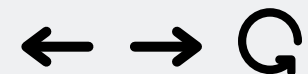




Lesson



Baku Engineering University(26.01.2024)

Mobile Programming

Nabi Shirinov

Lecturer : Yusif Yusifov



Lesson

App Overview



← → 🔍 Group Members

Rental App Overview



The rental car app lets users share their cars for rent, allowing others to easily find and learn about available rentals. Users add rentals for renting.



Lesson

App Overview

Clean Structure



← → 🔍 Insert your topic here

Clean Structure

My code features a clean structure with well-named classes and modular design, enhancing readability and maintainability.

```
java
├── com.example.rentalcarmobile
│   ├── adapters
│   │   ├── All_Rentals_RecyclerViewAdapter
│   │   └── My_Rentals_RecyclerViewAdapter
│   ├── api
│   │   ├── ApiClient
│   │   └── ApiInterface
│   ├── DBHelper
│   ├── dto
│   │   └── RentalResponse
│   ├── entities
│   │   ├── Rental
│   │   └── User
│   ├── fragments
│   │   ├── HomeFragment
│   │   ├── ProfileFragment
│   │   └── SettingsFragment
│   ├── interfaces
│   │   └── RecyclerViewInterface
│   ├── screens
│   │   ├── AddRentalActivity
│   │   ├── LoginActivity
│   │   ├── RentalSeperateActivity
│   │   └── SignUpActivity
│   └── Service
│       ├── MySharedPreferences
│       └── MainActivity
```




Lesson

App Overview

Clean Structure

Database Connection

CRUD operations



← → 🔍 Insert your topic here

CRUD operations(Rentals & User)

```
@POST("api/v1/auth/register")
Call<User> saveUser(@Body User user);

1 usage  👤 NabiShirinov
@POST("api/v1/auth/login")
Call<User> loginUser(@Body User user);

1 usage  👤 NabiShirinov
@GET("api/v1/rental/all")
Call<RentalResponse> getAllRentals(@Header("Authorization") String accessToken);

1 usage  👤 NabiShirinov
@GET("api/v1/rental/myRentals")
Call<RentalResponse> getMyRentals(@Header("Authorization") String accessToken);

1 usage  👤 NabiShirinov
@POST("api/v1/rental/add")
Call<Rental> addRental(@Header("Authorization") String accessToken,@Body Rental rental);

1 usage  👤 NabiShirinov
@DELETE("/api/v1/user/delete")
Call<Void> deleteUser(@Header("Authorization") String accessToken);|
```



Lesson

App Overview

Clean Structure

Database Connection

CRUD

Login SignUp



← → 🔍 Insert your topic here

Login/Sign Up

7:37

Welcome To
Car Rental App

Login

Email

Password

Login

Don't have account? Create Account

7:37

Welcome To
Car Rental App

Sign Up

Full Name

Email

Password

Confirm Password

Sign Up

Already have account? Login



Lesson

App Overview

Clean Structure

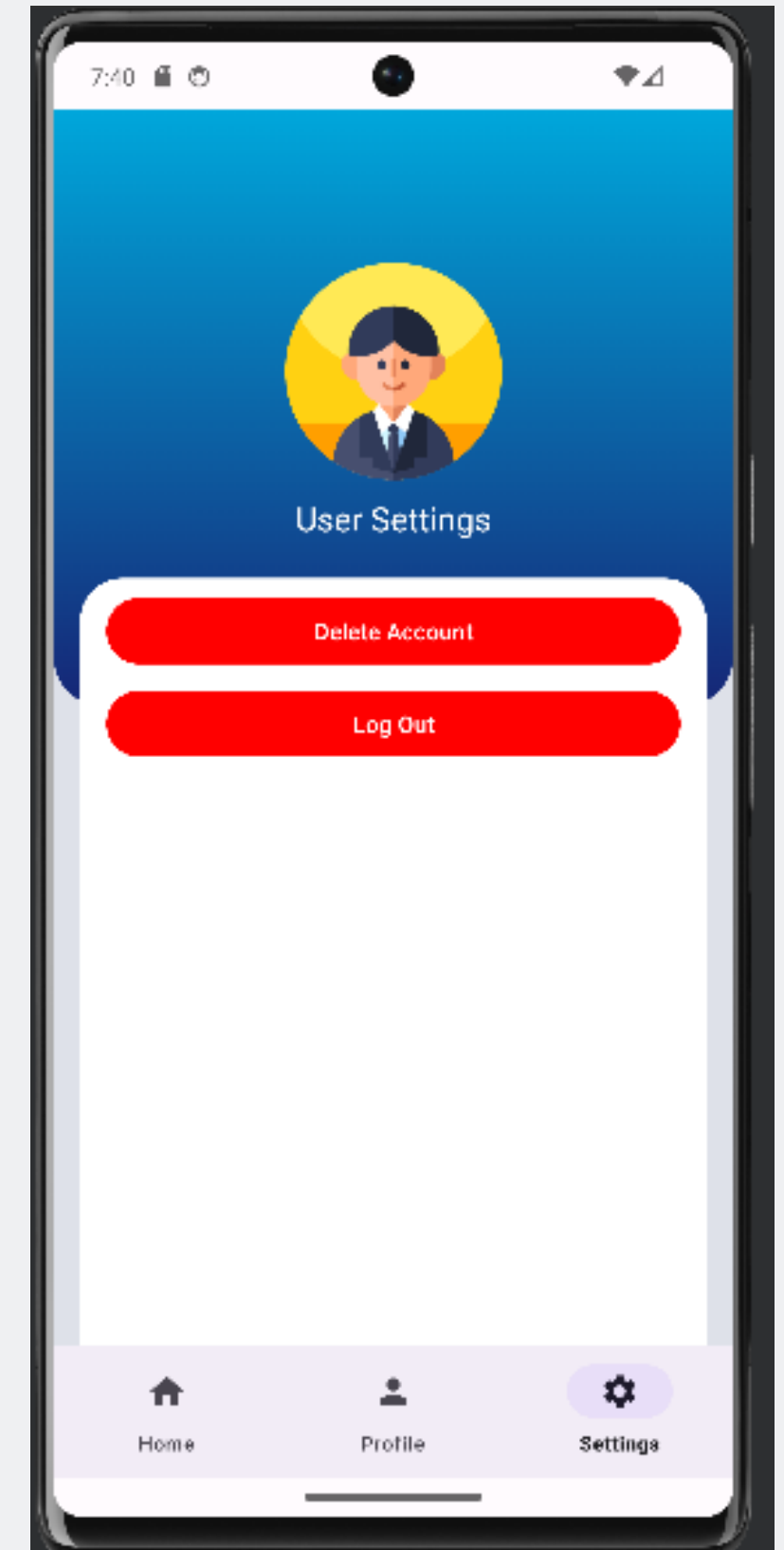
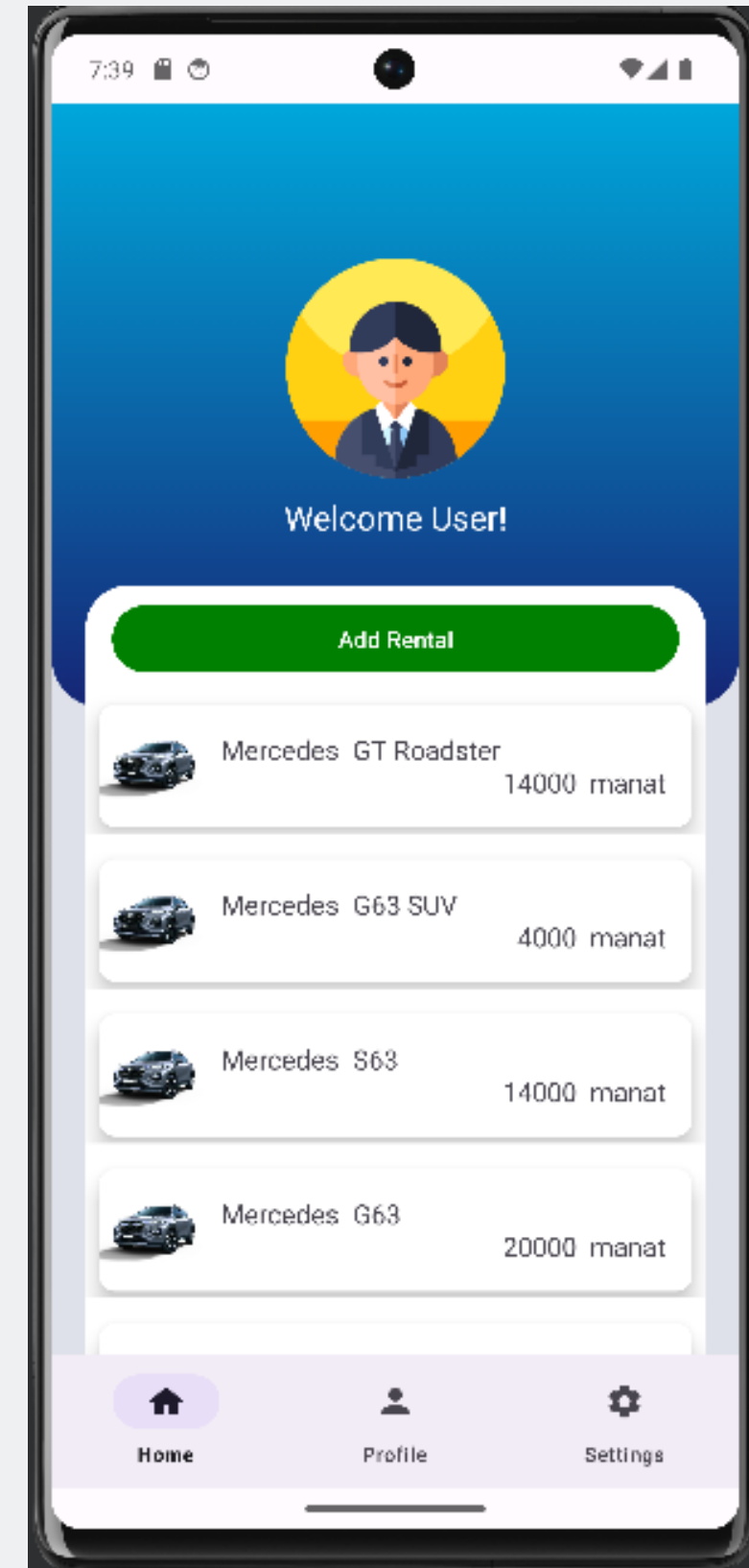
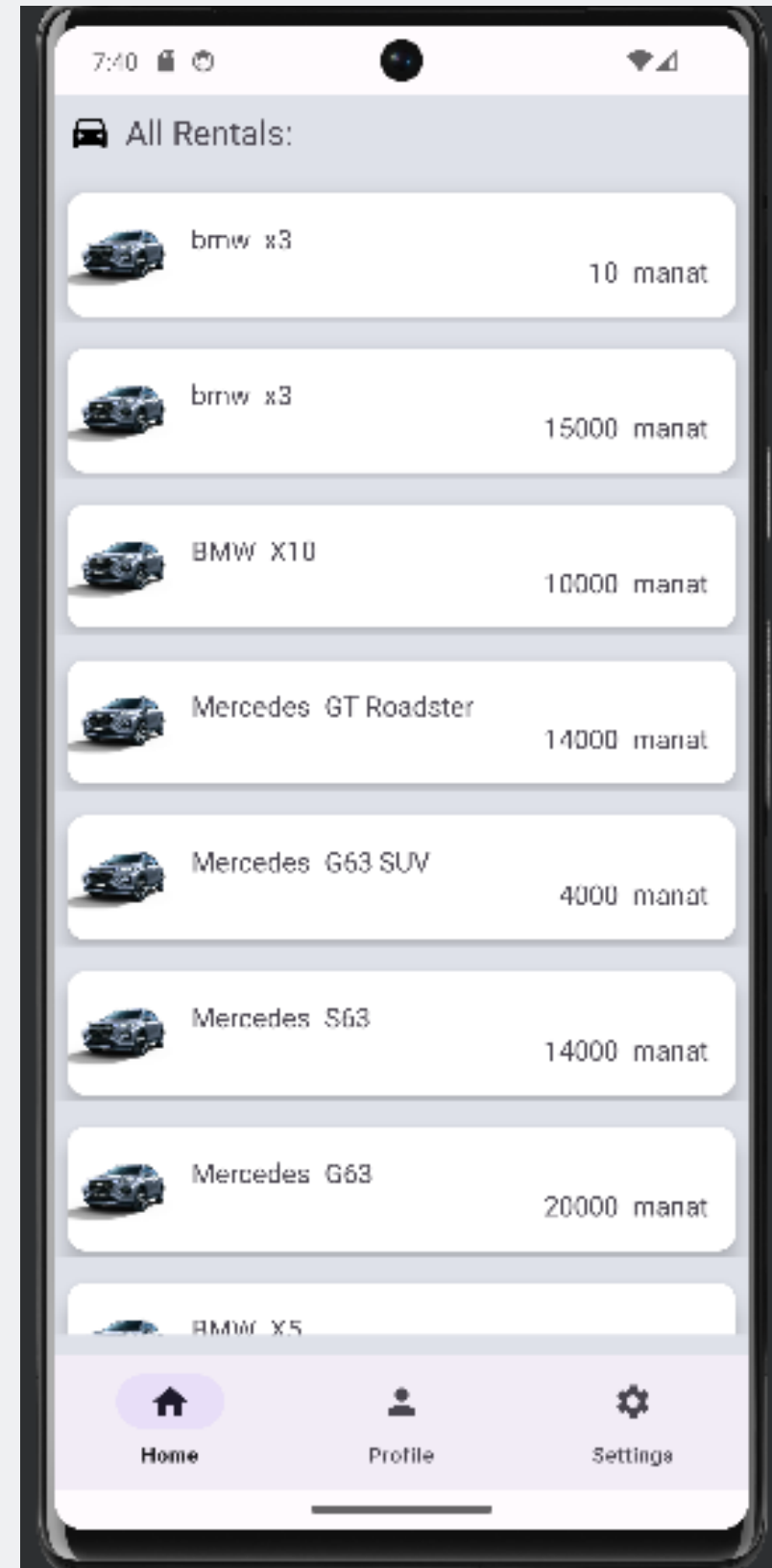
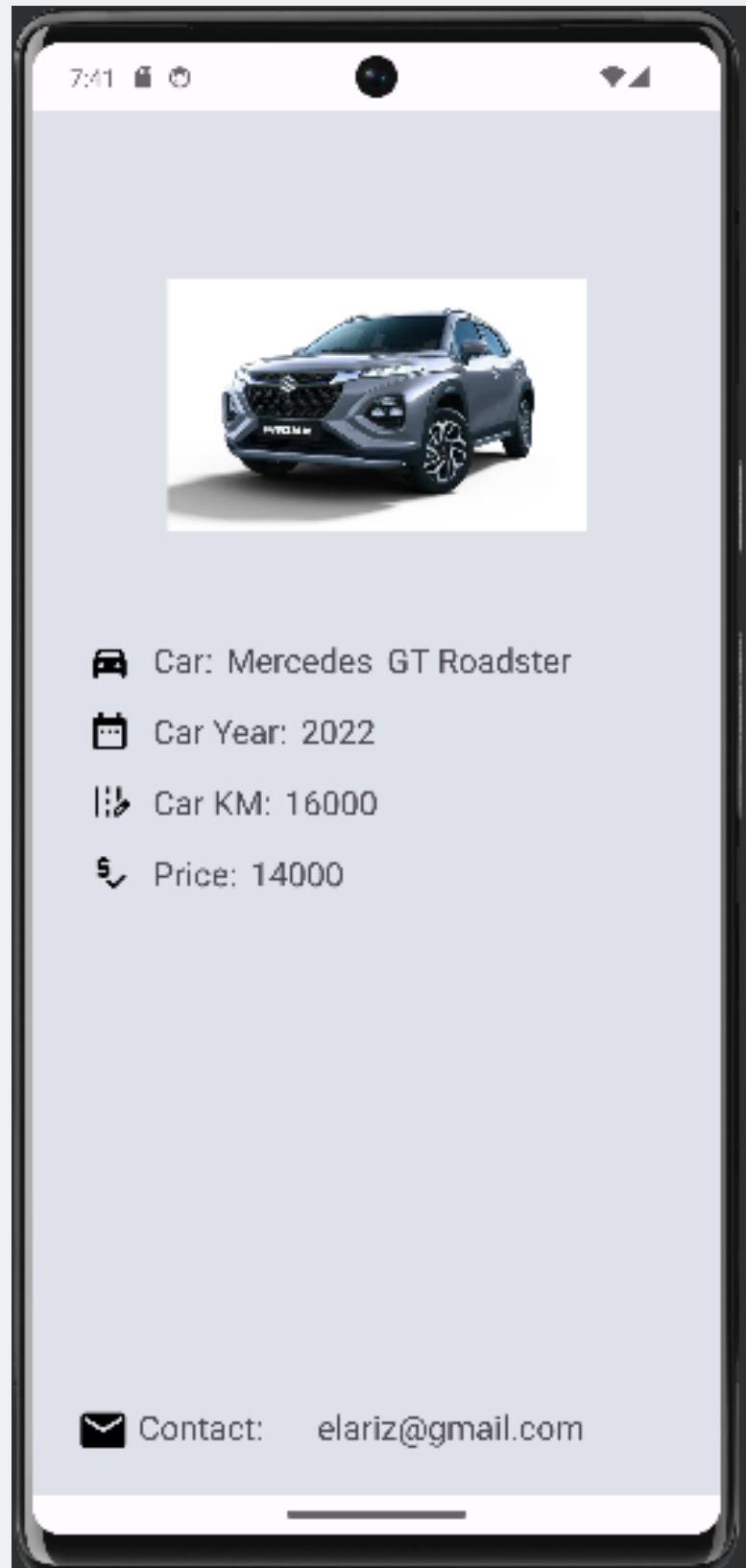
Database Connection

Login/Signup

Design



← → 🔍 Insert your topic here





Error Handling

Effectively managing errors and providing custom exceptions enhances the robustness of the application, ensuring a more graceful handling of unexpected situations and improving the user experience.

```
private void handleErrorResponse(Response<RentalResponse> response) {  
    // Handle different HTTP error codes  
    switch (response.code()) {  
        case 401:  
            showToast("Unauthorized access. Please login again.");  
            // Handle unauthorized access, maybe redirect to login screen  
            break;  
        case 404:  
            showToast("Resource not found.");  
            // Handle resource not found error  
            break;  
        case 500:  
            showToast("Internal server error. Please try again later.");  
            // Handle internal server error  
            break;  
        default:  
            showToast("Error: " + response.code());  
            // Handle other errors  
            break;  
    }  
}
```




Lesson

App Overview

Clean Structure

Database Connection

CRUD

Custom Logo



← → 🔍 🔍 Insert your topic here

Custom Logo



Thank you!