

2017 - 1 졸업작품 결과 보고서

송이야 학교 왔니

주제 : 비콘을 활용한 교내 위치기반 시스템

팀명	추적추적
팀원	김민주(1415191) 안진일(1416898) 정혜원(1410604)
제출날짜	2017년 10월
지도교수	이광수 (서명)

목차

1. 서론

- i. 개발 동기 및 목표
- ii. 개발 언어 및 환경

2. 본론

- i. 기능 설명 및 구조도
- ii. 프로그램 도큐먼트

3. 결론

- i. 개선사항
- ii. 기대효과 및 발전방향

4. 부록

- i. 소스코드
- ii. 참고자료

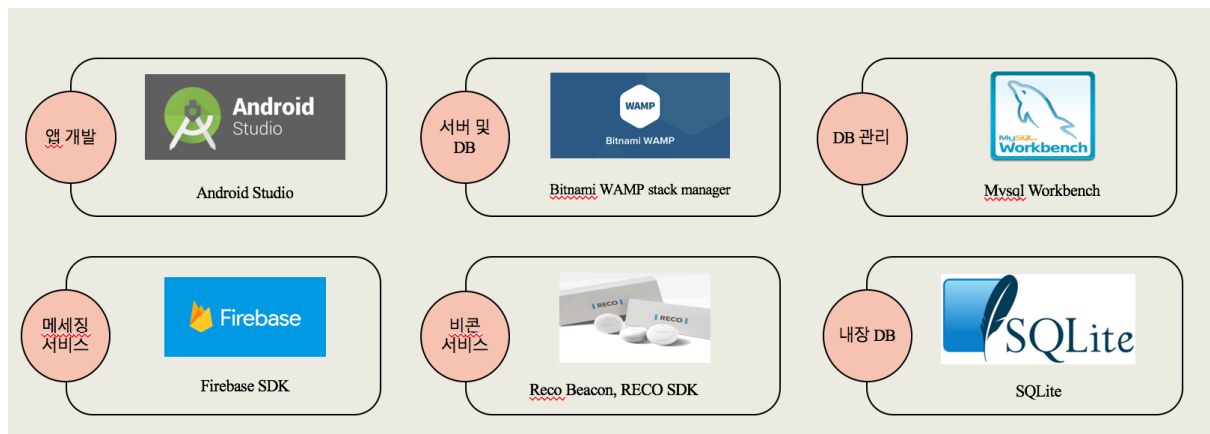
서론

i. 개발 동기 및 목표

현재 우리 학교에는 출석 체크를 위한 비콘이 강의실마다 비치되어 있는데 출석을 제외한 용도로는 쓰이는 곳이 없다. 요즘 부상하고 있는 저전력 블루투스 기능과 더불어 교내 곳곳에 위치한 비콘을 더 활용할 수 없을까 하는 의문을 갖게 되었고, 학생들의 편의를 고려한 위치 기반 시스템을 구현해보고자 이 프로젝트를 시작하게 되었다.

우리 프로젝트의 목적은 비콘과 스마트폰 간의 통신을 이용하여 위치 기반 시스템 및 교내 일정과 출석 관련 기능을 구현함으로써 사용자간의 위치 정보를 서로 공유하고 보다 편리한 교내 생활을 제공하는 것이다.

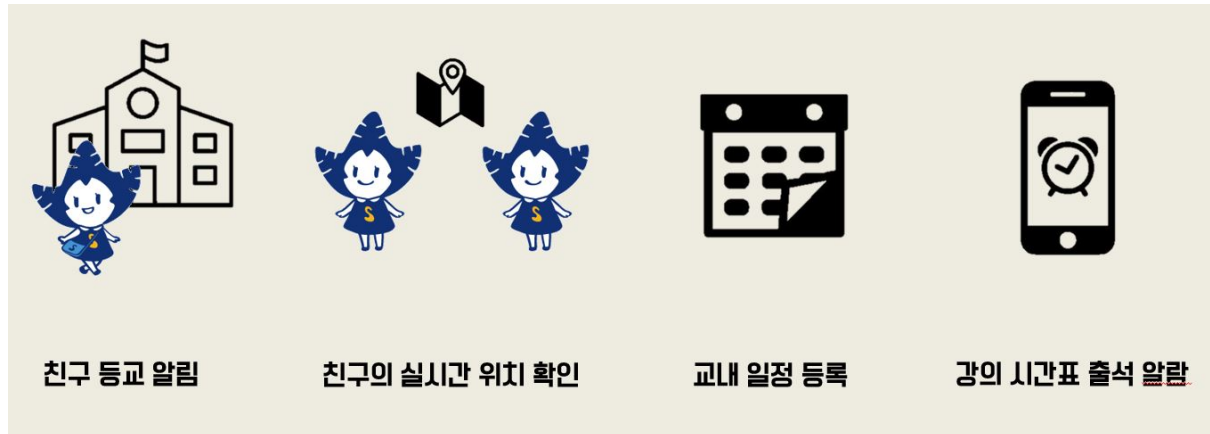
ii. 개발언어 및 환경



앱 개발은 Android Studio에서 진행했고 서버와 DB는 WAMP(Bitnami)와 SQLite를 사용하여 PHP와 Android JAVA로 구현하였으며, DB 관리를 위해서는 Mysql Workbench를 사용했다. 비콘 서비스로는 Reco사의 비콘과 RECO SDK, 푸시 메시지 전송을 위한 메세징 서비스로는 구글의 Firebase SDK를 활용하였다.

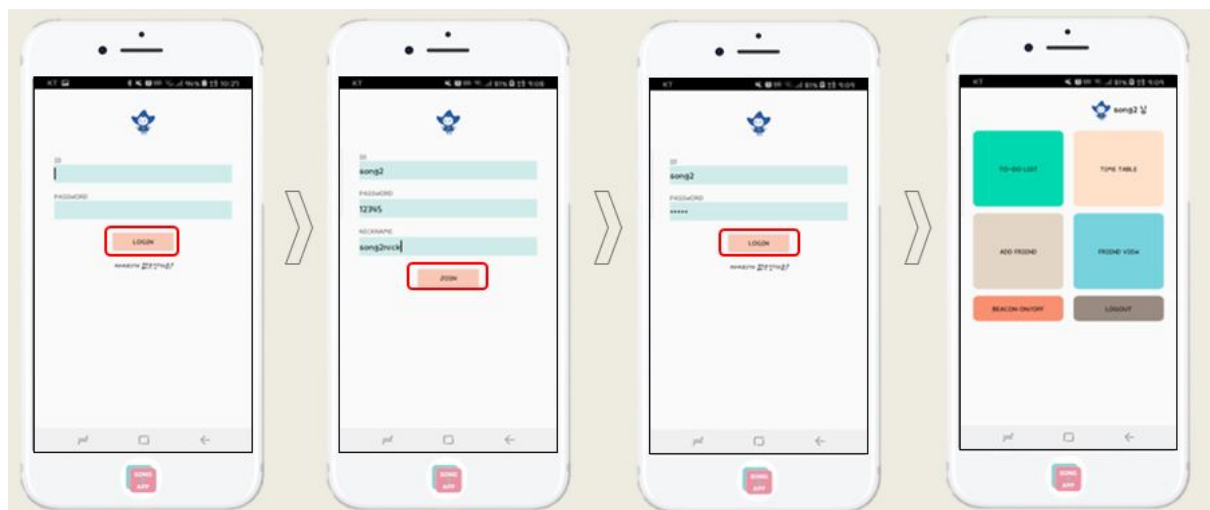
본론

i. 기능 설명 및 구조도



기능은 크게 친구 등교 알림, 친구의 실시간 위치 확인, 교내 일정 등록, 강의 시간표 출석 알람이 있다.

- 회원가입/로그인

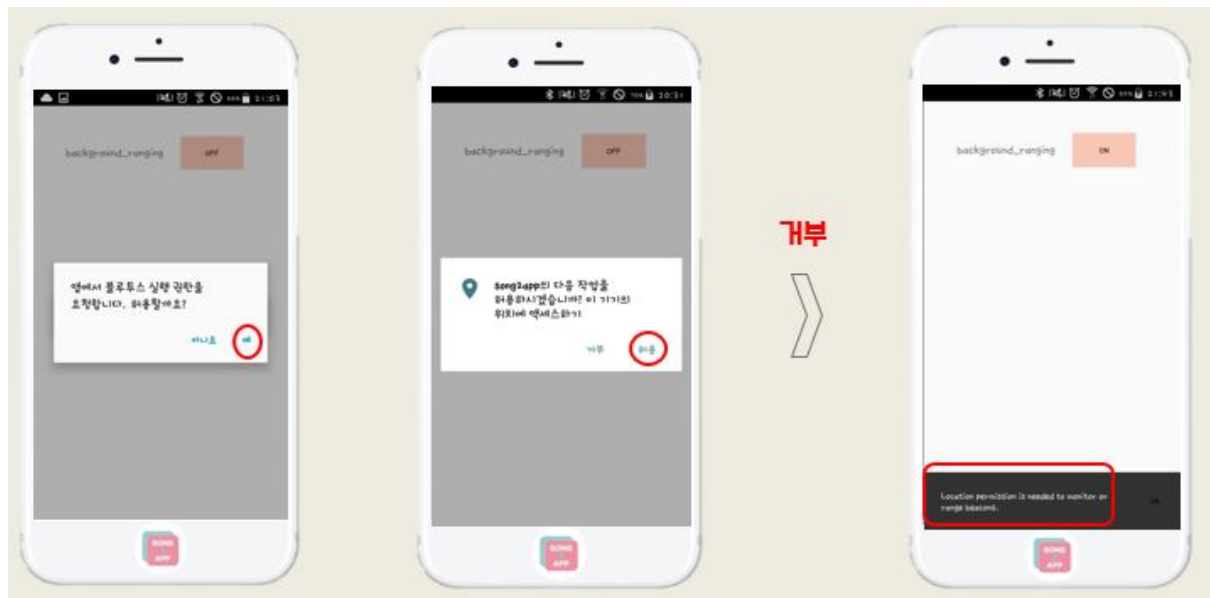


앱 최초 실행 시에 로그인 화면이 실행된다. 아이디가 없을 경우에는 회원가입을 한 후에 로그인할 수 있다.

- 자동로그인/로그아웃

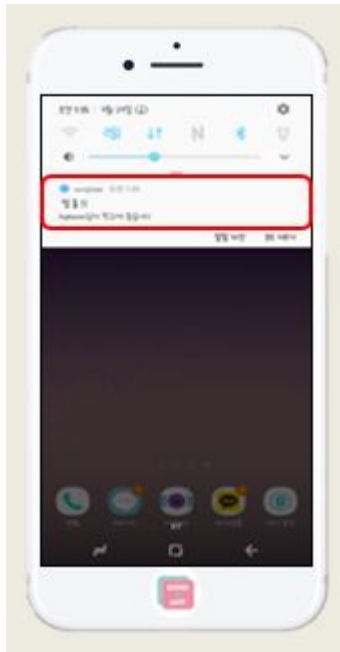
한 번 로그인을 하면 앱을 종료 시켜도 로그아웃을 해주지 않는 이상 로그인 상태가 유지된다. 로그아웃은 메인 화면의 로그아웃 버튼을 통해서 가능하다.

- 비콘 설정



메인 화면의 비콘 설정에 들어가서 Background_ranging버튼을 on으로 설정해줘야 이후의 모든 기능들에 대한 서비스를 받을 수 있다. 비콘 설정에 들어가서 앱의 블루투스 실행 권한 요청 허가과 기기의 위치 액세스에 대한 허용을 해준 후에 Background_ranging 기능을 on 해줄 수 있다.

- 친구 등교 알림



학교에 와서 교내 비콘이 기기에 인식 될 시에 친구 리스트에 등록된 친구들에게 푸시 메시지로 알림이 간다.

- 친구검색/친구추가/친구삭제/친구보기(실시간위치확인)

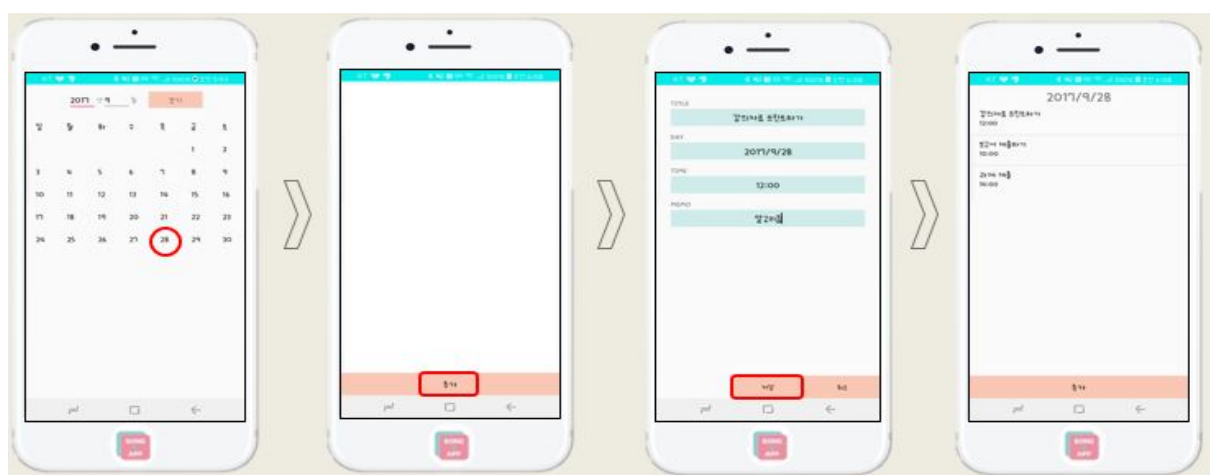


메인 화면에서 친구 추가 버튼을 통해서 친구의 아이디를 검색하여 친구 추가를 할 수 있다. 친구 추가 후에는 친구리스트에서 확인할 수 있으며 친구의 실시간 위치가 함께 표시된다.



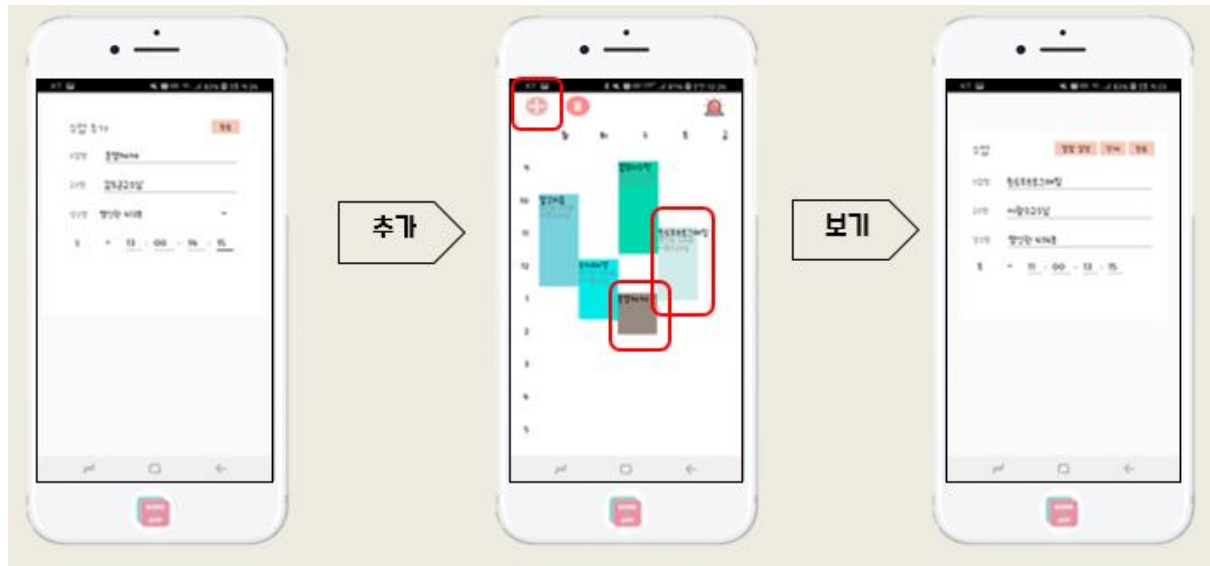
친구 리스트에서 친구 삭제를 통해 리스트에서 삭제도 가능하다.

- 교내 일정 등록 및 일정 알림



교내에서 해야 하는 일정을 등록해 놓으면 알림을 받을 수 있다. 백그라운드 기능으로 앱이 켜져 있지 않아도 교내 비콘이 인식 되면 일정 당일 1 회 일정 팝업 창이 뜨게 된다.

- 강의 시간표 출석 알림

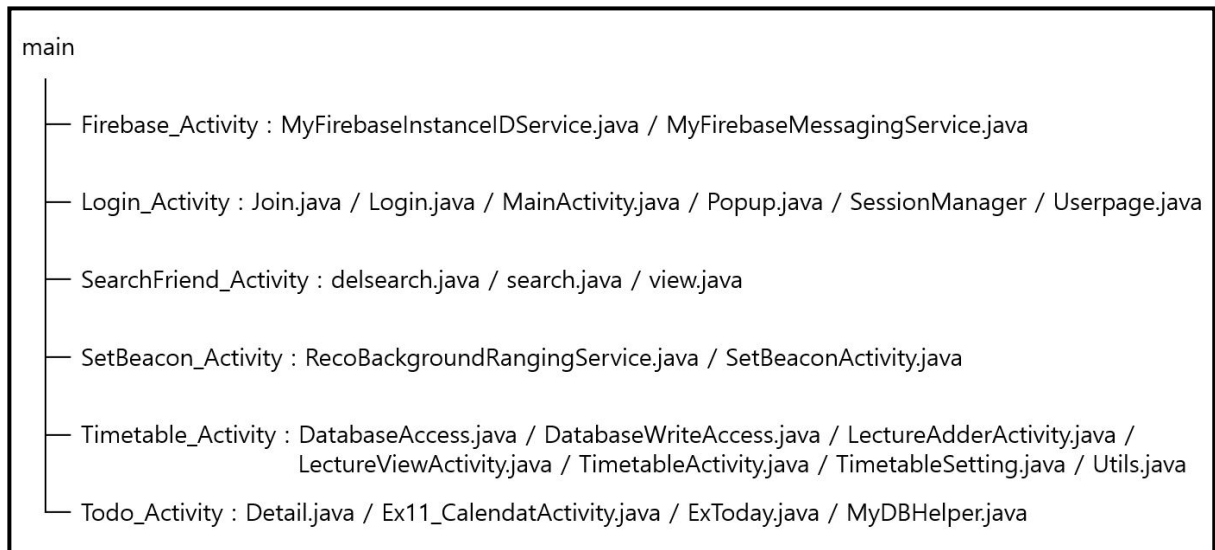


시간표 기능을 통해서 강의 명, 강의실, 교수님, 강의 시간 등을 입력하여 개인 시간표를 생성/삭제할 수 있다.



시간표 출석 알림 설정을 on 으로 설정 해놓으면 수업 시간 5 분전 해당 강의실 비콘을 인식했을 때 출석 알림을 받을 수 있다. 알림 메시지 클릭 시에 스마트 출석 앱으로 연결되며 없을 경우에는 스마트 출석 앱을 다운받을 수 있는 스토어로 연결된다.

ii. 프로그램 도큐먼트



① Firebase_Activity

ClassMyFirebaseInstanceIdService

형식 : `public class MyFirebaseInstanceIdService extends FirebaseInstanceIdService`

파일명 : `MyFirebaseInstanceIdService.java`

용도 : 구글의 Firebase Messaging 서비스를 활용하기 위해 앱의 token을 생성 하고 데이터베이스에 전송한다. 이때 생성된 토큰들을 활용하여 푸시 메시지를 전송할 수 있다.

메소드 :

public void onTokenRefresh()

토큰을 재 생성한다. Firebase의 SDK에서는 해당 앱이 설치 될 때 token이 생성 되는데 '송이야 학교 왔니'앱에서는 로그인 아이디를 단위로 token이 재생성 되게 하도록 한다. 세션아이디와 token을 매개변수로 `sendRegistrationToServer()` 함수를 호출한다.

public void sendRegistrationToServer(String token, String sessionid)

토큰과 세션아이디를 `register.php`를 통하여 데이터베이스에 전송한다.

Class MyFirebaseMessagingService

형식 : `public class MyFirebaseMessagingService`

`extends com.google.firebase.messaging.FirebaseMessagingService`

파일명 : `MyFirebaseMessagingService.java`

용도 : 전송될 푸시 메시지에 대한 설정을 해준다.

메소드 :

`private void sendNotification(String messageBody)`

푸시 메시지의 `smallIcon`, `content title`, `content text`, `sound` 등을 설정해준다.

② Login_Activity

Class Join

형식 : `public class Join extends Activity`

파일명 : `Join.java`

용도 : 자신의 회원 계정을 가지고 있지 않는 경우 회원가입 창에서 아이디, 비밀번호, 닉네임을 입력하면 회원가입이 가능하다. 입력된 값은 연동된 `join.php`에 보내져 `mysql`의 `accountinfo` 테이블에 회원 레코드로 추가된다. 회원 정보가 테이블에 존재하면 자신의 계정을 가지게 되는 것이므로 로그인을 통해 앱을 사용할 수 있게 된다. 아이디와 닉네임 비밀번호를 모두 입력해야 회원가입이 가능하고 비밀번호는 5~12자 사이로 입력해야 한다.

Class Login

형식 : `public class Login extends Activity`

파일명 : `Login.java`

용도 : `login.php`를 통해 입력한 아이디와 비밀번호가 서버와 일치하는지 확인 후 로그인 여부를 결정한다. 이 화면은 한 번 로그인하면 로그아웃하기 전까지 다시 뜨지 않는다.

메소드 :

session.createLoginSession(et.getText().toString())

로그인에 성공한 경우 계속 유지될 세션을 만들어 자동로그인을 활성화해준다.

Class MainActivity

형식 : public class MainActivity extends Activity

파일명 : MainActivity.java

용도 : 세션을 통해 로그인이 되어 있는지의 여부를 확인하고 로그인이 되어있는 경우

Userpage.class를 가져오고 되어있지 않은 경우 Login.class를 가져온다.

Class Popup

형식 : public class Popup extends Activity

파일명 : Popup.java

용도 : 로그인이 되어있는 경우 하루에 한 번 앱을 처음 켰을 때 당일에 해당하는 교내에서 해야 할 일정을 팝업 해준다.

Class SessionManager

형식 : public class SessionManager extends Activity

파일명 : SessionManager.java

용도 : 매 번 앱을 켤 때마다 로그인을 한다면 매우 불편하기 때문에 세션을 생성해줘서 로그인이 유지되는 것을 가능케 한다. 자바에서 흔히 알려진 세션매니저를 사용한다.

메소드 :

Public SessionManager(Context context)

Context에 따른 세션을 생성한다.

Public createLoginSession(String user)

생성된 세션을 user 정보를 갖는 로그인 세션으로 만든다.

Public void logoutUser()

세션을 없애고 Login.class를 불러와 로그인 화면을 띄운다.

Class Userpage

형식 : public class Userpage extends Activity

파일명 : Userpage.java

용도 : 로그인 후 화면의 버튼들을 누르면 해당하는 기능의 페이지로 넘어가게 해주는 앱의 메인 페이지다. onClick 함수를 사용하여 각각의 버튼을 활성화 시켰으며 뒤로 가기 버튼을 두 번 누를 시 앱이 종료된다.

③ SearchFriend_Activity

Class Delsearch

형식 : public class Delsearch extends Activity

파일명 : Delsearch.java

용도 : search_d.php를 통해 입력한 아이디를 유저 DB에서 검색한 후 deletefriend.php를 통해 해당 아이디를 친구 리스트에서 삭제한다.

메소드 :

private void insertToDatabase(String friend_search)

friend_search라는 아이디를 갖고 있는 유저를 DB에서 찾는다.

public void beFriend(String friend_search, String friend_logged)

친구를 생성하고 삭제하는 과정은 php 파일 내 쿼리 문 중 INSERT와 DELETE의 차이이기 때문에 함수는 Search에서 사용된 befriend 함수를 사용하되 쿼리 문을 DELETE를 사용한 deletefriend.php를 사용한다. 로그인 된 유저의 친구 리스트 DB에서 해당하는 아이디를 찾아 목록에서 제거해 친구가 아니게 한다. 이 과정은 쌍방향으로 이루어지기 때문에 친구 삭제를 당한 아이디의 친구 리스트 DB에서도 유저의 아이디가 친구 목록에서 삭제된다.

Class Search

형식 : public class Search extends Activity

파일명 : Search.java

용도 : 메인 화면인 Userpage.java에서 id가 FriendSearchBtn인 버튼을 클릭하면 들어오는 화면을 구성한다. 검색하고자 하는 친구의 아이디를 입력 칸에 입력을 하면 search.php를 통해 아이디를 유저 DB에서 검색한 후 makefriend.php를 통해 해당 아이디를 친구 리스트에서 추가한다. 위의 Delsearch.java에서 사용된 beFriend가 INSERT 쿼리 문으로 바뀌어 같은 방식으로 수행된다.

Class View

형식 : public class Userpage extends Activity

파일명 : Userpage.java

용도 : 메인 화면인 Userpage.java에서 id가 ViewFriendlist인 버튼을 클릭하면 들어오는 화면을 구성한다. viewfriend.php를 통해 유저의 친구 리스트 DB에서 친구들의 아이디와 위치 정보를 가져와 리스트 형식으로 출력해 준다. 하단의 친구 삭제 버튼을 누르면 Delsearch.java로 넘어간다.

메소드 :

getData("http://203.252.195.99/viewfriend.php?ID=" + str_loggedID)

str_loggedID라는 아이디의 유저의 친구 목록 DB를 불러온다.

④ SetBeacon_Activity

Class SetBeaconActivity

형식 : public class SetBeaconActivity extends Activity

파일명 : SetBeaconActivity.java

용도 : 비콘의 Background Ranging 기능에 대한 on/off 를 설정하고 블루투스 사용 허가 및 위치 액세스 허용 여부에 대한 메소드를 포함한다.

메소드:

private void requestLocationPermission()

사용자로부터 위치 허가 여부를 승인 받는다. 허용을 받지 못하면 beacon 기능을 사용할 수 없다.

public void onRangingToggleButtonClicked(View v)

토글 버튼 클릭 시에 Background ranging 기능이 on/off 된다.

Class RecoBackgroundRangingService

형식 : public class RecoBackgroundRangingService extends Service

implements RECORangingListener,

RECOMonitoringListener, RECOServiceConnectListener

파일명 : RecoBackgroundRangingService.java

용도 : 백그라운드에서 실행되는 파일로 비콘을 인식하여 해당 비콘 값을 받아오고 세션아이디와 함께 데이터베이스에 전송한다. 비콘 인식 시에 일정 팝업을 띄워주거나 친구들에게 등교 알림을 보낸다.

메소드:

**public void didRangeBeaconsInRegion(Collection<RECOBeacon> beacons,
RECOBeaconRegion region)**

인식되는 비콘 값들을 받아와서 가장 신호가 센 비콘 아이디를 현재 위치로 지정한다.
인식되는 비콘 값을 매개변수로 becaontoDatabase()를 호출하고, MainActivity.firstState

변수가 0 이고 인식된 비콘 아이디 값이 존재할 때 imformFriends() 함수를 호출하고 startActivity(popup)을 통해서 일정 팝업을 실행시킨다.

private void beacontoDatabase(String bid)

비콘 아이디 값을 매개변수로 받아서 updateBID.php 를 통해 데이터 베이스에 세션아이디와 함께 전송한다.

private void informFriends()

친구리스트에 등록된 친구들에게 push_notification.php를 통해서 푸시 메시지를 보낸다.

⑤ Timetable_Activity

Class DatabaseAccess

형식 :public class DatabaseAccess

파일명 :DatabaseAccess.java

용도 : android.database.sqlite.SQLiteDatabase와 android.database.sqlite.SQLiteOpenHelper를 import하여 timetable.db에 접근 가능 SELECT 쿼리 문을 사용하여 데이터베이스에 저장되어 있는 값을 읽어올 수 있다.

Class DatabaseWriteAccess

형식 :public class DatabaseWriteAccess

파일명 :DatabaseWriteAccess.java

용도 : android.database.sqlite.SQLiteDatabase, android.database.sqlite.SQLiteOpenHelper, com.readystatesoftware.sqliteasset.SQLiteAssetHelper를 import하여 timetable.db에 접근 가능하다. INSERT DELETE UPDATE 쿼리 문을 사용하여 데이터베이스에 값을 입력하거나 삭제하거나 추가 할 수 있다.

Class LectureAdderActivity

형식 :public class LectureAdderActivityextends Activity

파일명 :LectureAdderActivity.java

용도 : 강의명, 강의실, 교수님 성함, 강의 시간을 입력하여 시간표에 추가 할 수 있다. 추가된 강의는 데이터베이스에 저장된다. 추가된 강의는 TimeTableActivity.java를 통해 시간표 형식 레이아웃으로 보여지게 된다.

메소드 :

public String changePlacenameToBid(String placename)

final String lectureBid = changePlacenameToBid(lecture.place)에서 강의실 이름 lecture.place가 changePlacenameToBid(String placename)을 통해 해당 강의실 이름에 맞는 비콘 아이디 값 lectureBid으로 변경된다. 강의실과 강의실에 맞는 비콘아이디값이 데이터베이스에 저장된다.

Class LectureViewActivity

형식 :public class LectureViewActivity extends Activity

파일명 :class LectureViewActivity.java

용도 : 시간표 레이아웃에서 보여지는 강의 하나를 눌러 추가할 때 입력하였던 강의에 대한 상세 정보를 볼 수 있다. 개별 강의에 대해 알람 여부를 설정하거나 강의를 삭제할 수 있다. 알람 설정 버튼을 누르면 데이터베이스의 lecture.alarmFlag 값이 1로 변경되어 해당 강의에 대한 알람을 받을 수 있다. 삭제버튼을 누르면 데이터베이스에서 해당 강의에 관련된 레코드가 삭제된다.

Class TimeTableActivity

형식 :public class TimeTableActivity extends Activity

파일명 :TimeTableActivity.java

용도 :GridView를 사용하여 시간표 형식의 레이아웃을 보여준다. COLUMN 과 ROW를 설정하여 요일과 시간의 칸을 구분한다. LectureAdderActivity.java를 통해 추가된 강의의 이름과 교수님 성함, 강의실 이름을 색이 있는 사각형으로 간략하게 표현한다. 입력된 시간과 요일에 따라 강의실 칸의 위치를 계산하여 해당 시간과 요일에 강의를 보여줄 수 있다. 삭제 버튼은 Lecture테이블에 저장된 모든 레코드를 삭제한다. 설정 버튼을 누르면 시간표 전체 알람을 설정할 수 있는 화면으로 이동 가능하다.

Class TimetableSetting

형식 :public class TimetableSetting extends AppCompatActivity

파일명 :TimetableSetting.java

용도 : 알람이 설정된 강의에 대해 전체적으로 알람을 끄거나 켤 수 있다. 알람을 끄면 알람이 설정된 강의가 있더라도 알람이 울리지 않고 알람을 키면 알람이 설정된 강의에 대해서만 알람이 울린다. 앱을 사용하고 있지 않더라도 알람이 설정된 강의의 강의시간 5분전, 해당강의실에 있다면 상단 바에 "수업시간 5분 전 입니다. 비콘을 이용하여 출석 하세요!"Notification이 뜨게 된다.

1. if (time == classT&& min + 5 == classM&& day[alarmFlagIndex].equals(today))

데이터베이스에 저장된 강의 요일과 시간을 배열로 만들어 현재 시간과 요일을 비교한다. 현재시간에서 5분을 더한 값이 강의 시간과 같아야 한다는 조건.

2. if (classPlace[alarmFlagIndex].equals(rangingBid))

비콘 아이디 값을 주기적으로 받아오는 RecoBackgroundRangingService.minor를 rangingBid에 저장하여 현재 인식하고 있는 비콘 아이디 값과 데이터베이스에 저장된 비콘 아이디 값을 비교한다. 시간표에 저장된 강의실 비콘을 해당 시간에 인식 해야하는 조건.

Notification을 눌렀을 때 booleansookmyungsExist == getPackageList()이 true라면 otherintentgetManager().getLaunchIntentForPackage("kr.ac.sookmyung.smartcampus")에 의해 스마트 숙명 앱로 이동하게 되고 false라면 String url = "market://details?id=" + "kr.ac.sookmyung.smartcampus"에 의해 스토어의 스마트 숙명 다운로드 화면으로 이동하게 된다.

메소드 :

publicbooleangetPackageList()

PackageManager를 통해 본인의 핸드폰에 스마트 숙명 앱이 존재하는지 아닌지 판별하는 메소드. kr.ac.sookmyung.smartcampus의 패키지 명이 존재한다면 return isExist에서 true를 return하고 아니면 false를 return한다.

Class Utils

형식 :public class Utils

파일명 :Utils.java

용도 : 시간표에 활용할 요일과 시간을 계산한다.

메소드 :

public void setDay(String day)

요일을 구분한다.

public void setTime(intstartTime, intendTime)

강의 시작 숫자와 끝 숫자를 받아온다.

public void setRect(float width, float height , intpaddingLeft, intpaddingTop)

rect객체를 만든다

private static float calculateTime(intstartTime, intendTime)

받아온 숫자를 통해 시간으로 전환한다.

private static float calculateStartPosition(intstartTime)

시작 시간을 설정한다.

⑥ Todo_Activity

Class Detail

형식 : public class Detail

파일명 : Detail.java

용도 : 사용자가 입력한 정보에 따라 해당 날짜에 일정을 추가해준다. 추가하고 싶은 일정의 날짜, 제목, 시각, 내용을 입력하면 SQLite의 Today.db에 저장해준다. db.execSQL 함수에서 쿼리 문을 UPDATE, INSERT, DELETE 로 입력하면 각각 일정을 수정, 추가, 삭제가 가능하다.

Class Ex11_CalendarActivity

형식 : public class Ex11_CalendarActivity

파일명 : Ex11_CalendarActivity.java

용도 : 일정 달력의 연도, 월, 일, 요일 등 세부 디자인을 설정한다.

Class ExToday

형식 : public class ExToday

파일명 : ExToday.java

용도 : 일정 달력에서 일정을 보고 싶은 날짜를 누르면 해당 날짜에 입력된 일정들을 리스트 형식으로 보여준다. MyDBHelper를 통해 Today.db에서 해당 date의 일정 정보를 가져온다.

메소드 :

public void onItemClick(AdapterView<?> parent, View view, int position, long id)

사용자가 누른 날짜에 cursor.moveToPosition(position)을 통해 옮긴다.

protected void onActivityResult(int requestCode, int resultCode, Intent data)

위에서 옮겨 놓은 cursor에 해당하는 날짜에서 db.rawQuery("SELECT * FROM today WHERE date = '"+ today + "'", null)를 통해 일정의 모든 정보를 가져온다.

Class MyDBHelper

형식 : public class MyDBHelper

파일명 : MyDBHelper.java

용도 : 안드로이드에서 SQLite를 사용할 때 같이 쓰는 SQLiteDatabase와 SQLiteOpenHelper를 import한 DB 관련 java 파일이다.

메소드 :

public MyDBHelper(Context context, String name, CursorFactory factory,int version)

DB관리를 도와주는 Helper를 만들어 다른 java 파일에서도 이를 사용케 한다.

public void onCreate(SQLiteDatabase db)

SQLite DB를 생성한다. 일정의 정보에 맞게 db.execSQL("CREATE TABLE today(_id INTEGER PRIMARY KEY AUTOINCREMENT, " + "title TEXT, " + "date TEXT , " + "time TEXT, " + "memo TEXT);") 와 같이 설정한다.

결론

i. 개선사항

① 비콘 우선순위 여부

블루투스를 통한 무선 통신이다 보니 신호만 먼저 잡히면 가까이 있는 비콘보다 멀리 있는 비콘을 인식하는 경우가 종종 발생한다. 이에 대해 비치되어 있는 비콘마다 GPS 정보와 같은 거리 정보 등과 같은 추가적인 정보를 활용하여 우선 순위 문제를 개선해야 한다.

② 비콘 백그라운드 레인징

비콘을 통해 강의실 위치를 받아오려면 비콘 레인징 기능을 켜놓아야 하는데 아무리 블루투스가 저전력이라 해도 이럴 경우 배터리 소모가 증가하게 된다.

③ 관리자 계정의 부재

현재 구현한 앱의 경우 우리가 개발자이자 사용자이므로 보안상 문제가 없었지만, 앱이 상용화 된다면 회원가입 승인 등과 같은 절차와 계정 관련 문제들이 발생할 경우 이를 컨트롤 할 수 있는 운영자가 필요할 것이다. 따라서 앱의 관리자 계정이 있어야 된다.

④ DB의 정보 잔재

사용자가 학교를 벗어났을 때에도 그 마지막 위치 정보가 DB에 계속 남게 되어 학교를 아예 떠난 경우를 표현하지 못하는 문제점이 있다. 이에 대해선 GPS를 통해 사용자가 학교와 일정 거리 이상 떨어진 경우 학교 처리를 하는

등의 추가적인 처리가 필요하다.

ii. 기대효과 및 발전방향

현재 비콘은 나날이 기술이 발달하여 다양한 곳에 사용되고 있다. 비콘에 고주파를 함께 사용하여 사용자의 동선에 따른 쿠폰과 같은 혜택을 알림 기능을 통해 제공하기도 하고, BLE 기반의 비콘 기술이 탑재된 단말기로부터 식별코드를 수신 및 조회하여 현관문을 열지 않고도 단말기의 주인이 택배 기사나 검침원인지의 여부를 알 수도 있다.

이처럼 블루투스 등의 무선 통신의 비중이 점차 늘어나고 있는 시점에서 '송이야 학교 왔니'와 같은 위치 기반 시스템은 동아리 같은 소모임의 출석 기능이나 유치원 같은 아동의 위치가 중요한 장소,또는 사용자의 이동경로 분석과 같은 실시간 위치 정보가 중요한 곳에 비콘이 사용될 수 있다는 것을 보여준다. 소모임의 경우 이러한 시스템에 채팅 기능과 게시판 기능을 추가한다면 커뮤니티 앱으로도 활용이 가능하다.

우리 학교 앱인 '스마트 숙명'도 '송이야 학교 왔니' 시스템을 활용한다면 학생들에게 더욱 편리한 기능을 제공할 수 있다. 출석 같은 경우, 구두 출석에서 QR코드 및 NFC 출석으로 바뀌었고 앞으로는 비콘 출석만 가능하게 되는데 바쁘게 준비하다 수업 시작 시간 20분 전후에 출석하지 못하면 수업 종료 후 교수님께 말하는 수 밖에 없다. 학생이 강의실에 있으면 출석 체크를 하라는 알림 기능을 추가한다면 위와 같은 문제를 미연에 방지할 수 있다. 더불어 일정 기능을 통해 교내 에서 해야 할 일을 잊지 않고 수행할 수 있고, 채팅 기능이 추가된다면 친구들의 위치 정보와 함께 더 빠른 소통이 가능할 것이다.위에서 언급한 고주파 기술까지 더해진다면 교내 공지사항 및 이벤트를 실시간으로 학생들에게 전달할 수 있게 될 것이다.

기술의 발달로 위치 파악 및 정보 전달이 지체되지 않고 '실시간으로' 전달되는 것이 중요해진 만큼 '송이야 학교 왔니'에 사용된 비콘을 통한 위치 기반 시스템은 목적은 다

르더라도 다양한 방면에서 활용될 것이다.

부록

i. 소스코드

ii. 참고자료

① 참고문헌

-김상형.『안드로이드 프로그래밍 정복』.한빛미디어, 2011

- RECO SDK guide

② 참고자료

-<http://stackoverflow.com>

-<https://firebase.google.com/docs/guides/?hl=ko>

-<http://charactermail.tistory.com/>

-<http://202psj.tistory.com/>

-<http://manniz.tistory.com/>

