

# Final Report for Term Project

## Gachon Lock



Team-7

소프트웨어공학

조용준

김민걸

최경호

장윤영

## **<Index>**

- 1. Introduction**
  - 1.1 Motivation**
  - 1.2 Scope of the system**
  - 1.3 Objectives and success criteria of the project**
  - 1.4 Technical skill**
- 2. Proposed system - Gachon Locker**
  - 2.1 Overview**
  - 2.2 Requirements analysis**
    - 2.2.1 User Requirements**
    - 2.2.2 System Requirements**
    - 2.2.3 Main Function**
    - 2.2.4 System stakeholders**
    - 2.2.5 UseCase Diagram**
  - 2.3 System modeling**
    - 2.3.1 Activity Diagram**
    - 2.3.2 Sequence Diagram**
  - 2.4 Architectural design**
    - 2.4.1 Layered architecture**
    - 2.4.2 Pipe and filter architecture**
- 3. Validation**
  - 3.1 Verification**
  - 3.2 Validation**
  - 3.3 Debugging**
- 4. Glossary**
- 5. Review**

# 1. Introduction

'Gachon Locker' is a locker application created for the convenience of students' locker applications. The city application was made for students of one department, and it was made so that students could easily apply for a locker as if they were making a reservation for a movie theater.

## 1.1 Motivation

- Currently, how to apply for a locker in the Department of Industrial Engineering - Applying for a locker using Naver Cafe
- How to apply for locker at the Faculty of Software - Apply for locker using Google Form

2022년도 모든 신입생 및 재학생을 대상으로 사물함 신규신청을 받습니다.

기간 : 무기한

원하시는 사물함 위치를 확인 후 댓글로 신청 바랍니다.  
(첨부파일에 숫자가 쓰여있는 사물함이 신청 가능한 사물함입니다.)

(1) 신청방법 : 학번 / 이름 / 사물함번호를 댓글로 써주세요.

ex) 22학번 홍길동 A-1

(2) 변경방법 : 학번 / 이름 / 사물함번호 → 새 사물함번호 (변경)을 댓글로 써주세요.

ex) 22학번 홍길동 A-1 → A-2(변경)

- 확인에 어려움이 있어 댓글 수정과 삭제는 불가능합니다.

\* 사물함은 신청 완료되면 바로 사용 가능합니다.

\* 연장을 못 하신 분들도 신규신청과 같이 원하시는 사물함 자리를 댓글로 신청 해주시면 됩니다.

\* 노란색 사물함이 현재 사용 중인 신청 불가능한 자리이고 숫자가 쓰여있는 자리가 신청 가능한 자리입니다.

\* 신청하러는 사물함이 타인과 겹치지 않도록 댓글 확인 후 신청해주시기 바랍니다.

\* 공정성을 위해 댓글의 시간 및 순서를 철저히 따지겠습니다.

여러분의 무거운 가방의 짐을 덜어드릴 사물함을 신청받습니다 📦

✅ 신청 날짜 : 4월 1일 금요일 18시  
(선착순마감)

✅ 신청 방법 : 구글 폼으로 전공, 이름, 학번, 전화번호 제출  
(소프트웨어, 인공지능 구글폼 다른 점 주의 부탁드립니다!)

※구글 폼은 신청 당일날 카톡방에 공지 해드릴 예정입니다.

✅ 전공별 사물함 갯수 : 인공지능 100개 / 소프트웨어 107개

✅ 사용 기한 : 4월 3일 일요일 부터 ~ 2022년 6월 21일까지

【주의사항】

1. 선착순으로 배정될 예정으로, 신청이 조기 마감될 수 있습니다.

2. 신청 기간 이외 시간에 신청된 사물함은 무료 처리되오니 시간에 꼭 맞춰 신청해주세요.

As above, there is currently no locker application method set at the school, and each department is receiving locker applications according to the situation of their department. It is very inconvenient and cumbersome as there is a high possibility that the student's application information will be omitted in the process of taking over the locker application information from the previous year, as it is necessary to create a new Google form. This creates a motivation for the locker application application that unifies and facilitates locker applications.

To summarize the inconveniences of the current locker application method

1. Inconsistent locker application system
2. Inconvenient, high probability of missing information
3. The hassle of moving information every year the student council changes
4. Each person must update the locker application information

## **1.2 Scope of the system**

The scope of the system is intended for department students. The main customers of the system are department students. The administrator of the system is the student council president or the student council team leader. We need to create an application using Android Studio and make it available to students.

<Schedule>

4/26 - Select a topic, write a proposal

5/3 - Requirements clearance, system modeling, prototyping

5/10 - Get all the information you need for architectural design, design and implementation

5/17 - design and implementation

5/24 - design and implementation

5/31 - Last error check, Software testing

## **1.3 Objectives and success criteria of the project**

### **1.3.1 Objectives of the project**

- ① Allows you to conveniently register a locker through a mobile app.
- ② Locker register are stored on the server and must be saved in real time.

### **1.3.2 success criteria of the project**

- ① It should be possible to conveniently apply for a locker through a mobile phone.
- ② It should be possible to visually check which lockers can be selected in real time
- ③ It can only be used by those who belong to their own department.
- ④ It should be accessible to users' information.

## **1.4 Technical skill**

- ① Using android studio
- ② Using JAVA
- ③ Using XML
- ④ Building a database system using Firebase

## **2. Proposed system - Gachon Locker**

### **2.1 Overview**

'Gachon Locker' is a locker application application conceived by the inconvenience of students and administrators regarding locker applications. In order to apply for a locker, students additionally sign up for a cafe, check the locker they want in an Excel file, and request a locker by commenting or using the Google form.

In this case, students feel cumbersome and cumbersome when applying for lockers.

In the case of administrators, when students apply for lockers through comments or Google Forms, it is cumbersome because it is necessary to check whether each person has applied for a locker duplicately, and to update the information every semester.

'Gachon Locker' is a system that allows both students and administrators to conveniently apply for and manage lockers.

### **2.2 Requirements analysis**

#### **2.2.1 User Requirements**

- ① It should be possible to conveniently apply for a locker through a mobile phone.
- ② It should be possible to visually check which lockers can be selected in real time.
- ③ It can only be used by those who belong to their own department.
- ④ It should be accessible to users' information.

### 2.2.2 System Requirements

- ① It should be possible to conveniently apply for a locker through a mobile phone.
  - Because it has to be made into a mobile application, we use languages and tools that can be used on mobile.
  - The UI is designed intuitively because it should be comfortable for users to use.
  
- ② It should be possible to visually check which lockers can be selected in real time.
  - Like the movie reservation system, you have to show a locker that can be requested visually.
  - It should be possible to check the application information in real time by linking with the database.
  
- ③ It can only be used by those who belong to their own department.
  - To apply for a locker, log in by entering your Google ID.
  - Students' information should be updated in the database.(firebase).
  
- ④ It should be accessible to users' information.
  - Administrators can check students' locker application information in the database.
  - Invite an administrator to Firebase to check student information.

### 2.2.3 Main Function

- ① Login
  - : Login using the student's Google ID in the database.
  
- ② Locker GUI
  - : Shows the GUI for the locker and allows you to apply.
  
- ③ Real – time locker reservation
  - : Locker application information should be updated in real time.
  
- ④ Manage Student's Information
  - : The system manager sets the locker application period.

## 2.2.4 System stakeholders

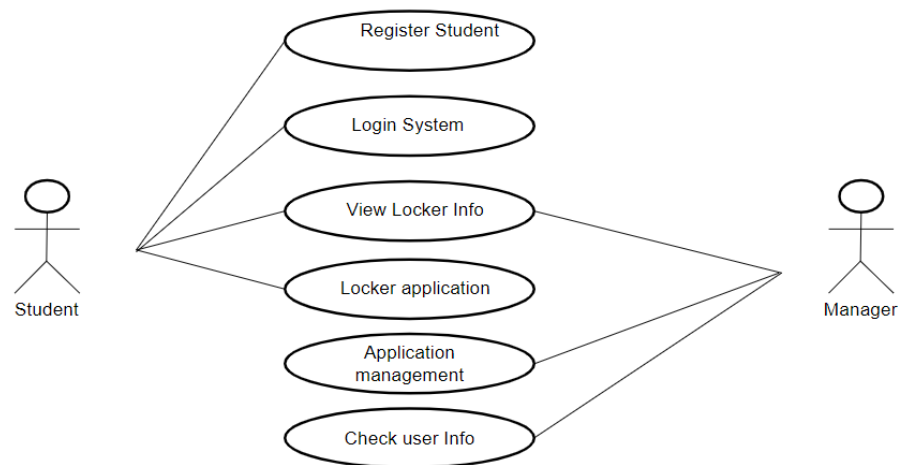
① Student

: Department students whose information is recorded in the system

② Student president(manager)

: Student council president who can manage student information and set the application period

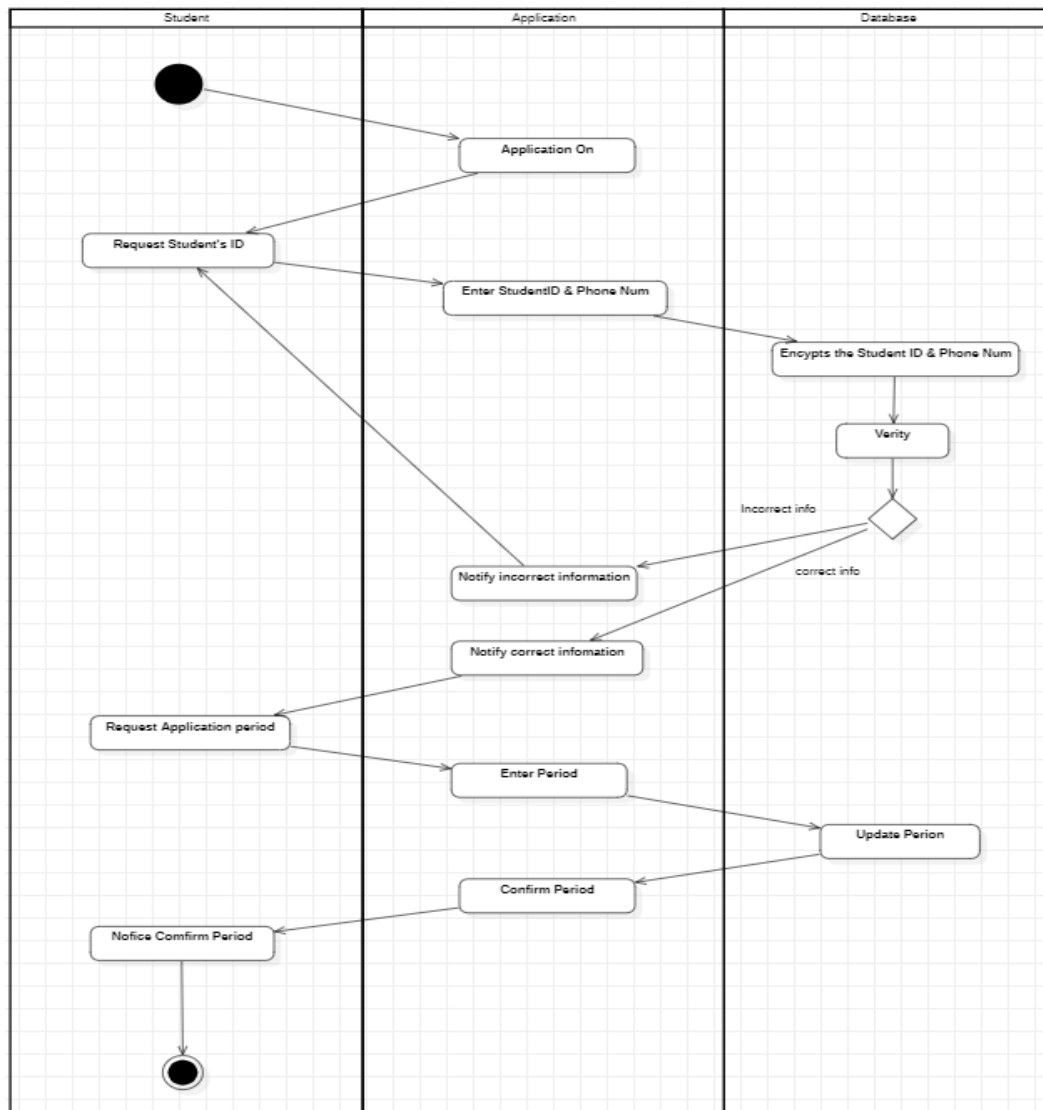
## 2.2.5 UseCase Diagram



Usecase	Description
Register Student	Students' school webmail (Google ID) receives information from the departmental archives and stores it in the database in advance, and students who have not signed up for webmail must register.
Login System	If you log in with students' Google IDs, you will be logged in if the students are stored in the database, but you will not be able to log in if your ID and password are incorrect or you are not a student.
View Locker info	Like the movie theater reservation system, it is impossible to touch a locker that has already been replaced, and only lockers that can apply for it can be touched.
Locker application	When a student applies for a locker, information is saved in the database in real time
Application management	The administrator can check the locker application number and application status of students on the management server
Check user info	You can access the database server and check student information.

## 2.3 System modeling

### 2.3.1 Activity Diagram

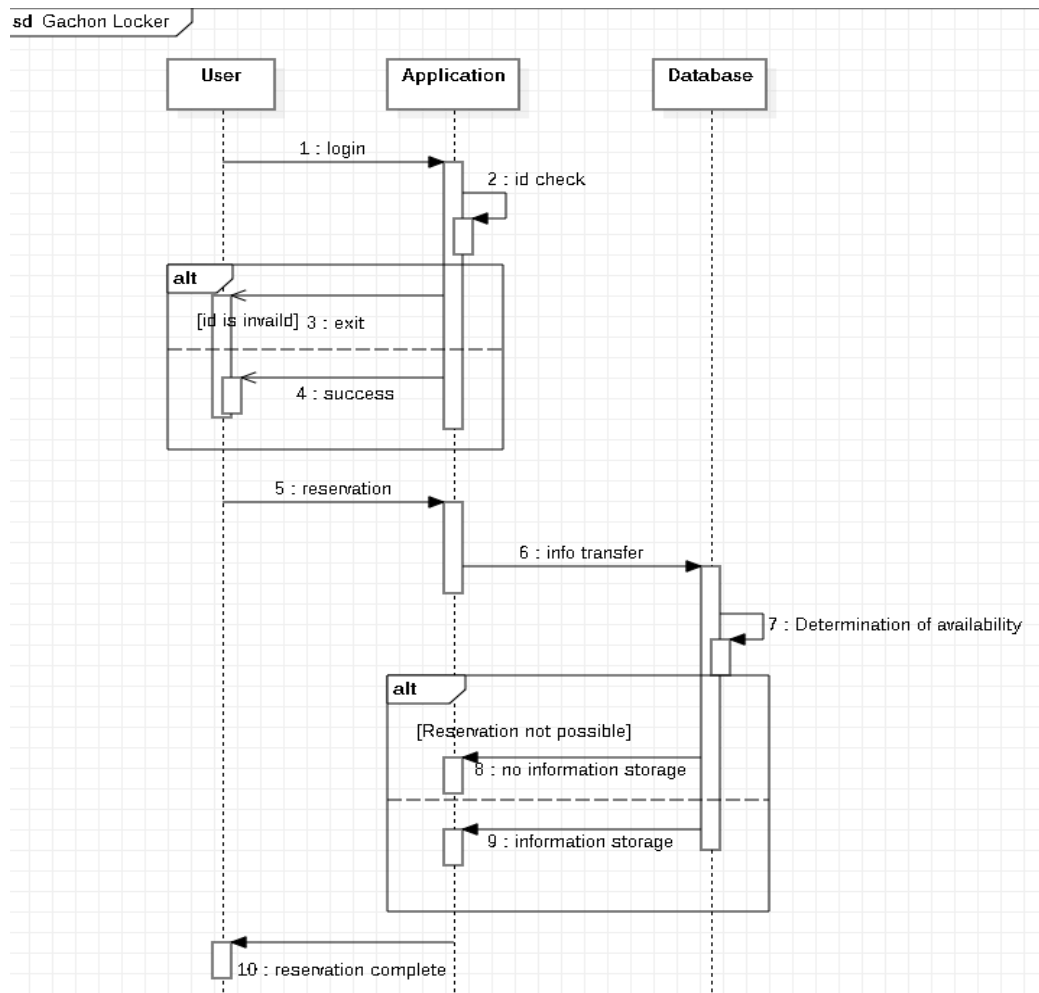


Through the activity diagram, the logic for students to apply for lockers is checked in order according to the processing flow according to the conditions.

The student opens the application and logs in. After logging in, if the locker application period is set, it is the process of requesting the desired locker.



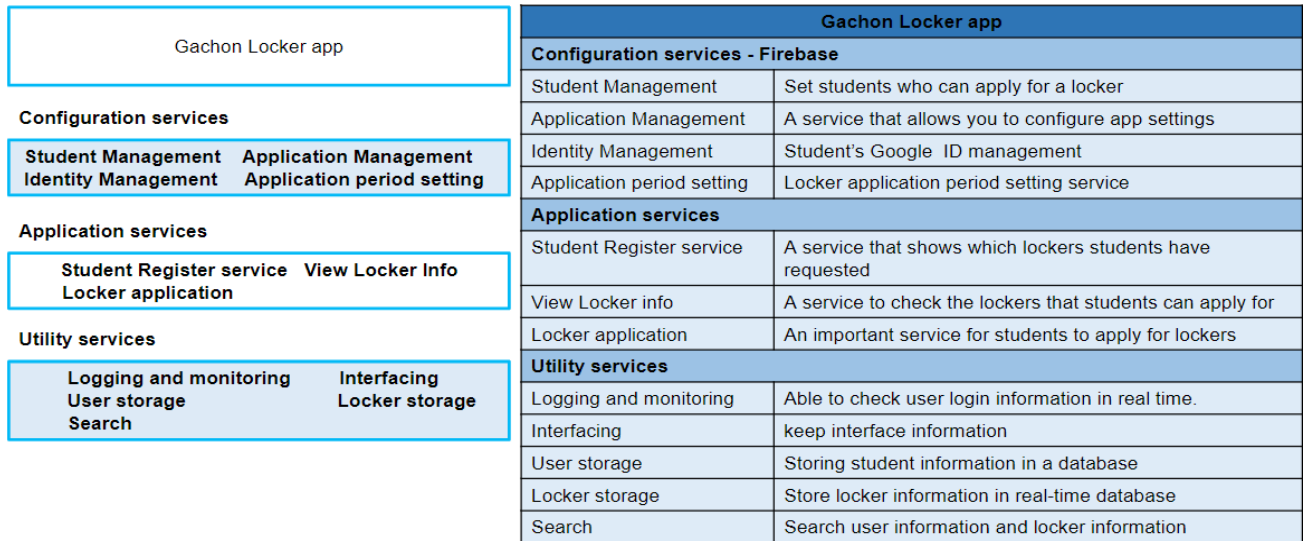
### 2.3.2 Sequence Diagram



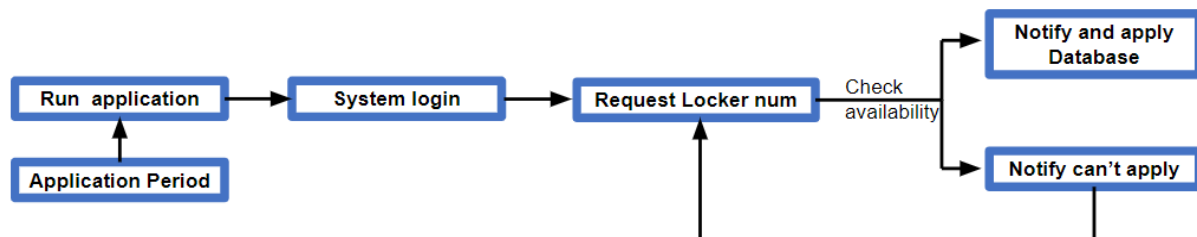
We redefine the objects of the use case diagram described earlier, and draw an activity diagram that shows interactions. Designates users, applications, and databases as active objects, and shows how information interacts when a user requests a locker. When you log in and apply for a locker, you store information with the database and year, and send a message to the student.

## 2.4 Architectural design

### 2.4.1 Layered architecture



### 2.4.2 Pipe and filter architecture



Description	
Application Period	The manager announces the locker application period
Run application	The student launches the application
System login	Only students can apply, so you need to log in.
Request Locker num	Shows locker information graphically
Notify and apply Database	If it is a locker that can be applied, a confirmation message is sent and stored in the database
Notify can't apply	If it is not possible to apply for a locker, a warning is issued and the applicant is asked to apply again.

## 3. Validation

### 3.1 Verification

You can make reservations using the app, and the UI is intuitive. You can check the reservation information in real time.

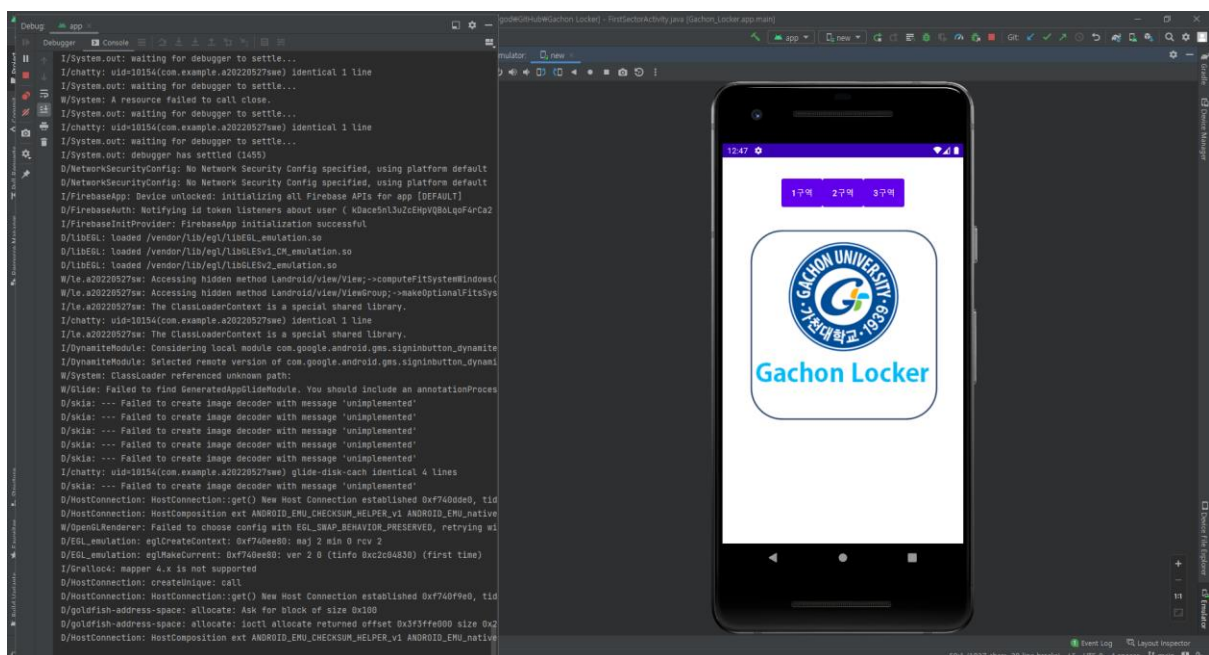
in conclusion use JAVA, XML, intuitive, Real-time reservation available, manage permissions with student Information data, admin can manage reservations

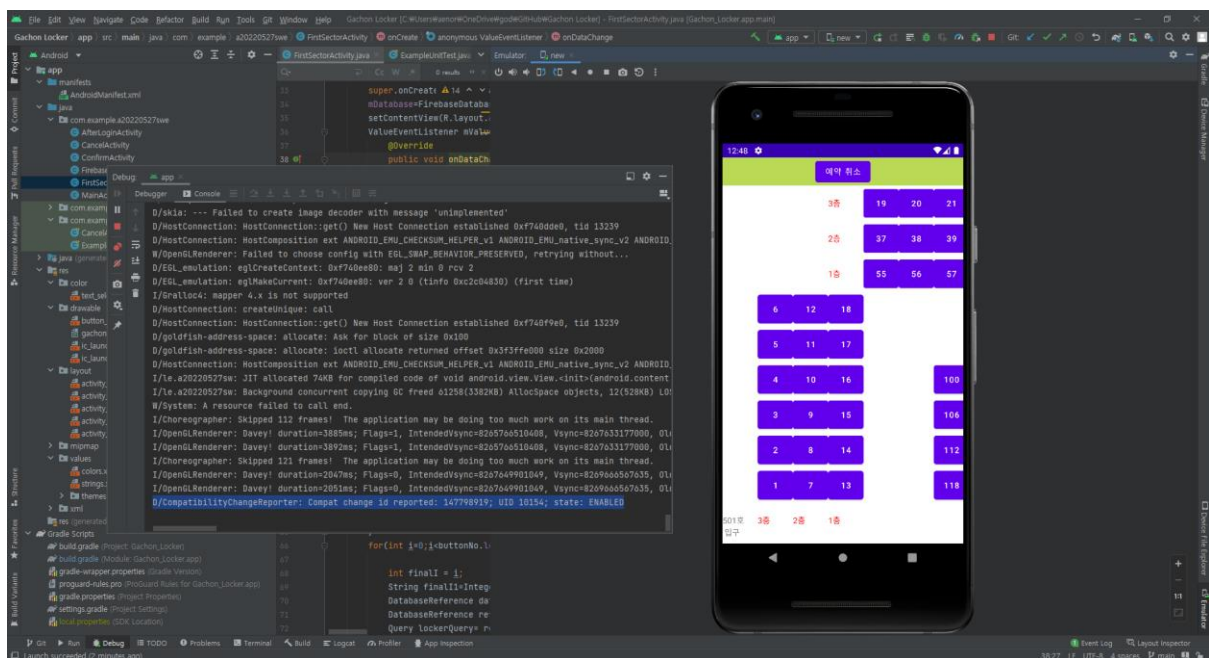
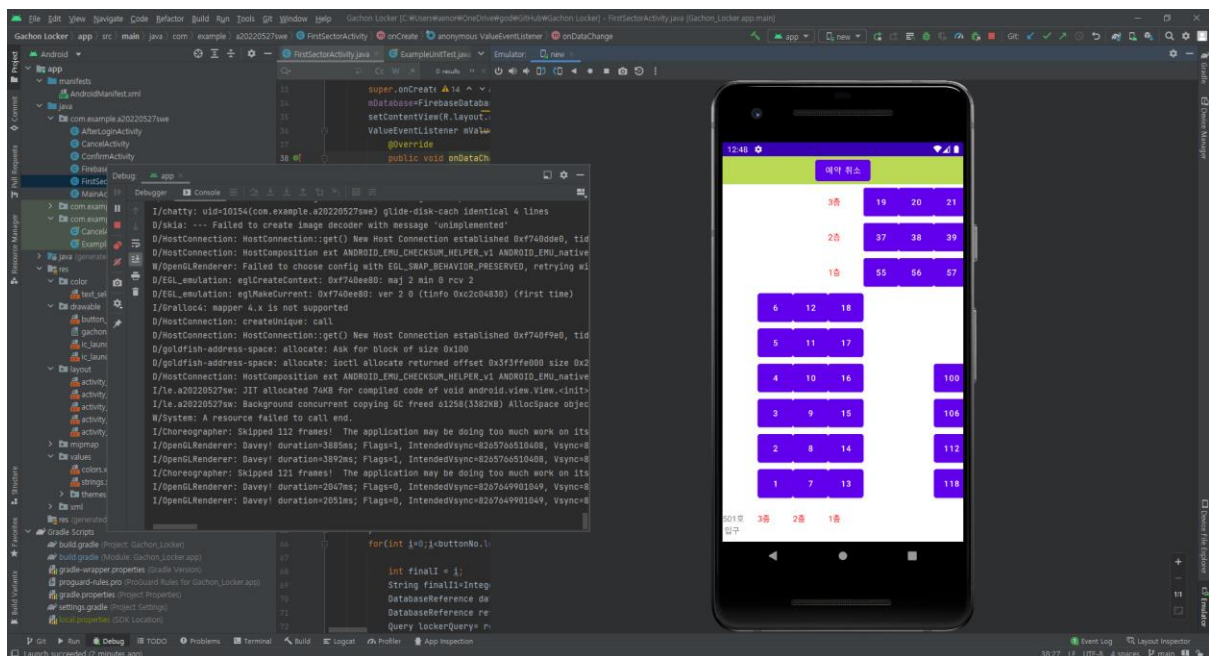
However, there may be duplicate reservations because the reserved area is not displayed, and anyone can make reservations because the user's information is not in the database.

### 3.2 Validation

I received an application through the comments of the existing cafe or applied using Google Form, but I applied easily through the app.

### 3.3 Debugging

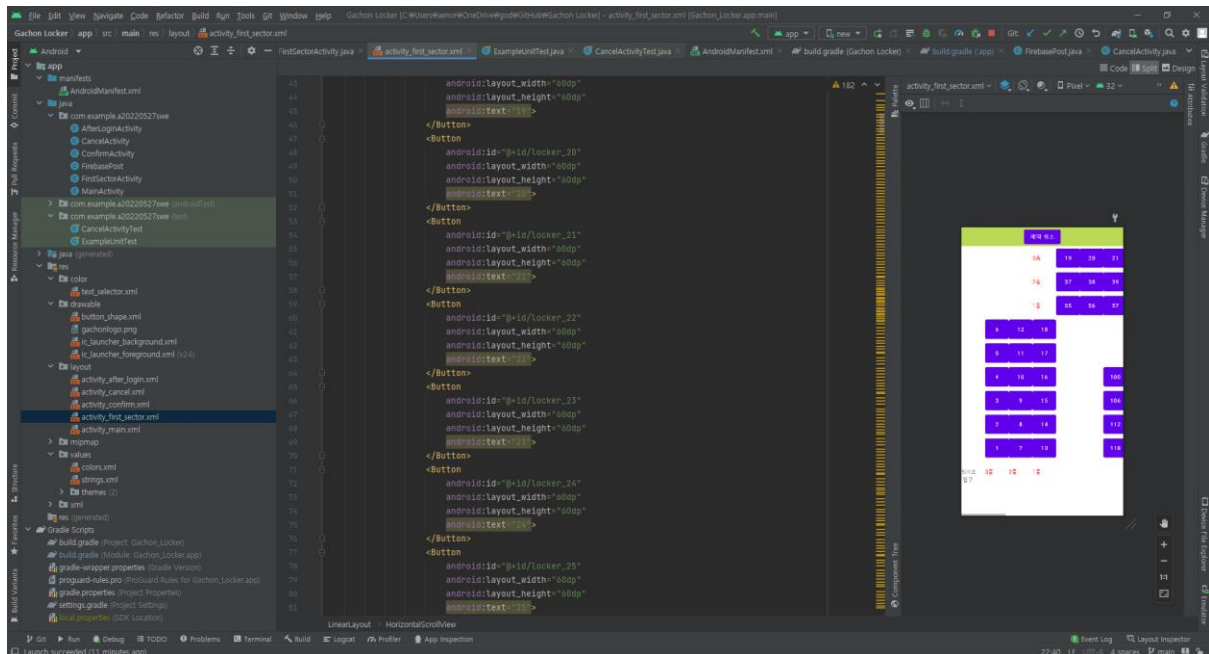




Found out that registering is well done.

Use debugging to correct the wrong code each time

## 4. Glossary



We made all the buttons in xml files. So we assigned all the address to the buttons with for loop below.

```
Button[] buttonNo=new Button[147];
for(int i=0;i<buttonNo.length;i++){
    String buttonId="locker_"+(i+1);
    buttonNo[i]=findViewById(getResources().getIdentifier(buttonId, defType: "id",getPackageName()));
}
}
```

```
buttonNo[i].setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(lockerQuery.toString()==finalI){
            Toast.makeText(getApplicationContext(), text: "이미 예약된 사물함입니다.",Toast.LENGTH_SHORT).show();
        }else{
            Intent intent=new Intent( packageContext: FirstSectorActivity.this,ConfirmActivity.class);
            intent.putExtra( name: "data", value: (finalI +1)+"라커를 예약하시겠습니까?");
            intent.putExtra( name: "LN", value: finalI+1);
            startActivity(intent);
        }
    }
});
```

If lockerQuery which has the value of the locker number in the realtime database equals to the clicked locker's number, it means that the locker is being used and Toasts a message.

Else it moves to Popup page for confirmation to register.

```
public void postFirebaseDatabase(boolean add){
    mPostReference = FirebaseDatabase.getInstance().getReference();
    Map<String, Object> childUpdates = new HashMap<>();
    Map<String, Object> postValues = null;
    if(add){
        FirebasePost post = new FirebasePost(ID, name);
        postValues = post.toMap();
    }
    childUpdates.put("/id_list/" + ID, postValues);
    mPostReference.updateChildren(childUpdates);
}
```

postFirebaseDatabase is a function that keeps the form of the data for writing it on the realtime database.

**Github URL** - <https://github.com/jangyyoung/20220527SWE.git>

## 5. Review

Our group made an app based on the needs of real team members in this term project. For this, we built a database and debugged it. pity that I couldn't handle Junit, but We planned and tested it with various system modeling and architecture design. It was a proud semester as it will be helpful for logical planning in future term projects.