

[2019년 1학기 실전프로젝트2 설계 보고서]

출석태깅 알람어플 '잊지말자 출책'

제출자: 김혜영

제출날짜: 2019-05-27

지도교수: 장소연 교수님

목차

1. 문제 정의

- A. Problem Statement
- B. Definition Input/Output
- C. Objective
- D. Constraints
- E. Function

2. 배경이론 및 기존 방법들

- A. 기존 어플리케이션 구현 방법들
- B. 한동대학교 학생들에게 서비스를 주는 기존 제품

3. 제안된 문제접근방법

- A. Key Approach
- B. System Design
- C. System Environment
- D. Function

4. 실험결과

- A. '잊지말자 출책!' 앱 실제 구현 모습
- B. 구현 및 그에 따른 실험 방법

5. 결론

- A. 결과를 바탕으로 한 요약
- B. 향후 과제

6. 참고문헌

1. 문제 정의

A. Problem Statement

한동대학교에서 출석체크를 NFC 태깅으로 하는 경우가 많다. 그러나 교수님들과 학생들이 매 수업시간마다 챙기기엔 번거롭고, 까먹는 경우엔 출석을 놓치는 경우가 생긴다. 이러한 불이익을 방지하기 위해 유저가 설정해 둔 수업시간 시작 몇 분 전에 미리 스마트폰 화면에 알리를 띄워 줌으로써 유저가 태깅을 까먹지 않고 출석체크를 할 수 있도록 돕는다.

B. Definition Input/Output

- Input
 1. 크롤링으로 데이터를 입력받는 부분
 - 유저의 Hisnet 로그인 정보(아이디, 비밀번호)
 - 해당 아이디의 수업 시간표 설정
 2. 유저의 입력으로 데이터를 얻는 부분
 - 알람 on/off
 - 시간표 색상 설정
 - 알람 시간
 - 유저의 알람 선택 저장 값
 - 메뉴 선택(Hisnet 로그인, 전체 삭제, 설정)
- Output
 1. 유저의 수업 시간표 출력(과목명, 수업장소, 메모 등)
 2. 기기에 따른 팝업 창
 3. NFC 태깅시 '스마트 캠퍼스' 어플 설치 여부에 따른 화면 이동
 - 설치되어 있지 않은 경우: 구글 스토어 설치화면으로 이동
 - 설치되어 있는 경우: 어플리케이션 실행 및 NFC 태깅화면 이동
 4. 알람 on이 되어 있는 과목에 아이콘 표시

C. Objective

이 어플리케이션을 통해 많은 새내기들과 한동대학교의 학우들이 출석태깅 시스템에서 실수를 통해 얻는 불이익을 예방 및 최소화 할 수 있도록

도와주는 것을 목표로 둔다.

D. Constraints

1. 한동대학교 학생들만 이용 가능하다.
2. '스마트 캠퍼스' 어플리케이션을 설치해야만 한다.
3. '스마트 캠퍼스' 어플리케이션을 통해서만 태깅이 가능하다.

E. Function

1. Hisnet으로부터 자신의 시간표(과목명, 교수명, 강의실 등)에 대한 정보를 가져올 수 있도록 한다.
2. Hisnet으로부터 가져온 시간표에서 추가적인 수정이 가능하도록 한다.
3. 과목마다 알람 on/off 기능을 설정할 수 있도록 한다.
4. 알람이 켜져 있는 시간표는 아이콘을 통해 별다른 실행없이 확인이 가능하도록 한다.
5. 한동대학교 지도에 건물 위치 표시 및 건물 이름에 대한 정보를 제공한다.

2. 배경이론 및 기존 방법들

A. 기존 어플리케이션 구현 방법들

HTML5, JQuery, MySQL을 이용하여 어플리케이션 개발.

B. 한동대학교 학생들에게 서비스를 주는 기존 제품

- 보완이 필요한 점

1. 출석태깅과 같은 특정 이벤트에 대한 알림 서비스가 없음.
2. 학교에 익숙하지 않은 신입생들에게 강의실 위치, 건물 이름을 쉽게 알려주는 서비스가 없음.
3. 한동스마트앱(학교 NFC 기능을 활용하는 앱) 기능을 쓰는 경우가 많은데 기능이 필요할 때마다 한동스마트앱을 찾아 실행하는 번거로움이 있음.

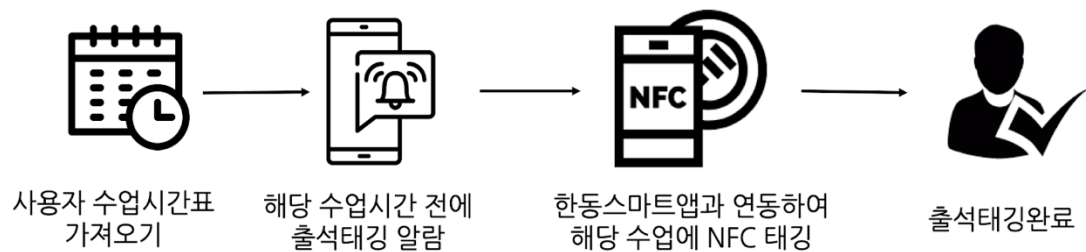
- 본 프로젝트의 해결책

1. 출석태깅 등 해당 시간표에 대한 일정에 대해 알림 서비스를 제공하는 어플리케이션 개발

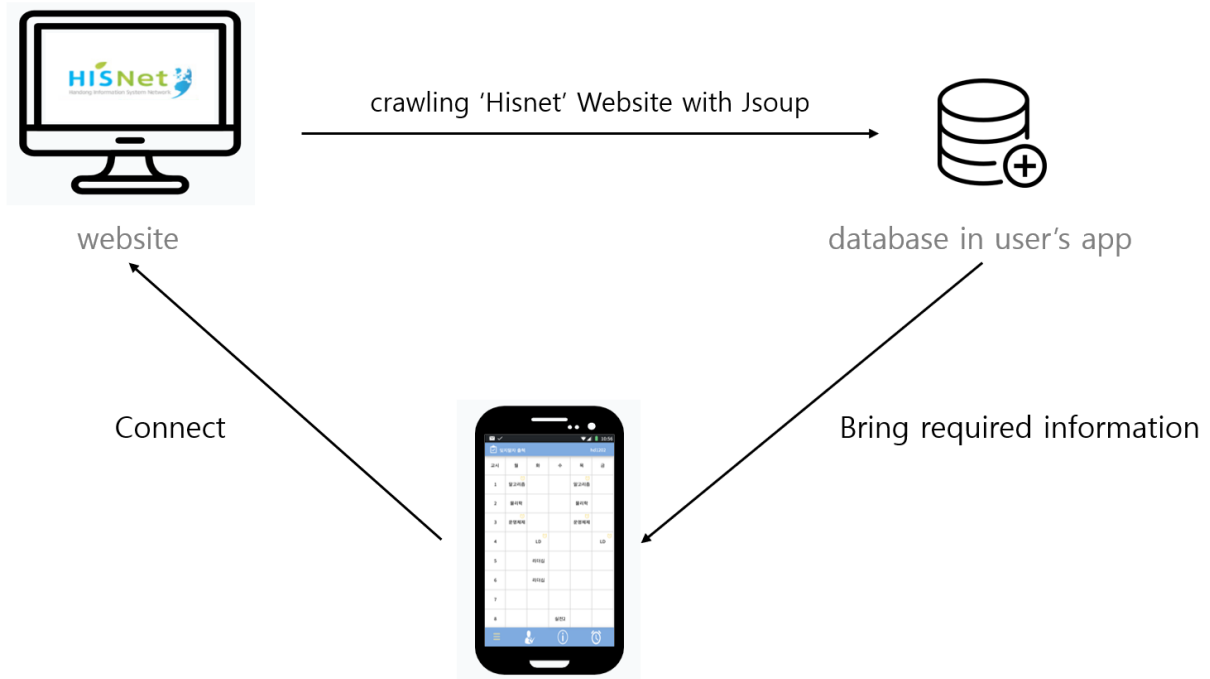
2. 학교 강의실 위치와 건물 이름을 알려주는 지도 페이지 제공
3. 한동스마트앱과의 연동 기능을 통해 바로 NFC기능을 활용 가능

3. 제안된 문제접근방법

■ Key Approach



■ System Design



Hisnet 웹사이트를 통해 jsoup을 이용하여 필요한 과목에 대한 정보만 가져온다. 이때, 가져온 데이터는 특정 서버가 아닌 유저의 어플리케이션에 있는 Database에 저장된다. 추가로 가져온 데이터를 가공하여 눈에 보기 쉬운 UI로 변경해 시간표 형태로 만들어 유저에게 보여준다. 또한, 유저가 어플리케이션을 초기에 실행할 때 로그인을 한 내역이 있으면 유저가 로그아웃을 하지 않는 이상 자동 로그인 상태가 되어, 어플리케이션을 재시작하는 경우에도 다시 로그인하는 번거로움을 줄여준다.

■ System Environment



■ Function



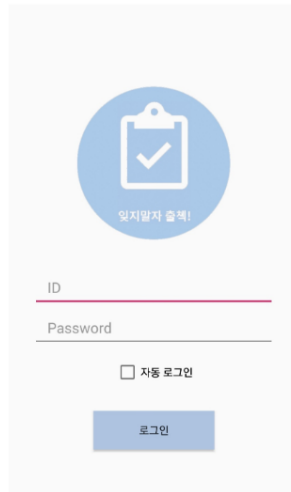
4. 실험결과

A. 앱 실제 구현 모습

1. 앱 초기 화면



<Intro 실행 화면>



<로그인 창 실행 화면>

- Intro 실행 화면

유저에게 개발 팀의 로고를 잠시 보여줌

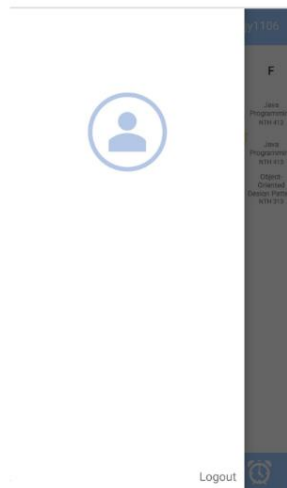
- 로그인 창 실행 화면

user에게로 부터 Hisnet id & password를 입력 받고 Crawling 진행함.

2. Main 화면 & 메뉴



<App Main 실행 화면>



<메뉴 창 실행 화면>

- APP Main 실행 화면

- (1) Hisnet으로 부터 시간표를 받아유저에게 보여준다.
- (2) 총 4개의 버튼이 있고 화면우측 상단에는 히즈넷 로그인 정보를 표기

- 메뉴 창 실행 화면

- (1) Hisnet 로그아웃, 알람 전체 설정 등 추가기능

3. 알람 버튼 & 세부설정



- 알람 버튼을 누르면

- (1) 아이콘이 저장 모양으로 바뀌고 알람을 하길 원하는 시간표의 셀을 누르면 알람 아이콘이 저장된다. 이후 하단의 저장 아이콘을 누르면 된다.

4. 태깅 버튼 & 한동 스마트 캠퍼스 앱 연결



<앱 설치가 안됐을 시>



<앱 설치가 됐을 시>



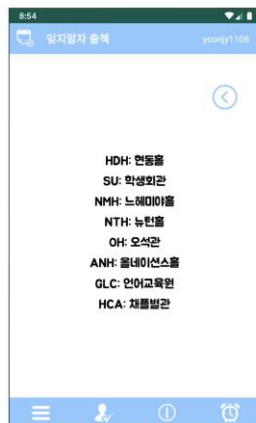
- 태깅 버튼을 누르면

- (1) 한동 스마트 캠퍼스 앱의 설치 여부를 확인 후 설치가 되어있지 않은 경우 Google PlayStore로 이동한다.
- (2) 한동 스마트 캠퍼스 앱이 설치가 되어있는 경우 바로 스마트 캠퍼스 앱이 실행 된다.

5. Info 버튼 & 한동 캠퍼스 맵



<캠퍼스 맵 화면>



<건물명 약자 표 화면>



- Info 버튼을 누르면

- (1) 한동 캠퍼스 맵을 유저에게 보여준다. (for 새내기)
- (2) 화살표를 누르면 한동대학교 건물 영어 약자 텍스트를 정리한 표를 보여준다.

6. Notification 잠금 화면 태깅 알람



<잠금화면 태깅 알람 화면>

- 태깅하러가기 버튼을 누르면

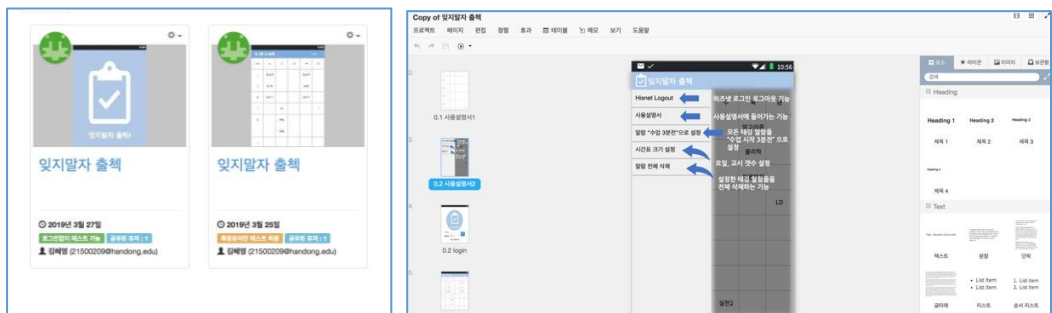
- (1) 한동대학교 스마트 캠퍼스 앱이 바로 실행되어 NFC태깅을 손쉽게 할 수 있다.

B. 구현 및 그에 따른 실험 방법

- 앱 디자인 초안

ovenApp을 기반으로 다른 여러 앱들 (인스타그램, 한동 어플, 알람 어플 등 여러가지 앱을 비교 분석 한 후 서로 프로젝트를 공유하며 적합한 스토리보드 (레이아웃 디자인 포함)를 만들어 냈다.

<작업 예시>





- 레이아웃 구현

스토리 보드 작성한 것을 기반으로 실제 디자인을 구현하는 작업이었으며, codelab에서 학습한 것을 기반 및 구글링을 통한 여러 포스팅 글을 통해 레이아웃 디자인을 구현할 수 있었다. 레이아웃 구현 중 MapFragment는 MainActivity 안의 container에서 띄워 구현을 진행하였다. DialogActivity는 Dialog 창을 Broadcast Receiver를 상속한 클래스(AlarmReceiver.java)에서는 띄울 수 없기 때문에 DialogActivity라는 Activity를 따로 만들어 구현함으로써, 위의 한계점을 극복했다.

<작업 예시>



[MapFragment Layout]

[DialogActivity Layout]

- 잠금 화면시 알람 기능 구현 (노티피케이션 구현)

노티피케이션을 구현하기 위해 기본적인 레이아웃과 액티비티들을 구성하였고 각 Gird에 저장된 값의 '알람'이 목적이기 때문에 변수를 받아와 따로 시간을 계산을 해 주는 작업이 필요했다.

우선 Crawling을 통해 가져온 데이터들을 파싱하고 그 파싱한 데이터를 DB에 넣었다. DB에서 저장된 데이터에 따라 알람 시간을 셋팅을 하는 작업을 했는데, 이는 알람 어플이라 사용자가 설정해놓은 시간표에 의 수업시간에 따라 알람이 울려야 하기 때문이다. 때문에 특정 형식으로 크롤링 된 데이터들을 데이터 parsing을 거쳐 어레이 리스트에 저장을 해주고 이것을 기반으로 알람 설정에 필요할 날, 시, 분, 초 등을 period(교시) 별로 변수를 만들어 시간을 설정해 주고 setAlarm이라는 함수를 만들어 calender 메소드를 이용하여 변수에 값을 저장 후 알람 시간을 세팅해 주었다. 그 다음 바로 broadcast Reciever(AlarmReceiver.java)에 intent로 전달을 해서 알람여부 (isAlarm) 까지 check를 해주고 activity를 해당 시간에 띄워주는 액티비티를 추가로 만들었다.

이를 통해 크롤링을 통한 데이터를 바탕으로 알람을 구현을 완료했다.

<작업 예시>

(Alarm.java) - DB에서 받은 시간표 정보를 활용해 알람을 설정한 후 intent로 AlarmReceiver.java에 전달한다.

```
package com.example.yun.crawlalarm;

import android.app.AlarmManager;
import android.app.PendingIntent;
import android.content.Context;
import android.content.Intent;
import java.util.Calendar;

public class Alarm {
    private static Context context;
    private static int alarmTime = 0; // 알람 시간
    private static int hour = 0; // 알람 시간
    private static int minute = 0; // 알람 시간
    private static int second = 0; // 알람 시간

    public static void setAlarm(Context context, int hour, int minute, int second) {
        context = context;
        AlarmManager alarmManager = (AlarmManager) context.getSystemService(Context.ALARM_SERVICE);
        Intent intent = new Intent(context, AlarmReceiver.class);
        PendingIntent pendingIntent = PendingIntent.getActivity(context, 0, intent, PendingIntent.FLAG_UPDATE_CURRENT);
        alarmManager.setExact(AlarmManager.RTC_WAKEUP, Calendar.getInstance().getTimeInMillis() + (hour * 60 * 60 * 1000) + (minute * 60 * 1000) + (second * 1000), pendingIntent);
    }
}
```

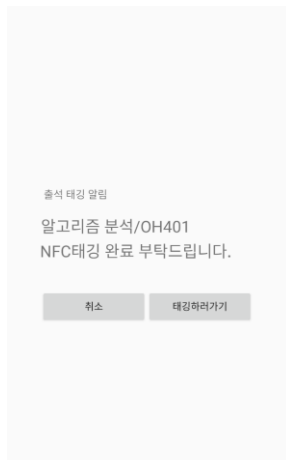
(AlarmReceiver.java) – 알람여부처리를 하지 못해 바로 DialogActivity창을 방지하기 위해 만든 클래스이다. 알람여부처리를 해준 뒤 그 결과에 따라 dialogActivity를 보낼지 말지 결정한다.

```

1 package com.example.yun.caalarm;
2
3 import android.app.PendingIntent;
4 import android.content.BroadcastReceiver;
5 import android.content.Context;
6 import android.content.Intent;
7 import android.util.Log;
8
9 public class AlarmReceiver extends BroadcastReceiver {
10
11     public void onReceive(Context context, Intent intent){
12         int alarmId = intent.getIntExtra("id", defaultValue -1);
13         int isAlarm = intent.getIntExtra("isAlarm", defaultValue -1);
14
15         if(alarmId == -1){
16             Log.i("tag", "AlarmReceiver intent error");
17             return;
18         }
19
20         if(isAlarm == -1){
21             Log.i("tag", "Error", "Don't get isAlarm");
22             return;
23         }
24
25         if(isAlarm == 0){
26             return;
27         }
28         else if(isAlarm == 1){
29             Intent activityIntent = new Intent(context, DialogActivity.class);
30             PendingIntent pendingIntent = PendingIntent.getActivity(context, REQUEST_CODE, activityIntent, PendingIntent.FLAG_ONE_SHOT);
31
32             try{
33                 pendingIntent.send();
34             }
35             catch(PendingIntent.CanceledException e){
36                 e.printStackTrace();
37             }
38         }
39     }
40 }

```

(실행화면) - 유저 시간표의 과목에 해당하는 요일과 시간이 되면 DialogActivity 알람이 울린다.



- DB 설계

User의 시간표 정보를 app을 종료시켜도 남아있도록 하기 위해 sqlite를 사용하여 데이터베이스(DB)에 데이터를 저장했다

DB를 연동하기 위해 기본설정, Table만드는 함수(CREATE), Data를 삽입하는 함수 (INSERT), Data를 검색하는 함수 (SEARCH), Data를 삭제하는 함수(DELETE), Data를

수정하는 함수(UPDATE)를 구현함으로써 크롤링으로 받은 User의 시간표 정보를 쉽게 저장하고 수정할 수 있게 하였다.

<작업 예시>

(DB.java)

```

package com.example.yun.caslam;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

public class DB {
    static int _id = 1;

    //DB 생성
    public static void createDB(Context context) {
        SQLiteDatabase db = null;
        if (db == null) {
            db = context.openOrCreateDatabase("caslam_test.db", Context.MODE_PRIVATE, null);
        }

        db.execSQL("CREATE TABLE IF NOT EXISTS SUBJECT ("
            + "_id INTEGER PRIMARY KEY AUTOINCREMENT,"
            + "TITLE TEXT NOT NULL,"
            + "PROFESSOR TEXT NOT NULL,"
            + "PLACE TEXT NOT NULL,"
            + "STARTTIME INTEGER NOT NULL,"
            + "ENDTIME INTEGER NOT NULL,"
            + "PERIOD INTEGER NOT NULL,"
            + "RED INTEGER NOT NULL,"
            + "GREEN INTEGER NOT NULL,"
            + "BLUE INTEGER NOT NULL,"
            + ");");

        db.close();
    }

    //DB에서 데이터 가져오기
    public static void selectDB(Context context) {
        SQLiteDatabase db = null;
        if (db == null) {
            db = context.openOrCreateDatabase("caslam_test.db", Context.MODE_PRIVATE, null);
        }

        //DB에서 데이터 가져오기
        Cursor c = db.rawQuery("SELECT * FROM SUBJECT", null);
        Log.d("DB", "Cursor: " + c.getCount());
        c.moveToFirst();

        while (c.moveToNext()) {
            Log.d("DB", "Cursor: " + c.getString(c.getColumnIndex("_id")) + " " + c.getString(c.getColumnIndex("TITLE")) + " " + c.getString(c.getColumnIndex("PROFESSOR")) + " " + c.getString(c.getColumnIndex("PLACE")) + " " + c.getString(c.getColumnIndex("STARTTIME")) + " " + c.getString(c.getColumnIndex("ENDTIME")) + " " + c.getString(c.getColumnIndex("PERIOD")) + " " + c.getString(c.getColumnIndex("RED")) + " " + c.getString(c.getColumnIndex("GREEN")) + " " + c.getString(c.getColumnIndex("BLUE")) + " ");
        }
    }
}

```



- Fragment 구현

Fragment 로 구현함으로써, 밑의 메뉴들의 디자인들을 신경쓰지 않게 하였다. 그리고 하나의 Activity 의 컨텍스트 안에 있게 함으로써 Activity 낭비를 줄이고 화면 전환 구성을 쉽게 구현하였다.

<작업 예시>

```

package com.example.yun.caslam;

import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentTransaction;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.ImageButton;
import android.widget.ImageView;

public class MapFragment extends Fragment implements View.OnClickListener {
    boolean isView;
    ImageButton rightBtn;
    FragmentManager manager;
    ImageView img;

    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
        isView = false;
        View view = inflater.inflate(R.layout.fragment_map, container, false);
        manager = getFragmentManager();
        img = (ImageButton) view.findViewById(R.id.img);
        rightBtn = (ImageButton) view.findViewById(R.id.rightBtn);
        rightBtn.setOnClickListener(this);

        return view;
    }

    public void onClick(View v) {
        if (v.getId() == R.id.rightBtn) {
            isView = true;
            if (isView) {
                img.setImageResource(R.drawable.map_img);
                rightBtn.setImageResource(R.drawable.rightBtn_img);
            } else {
                img.setImageResource(R.drawable.info_img);
                rightBtn.setImageResource(R.drawable.backBtn_img);
            }
        }
    }
}

```

[MapFragment.java]

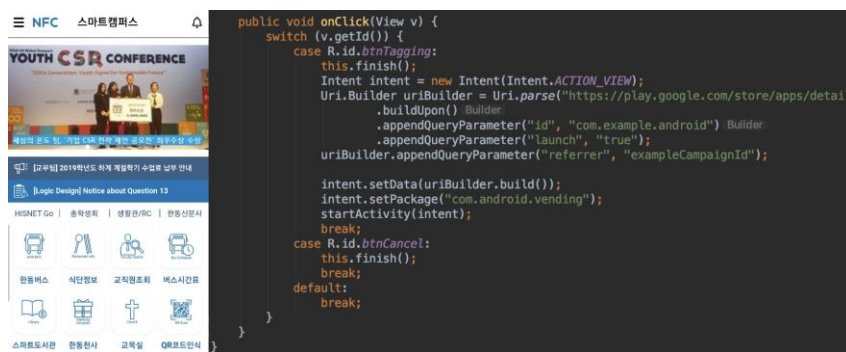


[MapFragemnt.java 실행화면]

- 잊지말자출책 앱 안에서 한동스마트 캠퍼스 앱 연동하기

유저로 부터 id , passwd를 입력받아 크롤링을 통해 데이터를 hisnet에서 가져왔다면 알람 설정 이후 잊지말자 출책 앱 안에서 한동앱으로 바로 이동 할 수 있게 하는 기능을 구현해야 했다. 이 부분에서 Uri 빌더를 통해 먼저 설치되어 있지 않을 시 구글 플레이스토어로 이동하게 하기 위해 해당 url을 parse해줬으며 해당 앱이 설치 되어있을 경우 바로 해당 intent가 실행 되겠끔 하였다.

<작업 예시>



5. 결론

A. 결과를 바탕으로 한 요약

- 사용하기 편리하고 실제 한동대학교 재학생들의 출석 태깅에 도움이 되는 앱을 개발완료

- ◆ 앱을 통해 히즈넷에서 시간표를 받아들여와 유저가 알람 설정해 둔 수업시간 시작 전에 미리 스마트폰 화면에 알림이 띄워 짐으로 유저가 태깅을 까먹지 않고 출석체크를 할 수 있도록 하였다.
- ◆ app을 설치 후 실행하면 히즈넷 아이디로 로그인을 필수적으로 하게 되어있고 이후 입력된 유저 정보를 통해 히즈넷의 시간표를 크롤링(Crawling)하여 해당 유저의 시간표 정보를 app 메인 화면에 시간표로 출력해 준다.
- ◆ 메뉴, 한동앱 태깅하러가기, Info, 알람 이렇게 4가지 기능을 담당하는 각 버

트들을 통해 유저가 원하는 수업시간에 알람 설정이 손쉽게 가능하며 한동 앱 또한 바로 연결되어 갈 수 있다.

- **출석 태깅에 있어 기존 앱의 3가지의 보완점을 해결**

(보완점)

- ◆ 출석 태깅과 같은 특정 이벤트에 대한 알림 서비스가 없음.
- ◆ 학교에 익숙하지 않은 신입생들에게 강의실 위치, 건물 이름을 쉽게 알려주는 서비스가 없음.
- ◆ 한동스마트앱(학교 NFC 기능을 활용하는 앱) 기능을 쓰는 경우가 많은데 기능이 필요할 때마다 한동스마트앱을 찾아 실행하는 번거로움이 있음.

- **개발 Objective 바탕으로 최종적으로 기능 리스트들을 모두 구현완료**

(기능)

1. Hisnet으로부터 자신의 시간표(과목명, 교수명, 강의실 등)에 대한 정보를 가져올 수 있도록 한다.
2. Hisnet으로부터 가져온 시간표에서 추가적인 수정이 가능하도록 한다.
3. 과목마다 알람 on/off 기능을 설정할 수 있도록 한다.
4. 알람이 켜져있는 시간표는 아이콘을 통해 별다른 실행없이 확인이 가능하도록 한다.
5. 한동대학교 지도에 건물 위치 표시 및 건물 이름에 대한 정보를 제공한다.

B. 향후 과제

출석태깅 알람어플인 '잊지말자 출책!'은 정규 학기 출석 태깅을 중점으로 알람 설정 기능이 구현되어 있다. 때문에 기타 다른 편의적인 기능들에 대해서는 추후 개발이 더욱 필요하다.

예를 들면 단순히 정규 학기의 수업 태깅알람 뿐 아니라 시간표 레이아웃을 다중으로 설계하여 유저가 여러 시간표를 테마별로 손쉽게 사용할 수 있도록 함으로써 유저의 편리를 더욱 고려할 수 있다.

Ex1. 계절학기용 시간표 (여름, 겨울)

Ex2. 밥고 등 약속 전용 알람

Ex3. 단순 알람기능(세탁기 건조기 완료시간 알람 설정 기능 등)

기타 다양한 기능들을 생각해 볼 수 있다.

또한 시험 테스트를 통해 사용자 경험을 토대로 앞으로 더 편리한 UI 제공과 더 나은 디자인으로 버전 업데이트를 통해 지속적인 서비스 제공이 필요하다.

6. 참고문헌

- 안드로이드 스튜디오 스터디

<https://blog.naver.com/yoonyj1106>

<https://developer.android.com/courses/fundamentals-training/toc-v2>

강샘의 안드로이드 프로그래밍 (책 저자: 강성윤)

안드로이드 스튜디오를 활용한 실전 앱 만들기 (책 저자: 조상철)

- noti피케이션

<https://stackoverflow.com/questions/8766739/show-an-alert-dialog-in-broadcast-receiver-after-a-system-reboot>

<https://stackoverflow.com/questions/17906037/broadcastreceiver-onreceive-open-dialog>

<https://dreamaz.tistory.com/213>

<https://debugdaldal.tistory.com/124>

<https://yenos.tistory.com/entry/Alarm-Intent-Receiver%EC%97%90%EC%84%9C-Intent%EB%A5%BC-%EC%9D%B4%EC%9A%A9%ED%95%B4-%EC%95%8C%EB%9E%8C-%ED%99%94%EB%A9%B4-%EB%9D%84%EC%9A%B0%EA%B8%B0>

<https://yenos.tistory.com/m/entry/Alarm-Intent-Receiver%EC%97%90%EC%84%9C-Intent%EB%A5%BC-%EC%9D%B4%EC%9A%A9%ED%95%B4-%EC%95%8C%EB%9E%8C-%ED%99%94%EB%A9%B4-%EB%9D%84%EC%9A%B0%EA%B8%B0?category=471322>

<https://m.blog.naver.com/PostView.nhn?blogId=yjsplay2002&logNo=50107917483&proxyReferer=https%3A%2F%2Fwww.google.com%2F>

<https://kgh940525.tistory.com/entry/Android-%EB%B8%8C%EB%A1%9C%EB%93%9C%EC%BA%90%EC%8A%A4%ED%8A%B8-%EC%88%98%EC%8B%A0%EC%9E%90>

<https://developer88.tistory.com/34>

<https://ccdev.tistory.com/14>

<https://www.androidpub.com/2155459>

<https://medium.com/@nhancv/android-show-dialog-without-activity-context-94661d48400f>

<https://sh-itstory.tistory.com/63?category=789711>

<https://medium.com/@henen/%EB%B9%A0%EB%A5%B4%EA%B2%8C-%EB%B0%B0%EC%9A%B0%EB%8A%94-%EC%95%88%EB%93%9C%EB%A1%9C%EC%9D%B4%EB%93%9C-intent-3-%EC%97%91%ED%8B%B0%EB%B9%84%ED%8B%B0%EA%B0%84%EC%9D%98-%EB%8D%B0%EC%9D%B4%ED%84%B0-%EC%A0%84%EC%86%A1-log-2561fc07e809>
<https://newgenerationkorea.wordpress.com/2015/07/09/%EC%9D%B8%ED%85%90%ED%8A%B8intent%EB%9E%80-%EB%AC%B4%EC%97%87%EC%9D%B8%EA%B0%80/>
<https://gubonny.tistory.com/12>
<https://developer88.tistory.com/83>
<https://namget.tistory.com/entry/%EC%95%88%EB%93%9C%EB%A1%9C%EC%9D%B4%EB%93%9C-AlarmManager-With-Notification-Example-%EC%95%8C%EB%9E%8C%EB%A7%A4%EB%8B%88%EC%A0%80%EB%A5%BC-%EC%9D%B4%EC%9A%A9%ED%95%98%EC%97%AC-%ED%91%B8%EC%89%AC%EC%83%9D%EC%84%B1-%EC%98%88%EC%A0%9C>
<https://programmingfbf7290.tistory.com/entry/7-AlarmManager-%EC%9E%91%EC%97%85-%EC%8A%A4%EC%BC%80%EC%A4%84%EB%A7%81>
<https://kanzler.tistory.com/315>
<https://doraetul.tistory.com/73>
<https://arabiannight.tistory.com/entry/%EC%95%88%EB%93%9C%EB%A1%9C%EC%9D%B4%EB%93%9CAndroid-dd>
<https://gogorchg.tistory.com/entry/Android-AlarmManager%EB%A5%BC-%EC%9D%B4%EC%9A%A9%ED%95%9C-Schedule-%EA%B4%80%EB%A6%AC>
<http://veryeasyandroid.blogspot.com/2016/02/service.html>

- 앱 디자인 초안

<https://ovenapp.io/>

- 외부 앱 실행(연동 or 설치) 시키기

<https://rockdrumy.tistory.com/1352>

<https://developer.android.com/training/basics/intents?hl=ko>

<https://860709.tistory.com/96>

- DB 설계

<https://bitsoul.tistory.com/41>

<https://blog.naver.com/yoonyj1106>

<https://recipes4dev.tistory.com/118>

<https://kuroikuma.tistory.com/75>

- Fragment 구현

<https://blog.naver.com/yoonyj1106>

<https://recipes4dev.tistory.com/58>

<https://developer.android.com/guide/components/fragments?hl=ko>

<https://wimir-dev.tistory.com/13>

- Dialog 상자

<https://blog.naver.com/yoonyj1106>

<https://mainia.tistory.com/2027>