## Final Project Report

**Outline**

1. *Project Statement*
2. *Application Design*
3. *Application Implementation and Evaluation*
4. *References*
5. *Experiences and Thoughts*

**1. Project Statement**

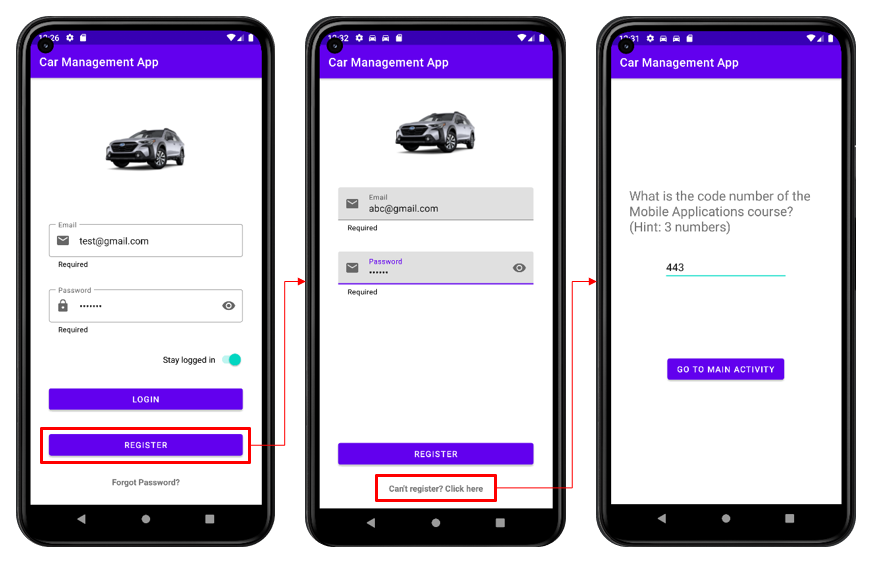
**1) About Car Management App**

This car maintenance app will help people keep track of their car maintenance history. And the app will give reminder based on the specific car model and car maintenance history. Also, you can see a bar chart that shows current year’s monthly fuel consumption amount of your car.

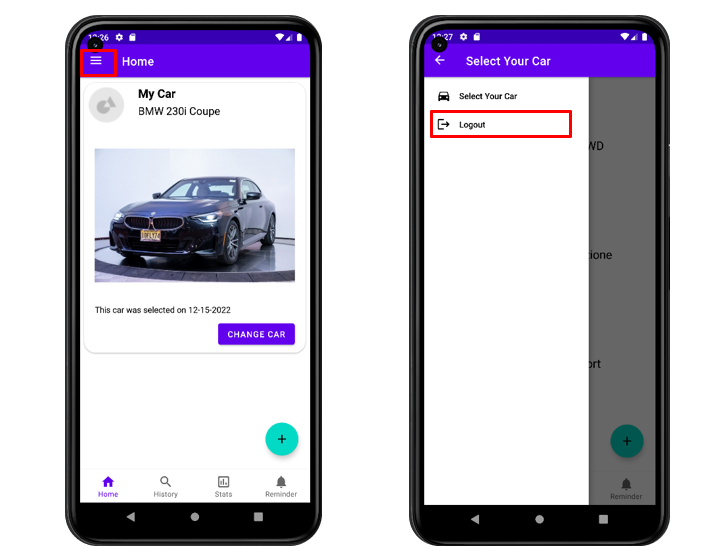
**2) Features**

**(1) Login / Register / Logout**

You may register or log in with your email and password. (ex. test@gmail.com / test123). The login/Register process relates to Firebase Authentication.



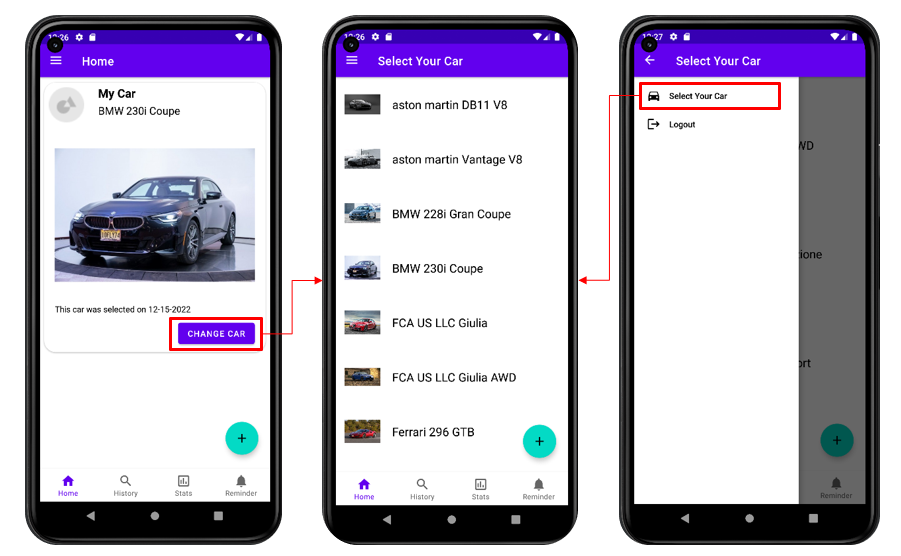
If Firebase Authentication failed, you can click the "forgot password" link on the Login page or the "Can't register" link on the Register page to see the Main page.



You can find the logout button by pressing the button on the upper left of the app.

**(2) Select Your Car (Home Menu)**

There are 50 cars you can choose from. You may change your car by clicking the "Change Car" button or "Select Your Car" in drawer menu.



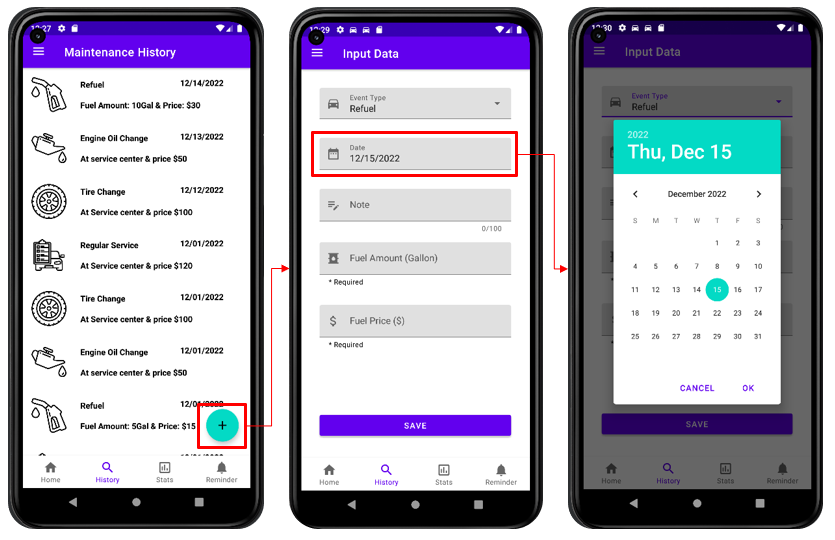
**(3) History Menu**

If there is no history data, you can see "ADD DUMMY DATA" button. You may add dummy data for test by clicking the "ADD DUMMY DATA" button.

텍스트이(가) 표시된 사진

자동 생성된 설명

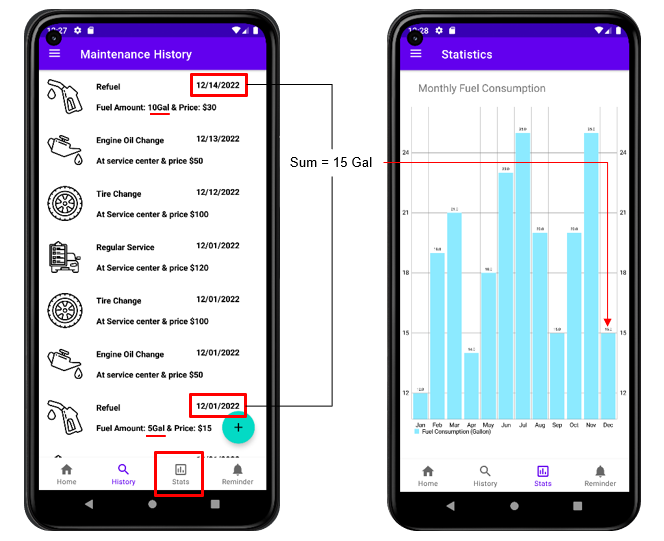
You can edit or delete the car maintenance history by clicking the item.



You can enter your new car maintenance history (ex., refuel, change engine oil, change a tire, get regular service) by clicking the floating action button (+) on the lower left.

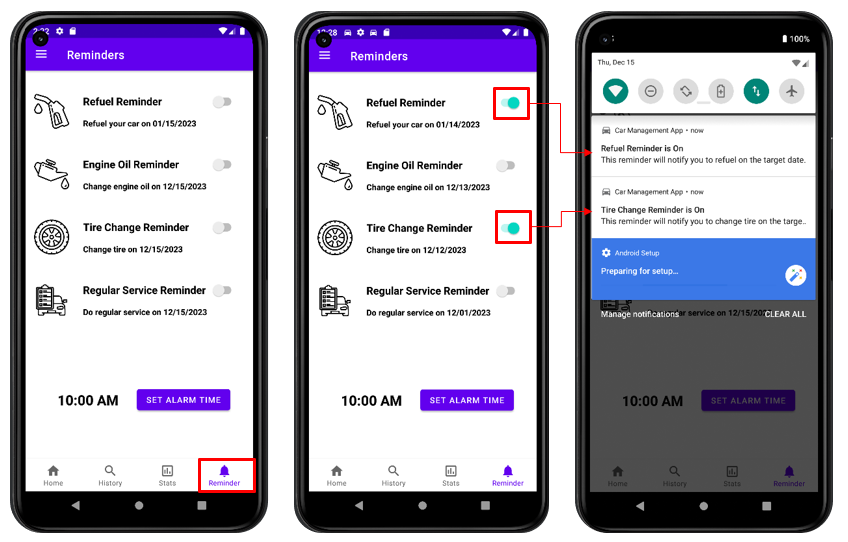
**(4) Stat Menu**

You can see the current year's monthly fuel usage. The app sum up the each month's fuel consumption amount.

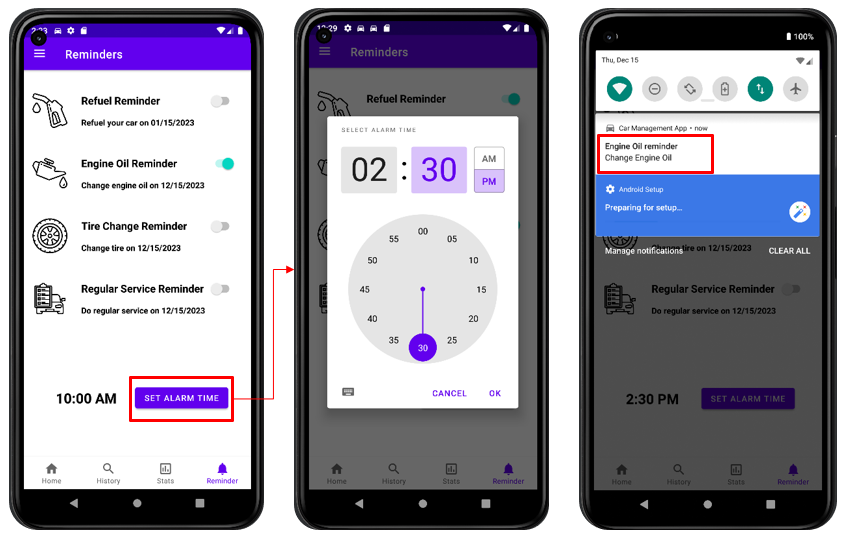


**(5) Reminder Menu**

The app calculates the recommended next date for the car maintenance action (ex., refuel, Change Engine Oil, Change Tire, Do regular service) and sets the reminder for the target date and time.



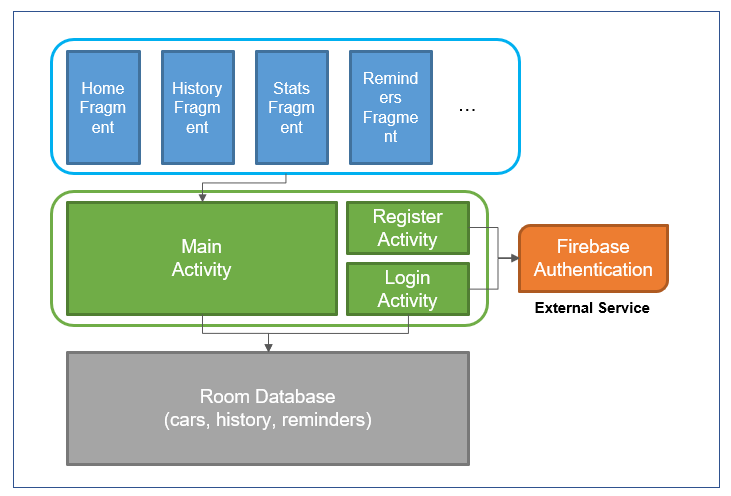
You may turn on/off the reminder and alter the reminder's alarm time.



**2. Application Design**

**1) High-level design**

When events occurred, each fragment was replaced through Main Activity. User data was stored in the app's Room database, and shared preference was used for simple preference. Login and Registration Activities were linked to Google Firebase.

****

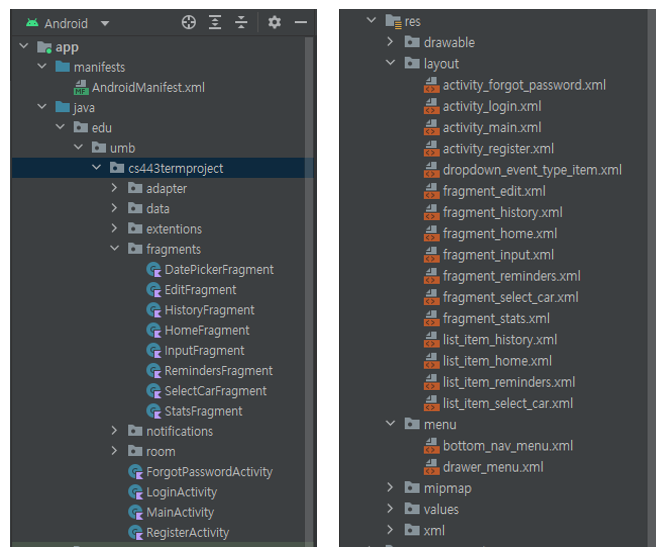
**2) Target Android API & Test Emulator**

* Target Android API: 32 (min API 26)
* Test Emulator: Pixel 5

**3. Application Implementation and Evaluation**

**1) App Implementation (source structure)**

We implemented the app by dividing the package according to each usage such as adaptor, data, fragments, notification, room, etc.



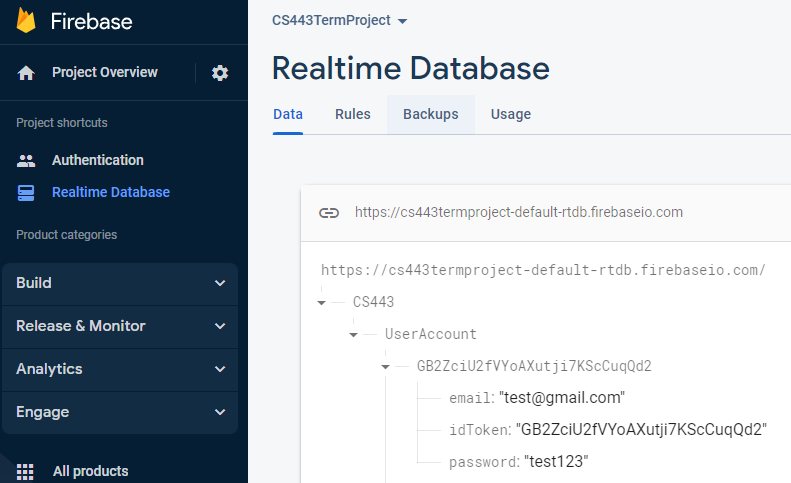
**2) Test and Problems**

The test was conducted through an Android emulator. It was time consuming because it was manually tested each time the source was changed. Using JUnit to automate testing would have saved more time. Numerous bugs were found during the test, especially bugs according to the Android version. There was not enough time to test the app for all Android versions.

**4. References**

**1) Firebase Authentication**

* https://firebase.google.com/docs/auth
* https://firebase.google.com/docs/auth/android/email-link-auth

****

**2) Bar Chart (MPAndroid Chart v3.1.0)**

* https://github.com/PhilJay/MPAndroidChart

**5. Experiences and Thoughts**

**1) Improvements**

Only 50 car information was used here. The actual app will have to deal with most commercial cars. It should also be easy for users to search for models of their own cars. In the statistics section, it would be more useful to show graphs from various perspectives on vehicle management history as well as monthly fuel usage.

**2) Difficulties in developing**

First, it was difficult to collect data related to cars that we wanted. And as features were implemented, the necessary parts continued to be found. From a user's point of view, we wanted to provide more useful information such as fuel usage efficiency and car parts efficiency of the car, but it was difficult to implement due to time constraints. In addition, to provide more appropriate services, it is necessary to build a back-end server that can manage users' data and update car information from time to time.