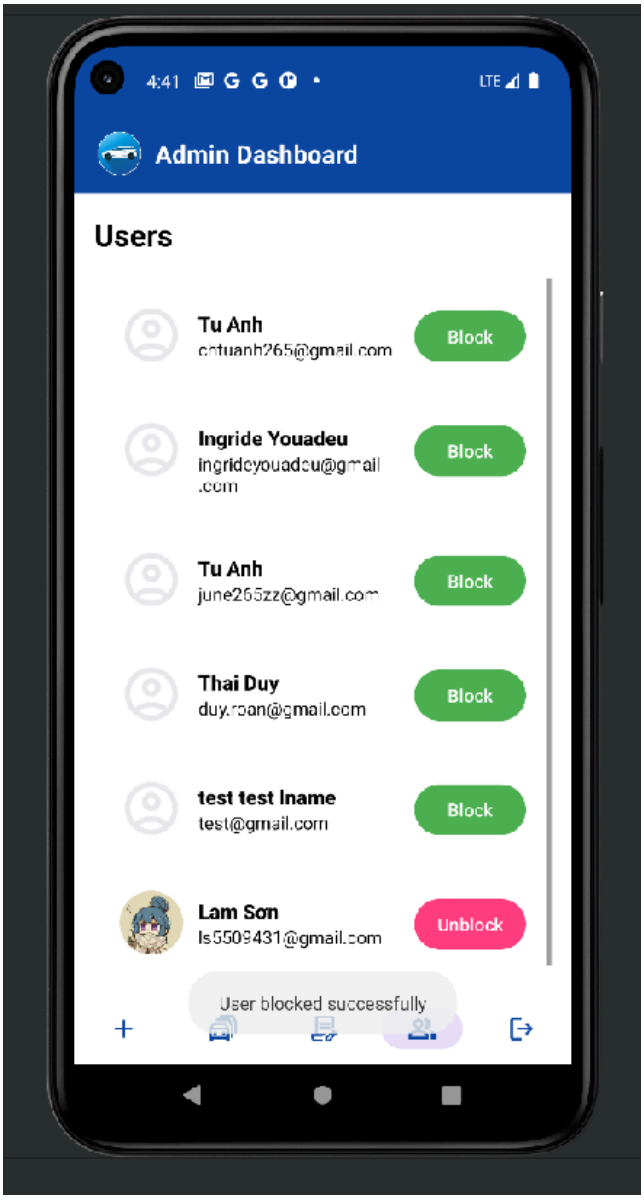

4:40
LTE
Admin Dashboard
Contracts
Email: ls5509431@gmail.com
User: Lam Son
Car ID: GGeksDHV8fC4UgKch1re
Start: 2024-12-25 16:29
End: 2024-12-31 16:29
Created At: 2024-12-01 16:35
Updated At: 2024-12-01 16:40
Total: $1200.0
Status: COMPLETED
Edit
Email: ls5509431@gmail.com
User: Lam Son
Car ID: GGeksDHV8fC4UgKch1re
Start: 2024-12-28 14:00
End: 2024-12-31 14:00
Created At: 2024-12-01 14:01
Updated At: 2024-12-01 14:22
Total: $600.0
Status: COMPLETED
Edit
Contract status updated to COMPLETED


4:41
LTE
Admin Dashboard
Lam Son
Email:
ls5509431@gmail.com
Car ID:
GGeksDHV8fC4UgKch1re
Start Date:
2024-12-25 16:29
End Date:
2024-12-31 16:29
Created At:
2024-12-01 16:35
Updated At:
2024-12-01 16:41
Total Payment:
$1200.0
Current Status:
COMPLETED
Set Status to Canceled:
Canceled
Update Status


4:41
LTE
Admin Dashboard
Contracts
Email: ls5509431@gmail.com
User: Lam Sơn
Car ID: GGeksDHV8fC4JgKch1re
Start: 2024-12-25 16:29
End: 2024-12-31 16:29
Created At: 2024-12-01 16:35
Updated At: 2024-12-01 16:41
Total: $1200.0
Status: CANCELED
Edit
Email: ls5509431@gmail.com
User: Lam Sơn
Car ID: GGeksDHV8fC4JgKch1re
Start: 2024-12-28 14:00
End: 2024-12-31 14:00
Created At: 2024-12-01 14:01
Updated At: 2024-12-01 14:22
Total: $600.0
Status: COMPLETED
Edit
Contract status updated to CANCELED

View User's Profiles UI:





How to set up the Application

FIRST STEP:
Project >> New >> Java Class

Firebase Configuration Class:

```
public class FirebaseConfig extends Application {
   public void onCreate() {
       super.onCreate();

       FirebaseOptions options = new FirebaseOptions.Builder()
               .setApiKey("AIzaSyBdfg_G19mw5tWqsHupVOb56gLvOl5cLeY")

.setApplicationId("1:655767929093:android:62d9fdc09cc7a93f17376c")

.setDatabaseUrl("https://carrentalmanagment-default-rtdb.firebaseio.com")
               .setProjectId("carrentalmanagment")
               .setStorageBucket("carrentalmanagment.appspot.com")
               .build();

       FirebaseApp.initializeApp(this, options);
       Log.d("FirebaseInit", "Firebase initialized in
CarRentalAppApplication");
   }
}
```

SECOND STEP:
Modify AndroidManifest.xml:

```
<application
   ……
   //add this line
   android:name=".FirebaseConfig"
   …….
</application>
```

THIRD STEP:
Gradle (App level):

```
// comment out this line
alias(libs.plugins.google.gms.google.services)
```

FOURTH STEP:
Gradle (Project level):

```
// comment out this line
alias(libs.plugins.google.gms.google.services) apply false
```

!!! SYNC PROJECT WITH GRADLE FILES BEFORE RUNNING

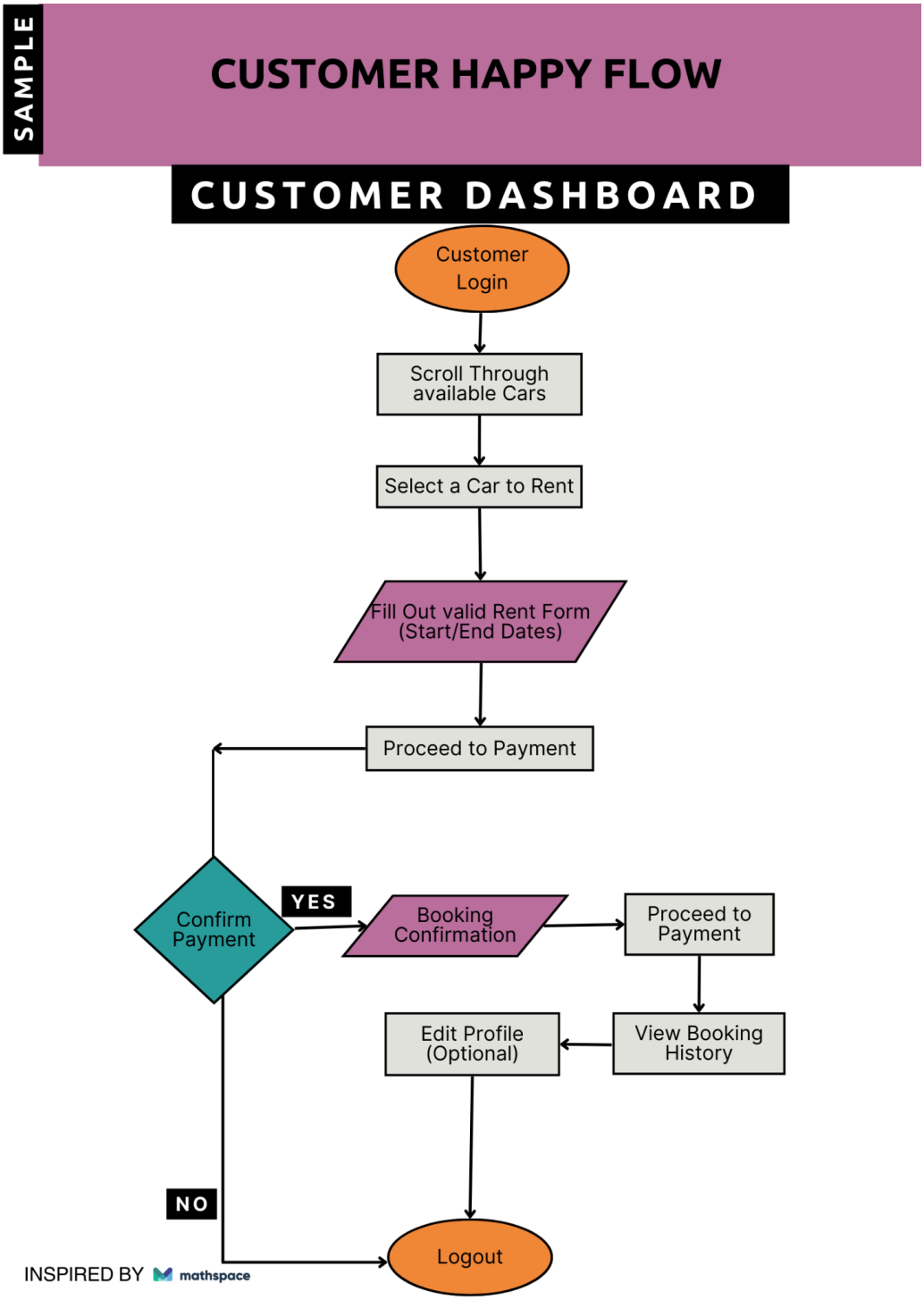
FINAL STEP:
Run this command line in Terminal of the project:

- node server .js

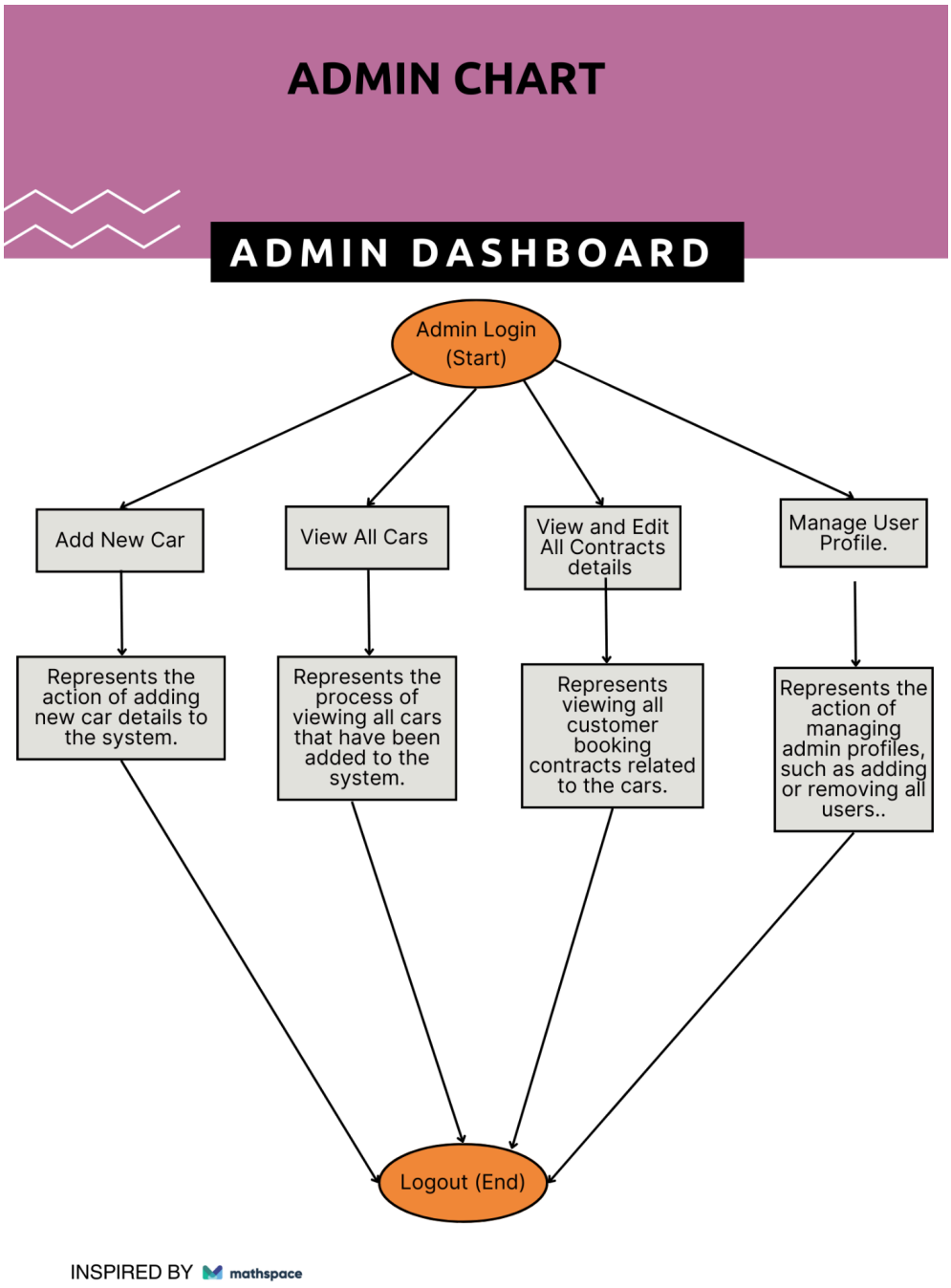
=> Enable Stripe API

Happy Flow

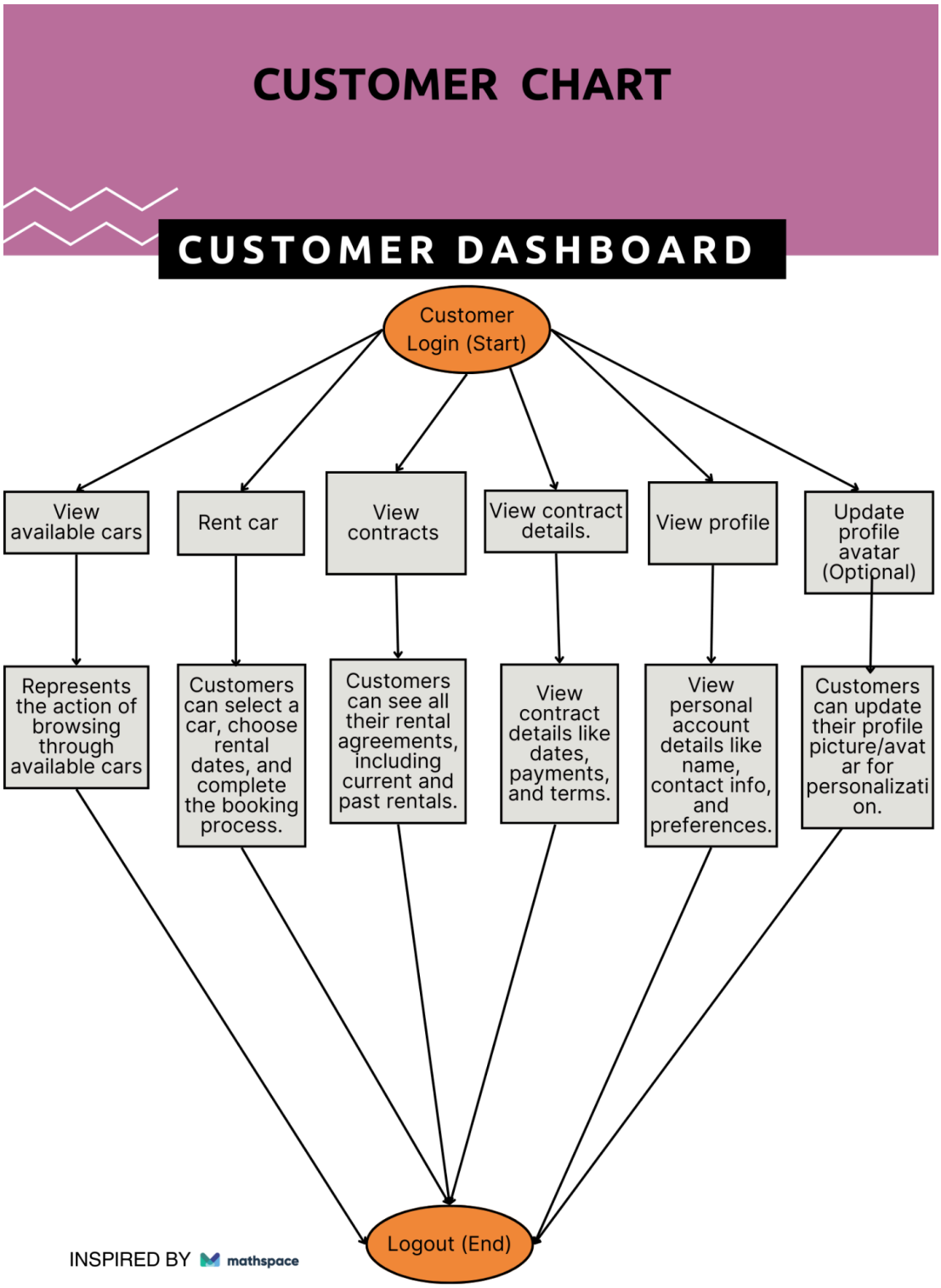
Customer's Flow:



Admin's Chart:



Customer's Chart:



Sample Password:

Password For Customer registration:

```
99
100     private void updatePasswordStrengthUI(int strength) {
101         switch (strength) {
102             case 0:
103
104                 passwordStrengthText.setText("Very Weak");
105                 passwordStrengthText.setTextColor(getResources().getColor(R.color.holo_red_dar
106                 break;
107             case 1:
108             case 2:
109                 //pass or Pass or pass12
110                 passwordStrengthText.setText("Weak");
111                 passwordStrengthText.setTextColor(getResources().getColor(R.color.secondaryDar
112                 break;
113             case 3:
114                 //Pass12
115                 passwordStrengthText.setText("Good");
116                 passwordStrengthText.setTextColor(getResources().getColor(R.color.secondaryCol
117                 break;
118             case 4:
119                 //password%12
120                 passwordStrengthText.setText("Strong");
121                 passwordStrengthText.setTextColor(getResources().getColor(R.color.secondaryLig
122                 break;
123             case 5:
124                 //Password%12
125                 passwordStrengthText.setText("Very Strong");
126                 passwordStrengthText.setTextColor(getResources().getColor(R.color.colorActive)
127                 break;
128         }
129     }
```