

경진대회 주제:

지역문제의 해결과 생활 편의성을 제공하는 모바일 앱

주 제	안양시 주말농장 경직자를 위한 앱 (주말Farm)
학 부	정보통신공학부 3학년
팀 명 (WebView)	20161015 이 보 미 20161023 이 지 원 20160970 고 정 연
제 출 일	2018년 5월 14일

평가	
주제 발표	
디자인 발표	
최종 발표	
합 계	

목차

1) 설계과제 제목

2) 추진 목적

3) 대상

3-1) 요즘 시대의 주말농장에 대한 관심

4) 주말 Farm 구상도

5) 화면구상도 (디자인)

6) 설계과제 목표

7) 개발 과정

7-1) 자료조사 및 자료정리

7-2) XML

- activity_main.XML
- 작물별.XML
- 기타.XML

7-3) Java

- MainActivity.JAVA
 - ① 인트로 로딩화면
 - ② 파싱 - 기상청 날씨 불러오기 (HTML파싱)
 - ③ 구글API - 구글맵 불러오기, 현재위치확인, 마커찍기
 - ④ 현재위치(GPS위도경도)와 주말농장위치 거리구하기
 - ⑤ 뒤로가기 버튼 2번 누르면 App 종료
- 작물별.JAVA (총 21개)

8) 개발 결과

9) 달성도

10) 향후계획

11) 소감

12) 참고문헌

부록) 회의록 - 33p (윤기호교수님 방문상담)

1) 설계과제 제목

“주말Farm”

- 주말에 시간을 내어 농사를 한다는 것을 가져와 ‘주말을 팔아 농사를 짓는다’라는 의미로 주말팜이라는 이름을 지었습니다.

2) 추진 목적

- 요즘 같은 분주한 도시화 시대에서 자신이 직접 농작물을 기르며 그 농작물들이 자라나는 모습들을 보면서 힐링을 느낄 수 있지만 주말농장 운영의 어려움으로 실패하는 현상도 곳곳 발생하고 있습니다.
- 주말농장을 하면서 실패하는 원인으로는 사전의 충분한 준비와 지식의 부족 등이 있습니다. 그래서 저희 조는 안양시의 주말농장을 운영하는 분들에게 도움을 드리고자 해 안양시의 주말농장을 운영하는 분들이 편리하게 사용할 수 있는 어플을 설계하고자 합니다.

3) 대상

- 주말농장을 하면서 실패하던 농장 경직자들.
- 자연으로부터 얻는 힐링을 받기위해 여가생활로 주말농장을 하시는 경직자들.
- 농사를 처음 시작하는 초보 농사 경직자들.
- 농사에 관한 과제나 숙제물을 수행할 때 정보가 필요한 모든 사람들.
- 주변에 주말농장을 지을 수 있는 공간이 필요한 안양시의 모든 사람들.

3-1) 요즘 시대의 주말농장에 대한 관심

학전정보화마을 관계자는 “주말농장은 직접 수확한 농산물을 가져가는 기회를 통해, 농사에 대한 이해와 일상속의 힐링, 어린자녀에게는 자연 학습교실의 장으로, 가족모두에게 좋은 시간을 만들 수 있을 것”이라고 말했다.

[사진 1] 주말농장 장점1

주말농장은 가족과 함께하는 농업활동 기회를 제공해 가족 간의 소통과 건강한 여가생활, 이웃 나눔 실천 등 시민의 삶의 질 향상에 기여하고 있다.

[사진 2] 주말농장 장점2

주말농장은 가장 보편적인 방식의 도시농업입니다. 최근 주말농장은 단순히 텃밭 분양에서 끝나는 것이 아니라, 도시 농부학교를 운영해 전문적인 농사 정보를 공유하고 인근 어린이집이나 유치원에 생태학습장을 제공하기도 합니다. 또한 팜 파티를 개최해 이웃끼리의 소통이 없는 도시의 삭막함을 깨뜨리고, 텃밭 중심 공동체를 형성하기도 하죠.

[사진 3] 주말농장 장점3

주말농장 텃밭은 씨앗 뿌리

기부터 수확까지 직접 기른

친환경 채소를 믿고 먹을 수

있으며 특히 아이들이 직접 체험을 통해 농업·농촌의 소중함과 생명의 신비로움을 보고 배울 수 있는 자연친화적 학습공간으로 해마다 시민들의 관심이 증가하고 있다.

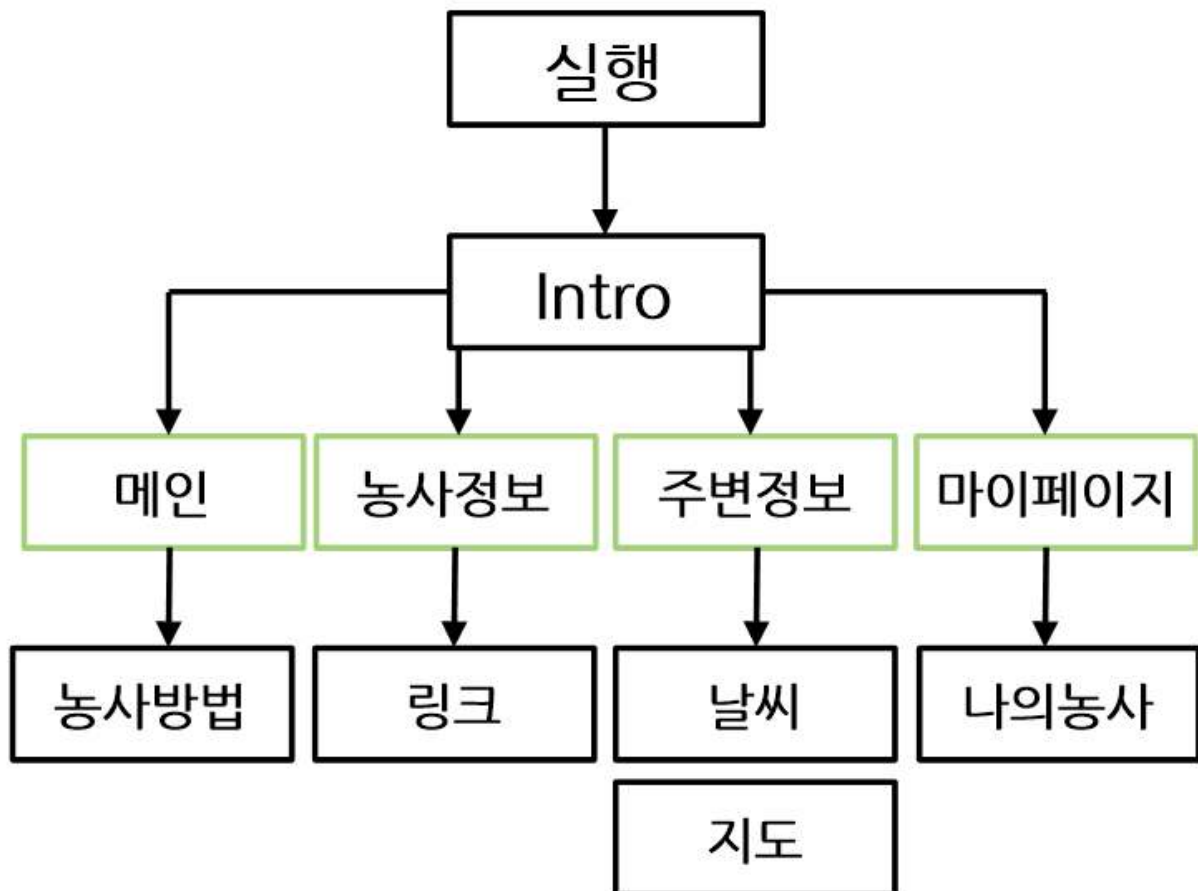


▲ 진해구 용원동 안골마을(안골동 21번지)에 마련된 '주말체험농장'을 무료로 분양한다.

[사진 4] 주말농장 장점4

- 위에 보여드리는 사진들은 주말농장의 이점에 관한 여러 기사들의 일부입니다.
- 요즘같이 도시의 숨 가쁜 생활에 지친 사람들이 주말농장에 관한 관심이 높아지고 있는 것을 볼 수 있습니다. 바쁜 일상 속에서 잠깐이라는 시간을 이용해 직접 농작물을 기르면서 자연과 소통하며 힐링을 느낄 수 있습니다. 또한 가족과 함께함으로써 가족 간의 소통을 이룰 수 있고, 자연과 소통을 하며, 직접 채소를 길러 먹으므로 바쁘고 지친 도시생활 중에서도 자연친화적인 삶을 살아갈 수 있습니다.

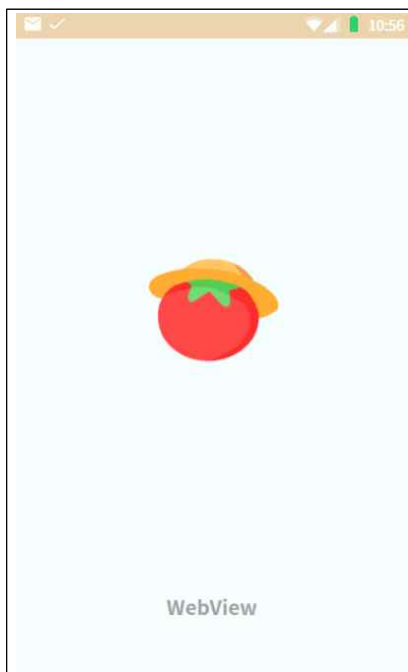
4) 주말 Farm 구상도



[사진 5] 어플 구상도

- 위 [사진 5]는 어플을 실행했을 때 메인에서 총4개의 탭의 역할을 구상하여 표현하였습니다. 기능 순서로는 아래와 같습니다.
- ① 어플을 실행합니다. 이후 약1초간 Intro화면이 보여지게 되고, 바로 메인화면을 보여줍니다.
- ② 메인화면 아래에는 4가지로 나뉜 탭이 있고 각 순서대로 메인, 농사정보, 주변정보, 마이페이지 순서입니다.
- ③ 메인Tab을 눌렀을 때 : 농사방법을 스크롤과 Horizontal스크롤로 구현하였습니다. 농작물의 구분은 뿌리, 열매, 잎줄기로 나누었으며 각 구분마다 이미지버튼을 누를 때 농작물의 정보와 농사방법을 확인 할 수 있습니다.
- ④ 농사정보Tab을 눌렀을 때 : 농작물을 키우는데 유용한 사이트를 연결해 줄 화면입니다. 모종, 비료, 퇴치제, 농사기구 총 4가지로 나누었으며 이때 해당 버튼을 누르면 이와 관련된 사이트에 링크를 걸어줍니다.
- ⑤ 주변정보Tab을 눌렀을 때 : 안양시 내에 있는 주말농장과 안양시 주변에 있는 주말농장을 확인하여 지도에 표시해 줍니다. 표시지역에는 주말농장 이름과 연락처를 제공해줍니다. 또한 GPS수신 상태일 때는 현재 나의위치와 주변 주말농장 간의 거리차를 알려줍니다. 그리고 기상청에서 제공하는 현재 날씨정보를 불러와 어플리케이션에서 한 번에 확인 할 수 있습니다.
- ⑥ 마이페이지Tab을 눌렀을 때 : 메인Tab에 있는 농사정보를 확인하다가 본인이 키우고 있거나, 키우고 싶거나, 배우고 싶거나 하는 농작물들에게 하트버튼을 누르면 마이페이지에서도 확인 할 수 있습니다. 즐겨찾기 같은 기능을 하고 있습니다.

5) 화면구상도 (디자인)



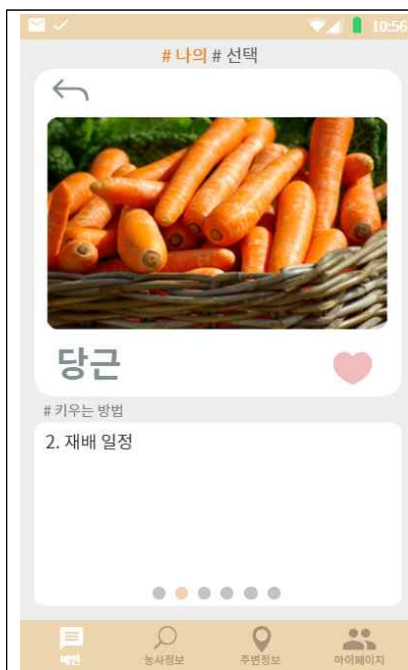
① 로딩 디자인화면



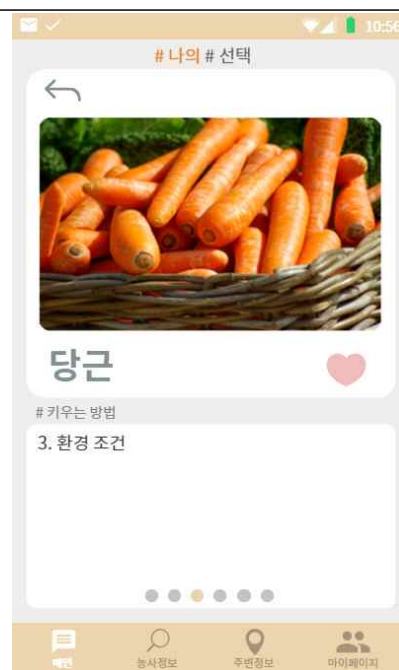
② 메인 디자인화면



②-1) 해당 농작물을 눌렀을 때



②-2) 해당 농작물을 눌렀을 때

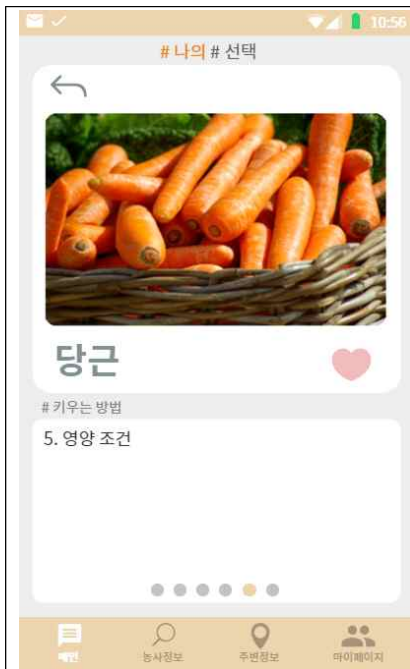


②-3) 해당 농작물을 눌렀을 때

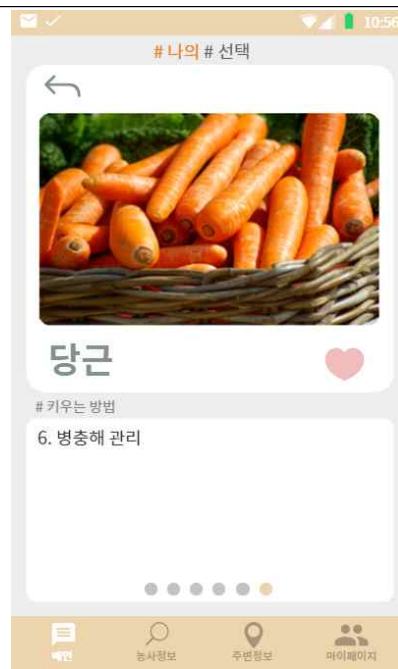


②-4) 해당 농작물을 눌렀을 때

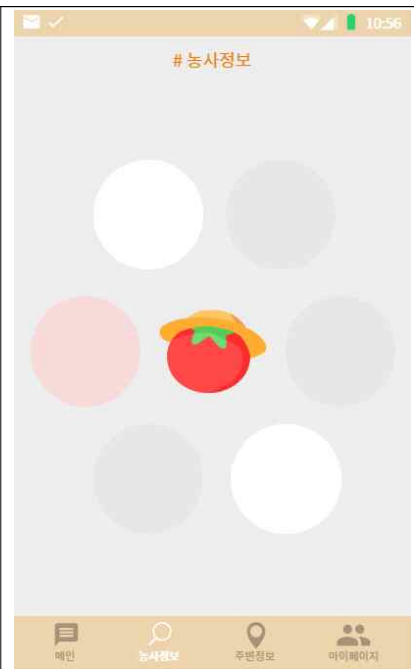
* 키우는 방법은 작물마다 기준이 다릅니다.



②-5) 해당 농작물을 눌렀을 때



②-6) 해당 농작물을 눌렀을 때

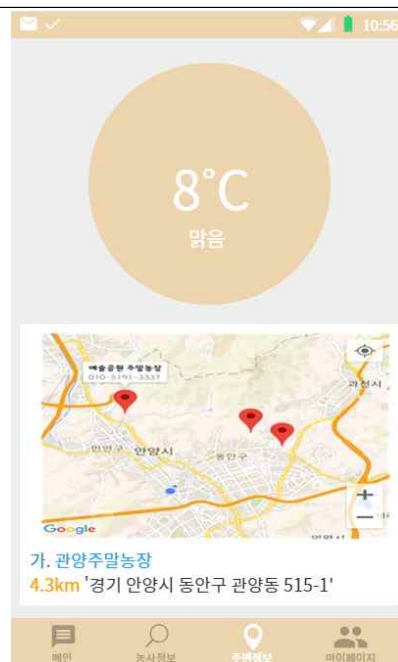


③ 농사정보 디자인화면

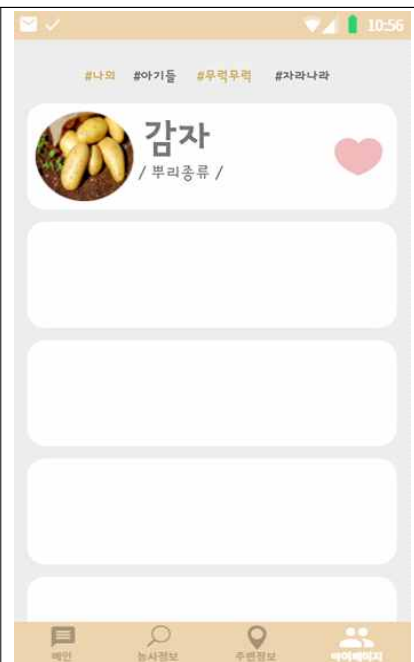
* 키우는 방법은 작물마다 기준이 다릅니다.



③-1) 해당 버튼을 눌렀을 때



④ 주변정보 디자인화면



⑤ 마이페이지 디자인화면

6) 설계과제 목표

- 목표 : 난이도 높은 기능구현보다는 디자인(GUI)과 디테일에 더 치중하여 성공적으로 어플을 완성 시킬 것입니다.
- 구현하고자 하는 기능 :
 - ▷뒤로가기 기능 (화면 넘김)
 - ▷스크롤 기능
 - ▷데이터베이스 (마이페이지)
 - ▷파싱 (날씨)
 - ▷구글 API (안양시주변 주말농장표시, 현재위치와의 거리)
 - ▷어플종료 : 뒤로가기를 2번눌러야 어플이 종료
 - ▷UI 구성 : 앞줄기, 뿌리, 열매별 테마
- 기능적 요구사항 명세서

NO.	요구사항	내용	설명	우선 순위
1	뒤로가기 기능	화면을 전환한다.	뒤로 가기 버튼을 눌렀을 때 화면이 전환되면서 보여준다.	3
2	스크롤 기능	화면을 위아래, 좌우로 움직인다.	화면을 위아래, 좌우로 움직여 원하는 내용을 찾을 수 있다.	2
3	데이터베이스	농사에 관한 정보를 저장한다.	원하는 작물을 누르고 마음에 들면 마이페이지에 따로 저장했다가 본인이 선택했던 농작물만 확인 할 수 있다.	6
4	파싱	외부에서 정보를 불러온다.	기상청에서 날씨 정보를 실시간으로 불러와 앱에 표시한다.	5
5	구글API	구글맵에 안양시 주변 주말농장을 표시하고 현재위치에서 거리를 보여준다.	안양시 주변에 있는 주말농장 위치와, 이름, 그리고 GPS상태일 때 현재위치와 주말농장사이의 거리를 보여준다.	4
6	UI 구성	앱의 전체적인 디자인을 구성한다.	로딩화면, 테마, 카테고리 등을 디자인하고 테마는 뿌리, 열매, 앞줄기로 나누어 선택할 수 있도록 한다.	1

표 1. 본 설계과제의 현실적 제한요소 항목

현실적 제한 요소들	내 용 (Content)
경제성	<ul style="list-style-type: none"> • 농사법을 알 수 있으므로 농사 실패에 대한 확률을 줄일 수 있고 경제적 손실을 막을 수 있다. • 경제적인 부담 없이 농사에 관한 어려움에 대해 정보를 얻을 수 있다.
편리성	<ul style="list-style-type: none"> • 농사법을 모를 때 지역주민에게 물어보지 않고 손쉽게 정보를 얻을 수 있다. • 휴대폰만 있으면 언제 어디서나 간편하게 정보를 얻을 수 있다. • 기억력이 좋지 않은 사람들에게도 마이페이지 서비스를 제공하여 스트레스와 고민을 덜어 줄 수 있다.
윤리성	<ul style="list-style-type: none"> • 농사를 하는데 필요한 정보들을 입력할 때 출처를 표시해 저작권 피해를 막는다. • 불법적인 부분이 없기 때문에 도덕적, 윤리적으로도 어긋나는 부분이 없다.
사회성	<ul style="list-style-type: none"> • 주말농장을 생각하거나 한 사람들에게 참고서 같은 어플이 된다. • 주말농장에 대해 어려움이 있던 사람들이 우리 어플의 정보 도움으로 인해 주말농장에 실패하는 사람들이 감소할 것으로 보인다.

표 2. 본 설계과제의 추진일정

수행 내용		일정 (2주 단위)						
		1,2	3,4	5,6	7,8	9,10	11,12	13
목표와 기준 설정	• 주제 및 목표 설정							
	• 자료조사 및 정리							
합성	• 소프트웨어 요구사항 파악							
	• 기능 블록 구성							
분석	• 기능별 구현방법 결정							
	• UI 설계							
제작	• 적용할 이론 및 기술							
	• 목표달성가능성 확인							
시험/평가	• 순서도 작성							
	• 프로그램 코딩							
결과	• UI 구현							
	• 기능 시험 및 검증							
결과	• 오류 수정							
	• 결과보고 및 시연							

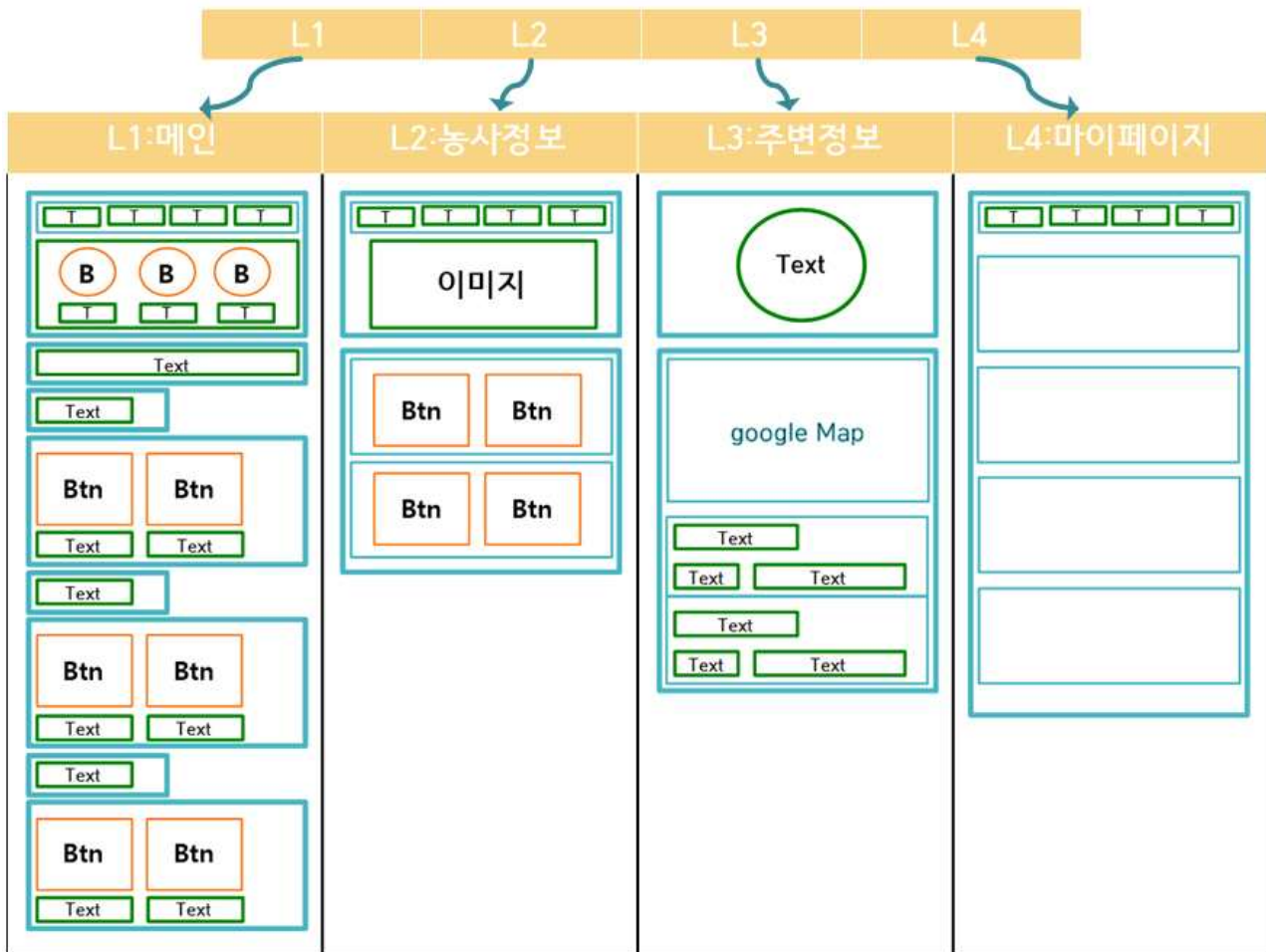
7) 개발 과정

7-1) 자료조사 및 자료정리

사진	내용
<ul style="list-style-type: none"> 작물 키우는 방법 주말팜 이미지-small 주말팜 이미지-동그라미 주말팜 이미지-원본 	<ul style="list-style-type: none"> [작물 키우는 방법] : 작물별 키우는 방법이 정리 되어있습니다. [주말팜 이미지-small] : 메인화면에서 사용 될 이미지를 필요 사이즈별로 정리 되어있습니다. [주말팜 이미지-동그라미] : 마이페이지화면에서 사용 될 이미지를 동그라미 모양으로 정리 되어있습니다. [주말팜 이미지-원본] : 버튼을 눌렀을 때 정보를 보여주는 화면에서 사용될 이미지를 원본크기로 정리 되어 있습니다. <p>※ 해당 파일을 누르면 뿌리, 열매, 잎줄기로 나누어져 있습니다. 아래에서는 뿌리부분만 작성하겠습니다.</p>
<ul style="list-style-type: none"> 감자 당근 마늘 무 양파 파 	<ul style="list-style-type: none"> [작물 키우는 방법]을 눌렀을 때 : 파일을 누르면 뿌리, 열매, 잎줄기로 나누어져 있습니다. 각 구분별 작물에 대한 키우는 방법이 정리 되어 있습니다.
 carrot  garlic  green onion  onion  potato  white radish	<ul style="list-style-type: none"> [주말팜 이미지-small]을 눌렀을 때 : 메인화면에서 사용 될 이미지사진을 필요에 맞게 작은 사이즈로 구분해서 정리해 놓았습니다. 그리고 각 사진마다 영어이름으로 저장했습니다.
 carrot  garlic  green onion  onion  potato  white radish	<ul style="list-style-type: none"> [주말팜 이미지-동그라미]을 눌렀을 때 : 마이페이지화면에서 사용 될 이미지를 동그라미 모양으로 정리 되어있습니다. 그리고 각 사진마다 영어이름으로 저장했습니다.
 carrot  garlic  green onion  onion  potato  white radish	<ul style="list-style-type: none"> [주말팜 이미지-원본]을 눌렀을 때 : 버튼을 눌렀을 때 정보를 보여주는 화면에서 사용될 이미지를 원본크기로 정리 되어 있습니다. 그리고 각 사진마다 영어이름으로 저장했습니다.

7-2) XML

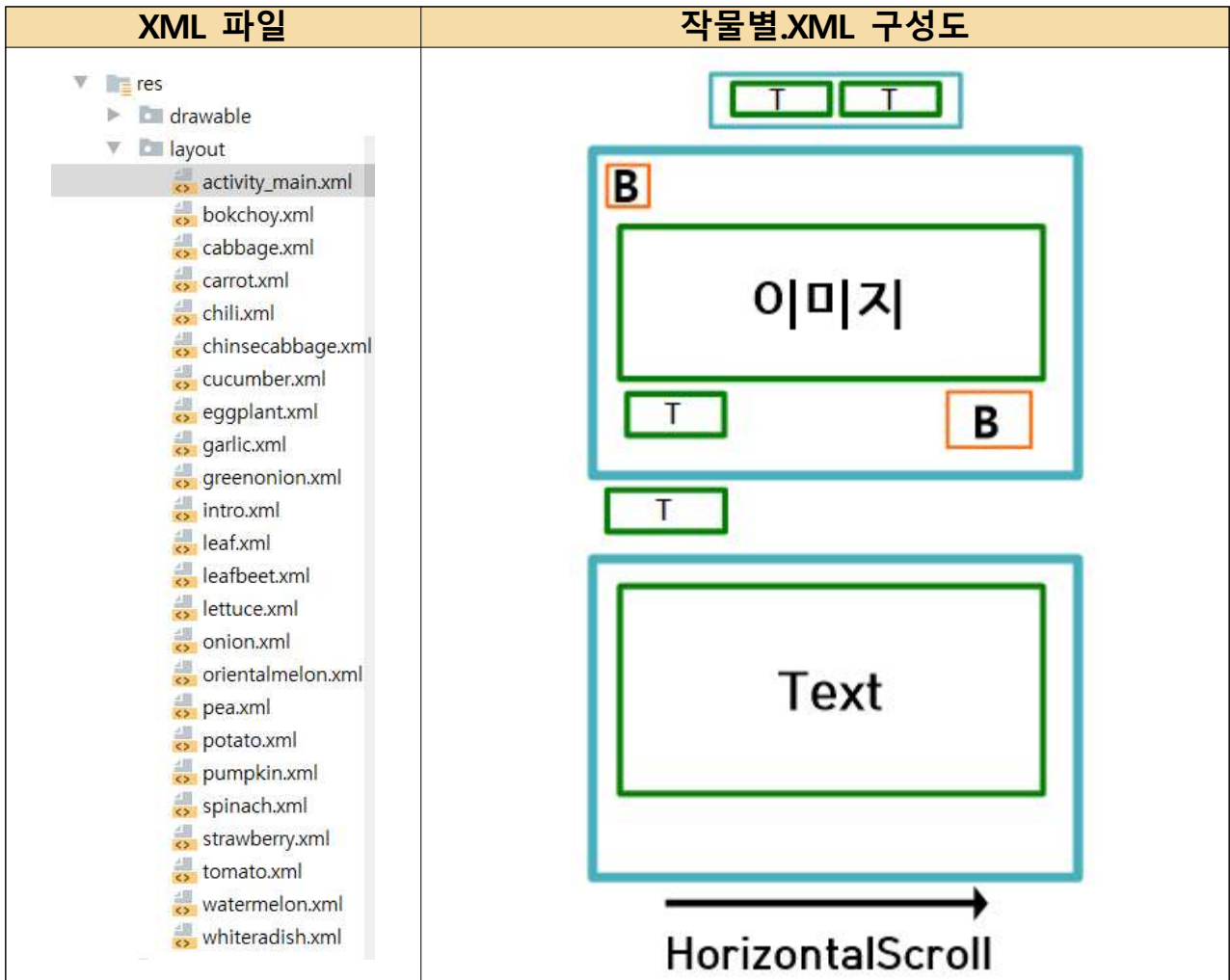
● activity_main.XML



[사진 6] activity_main.XML 구성도

- 위의 [사진 6]은 XML 구성도입니다. L1, L2, L3, L4는 TabHost를 표현하고 있습니다. 각 L1~L4의 탭에서는 순서대로 메인, 농사정보, 주변정보, 마이페이지를 나타내고 있습니다.
- L1 메인화면 : 하늘색 박스는 LinearLayout 또는 GridLayout을 나타내었습니다. 그리고 초록색 박스는 Text를 나타내었고, 주황색 박스는 이미지버튼을 나타내었습니다. 또한 이 메인화면에서는 전체적으로는 Scroll기능을 가지고 있고, 작물을 나타내는 부분에서는 HorizontalScroll 기능을 하고있습니다.
- L2 농사정보 : 마찬가지로 하늘색 박스는 LinearLayout 또는 GridLayout을 나타내었습니다. 그리고 초록색 박스는 Text를 나타내고있고, 주황색 박스는 이미지버튼을 나타내고있습니다. 이 화면에서는 스크롤기능이 없으며 LinearLayout으로 구성하였습니다.
- L3 주변정보 : 마찬가지로 하늘색 박스는 LinearLayout 또는 GridLayout을 나타내었습니다. 그리고 날씨 정보에 대한 기상청 파싱데이터가 들어올 동그라미 Text 박스와 구글맵이 들어올 공간과 이와 관련된 정보가 들어올 공간을 Text박스로 구성하였습니다.
- L4 마이페이지 : 마찬가지로 하늘색 박스는 LinearLayout 또는 GridLayout을 나타내었습니다. 이 화면에서는 메인에서 본인이 선택한 농작물을 이 페이지에서 확인 할 수 있습니다. 즐겨찾기 같은 기능을 하고 있습니다.

● 작물별.XML



[사진 7] XML파일 및 작물별.XML 구성도

- 위 사진은 작물별 XML 구성도입니다. 총 21가지의 농작물마다 xml파일을 만들어서 오른쪽 화면처럼 틀을 구성하였습니다.
- 하늘색 박스는 LinearLayout 또는 GridLayout을 나타내었습니다. 그리고 초록색 박스는 Text를 나타내었고, 주황색 박스는 이미지버튼을 나타내었습니다.
- 이때 작물을 키우는 방법에서는 HorizontalScroll을 이용하여 정보를 확인 할 수 있도록 하였습니다.
- 작물마다 키우는 방법은 다르기 때문에 각 작물에 맞는 기준으로 구현하였습니다.
- 위쪽에 있는 버튼은 뒤로가기 버튼을 나타냅니다. 이 버튼을 누르면 이전에 있던 메인화면으로 넘어가게 됩니다.
- 밑에 있는 버튼은 마이페이지에 추가 할 수 있는 버튼을 나타냅니다. 이 버튼을 누르면 해당 작물이 마이페이지에 있는 것을 확인 할 수 있습니다.

● 기타.XML

① color.xml에 있는 색상을 변경하여 툴바,상태바,탭버튼색상 변경

```
<color name="colorPrimary">#FF7DC8 </color> // 스크롤색 밑에 푸~옥 당겨주는거
<color name="colorPrimaryDark">#FFD2D2</color> // 상태바 색상
<color name="colorAccent">#FF7DB4 </color> // 툴바버튼색
```

② res-values-strings.xml 여기서 어플아이콘 이름 변경

```
<resources>
<string name="app_name">주말Farm</string> // 어플이름 바꾸기
</resources>
```

③ res-mipmap-ic_launcher.png 여기서 어플아이콘이미지 변경

Manifest.xml->(android:icon=" ") 여기에 무조건 png파일로 변경

7-3) java

● MainActivity.JAVA

① 인트로 로딩화면

```
public class Intro extends Activity {
    Handler h;
    protected void onCreate(Bundle savedInstanceState){
        super.onCreate(savedInstanceState);
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        setContentView(R.layout.intro);
        h=new Handler(); // 핸들러 생성
        h.postDelayed(run,1000); //시간 지연 1초
    }
    Runnable run = new Runnable() {
        public void run() {
            Intent i = new Intent(Intro.this, MainActivity.class);
            //인트로다음으로 메인엑티비티 화면을 보여준다.
            startActivity(i); //처음시작 어플을 시작할 때 인트로화면 보여줌
            finish(); // 마침
            overridePendingTransition(android.R.anim.fade_in, android.R.anim.fade_out);
        }
    };
}
```

- xml 파일을 만들어서 intro화면을 만든 다음에 매니페스트에 Intro.java를 등록합니다. 그리고 자바코드는 위와 같습니다.
- 인트로화면은 어플을 눌렀을 때 메인화면에 들어가기전에 1초간 보여지고 메인화면으로 넘어가게 됩니다. 여기서 중요한 것은 Intent()함수입니다.

② 파싱 - 기상청 날씨 불러오기 (HTML파싱)

```
protected void onCreate(Bundle savedInstanceState) 여기서
/* 날씨 파싱하기 위한 선언들*/
weather_tv = findViewById(R.id.weather_tv);
// 파싱해서 불러온 날씨 정보를 xml의 text에 보여주기 위해서
new WeatherAsyncTask(weather_tv).execute
("http://www.weather.go.kr/weather/forecast/timeseries.jsp", "div[class=now_weather1]");
// 본인이 필요한 정보를 파싱하기 위해서 WeatherAsyncTask라는 라이브러리에 .execute함수를 이용
해서
// .execute("http주소", 요소검사(F12)의 "클래스가 포함되어있는 파일 [필요정보의 클래스=필요정보요
소]") 순서대로 입력을 해주어 원하는 정보를 뽑아냅니다.
/*Tab3의 날씨 부분에서 현재 안양의 날씨를 기상청 HTML파싱을 이용하여
날씨를 표현하기 위한 라이브러리*/
/*파싱을 할때 Jsoup 라는 것을 이용할것 이고, 이것을 이용하기 위해서는 먼저 매니페스트에
compleie 'org.jsoup:jsoup:1.10.1' <-- 이것을 gradle:app 에 추가해준다.*//
class WeatherAsyncTask extends AsyncTask<String, Void, String> { // 파싱할때 필요한 필
수 AsyncTask라는 함수를 extetends 해준다. 이때 Asyn은 비동기식 의미한다
    TextView textView; // xml의 TextView에 보여주기 위해서 변수를 선언해준다.
    public WeatherAsyncTask(TextView textView) {
        // 파싱해온 날씨메세지를 보기위한 메소드
        this.textView = textView;
    }
protected String doInBackground(String... params) {
    String URL = params[0]; // http주소를 담을 공간 확보
    String El = params[1]; // El의 주소를 담을 공간 확보
    String result = ""; // ""안에 위를 담기 위함입니다.
    try { // 예외를 잡아내기 위해서 try catch 사용,
        org.jsoup.nodes.Document document = Jsoup.connect(URL).get();
        Elements elements = document.select(El);
        for (org.jsoup.nodes.Element element : elements) {
            result = result + element.text() + "\n";
        }
        return result;
    } catch (IOException e) {
        e.printStackTrace();
    }
    return null;
}
```

- 현재 안양의 날씨를 HTML 기상청의 정보를 Jsoup을 이용하여 파싱해 올 것입니다. 이때 먼저 jsoup 을 사용하겠다고 매니페스트에 “compleie org.jsoup.jsoup:1.10.1” 이것을 “gradle:app”에 추가해줍니다.
- 파싱할 때 필요한 AsyncTask라는 함수를 extents 해줍니다. 이때 Asyn은 비동기식을 의미합니다.
- .execute("http주소", 요소검사(F12)의 "클래스가 포함되어있는 파일 [필요정보의 클래스=필요정보요 소]") 순서대로 입력을 해주어 원하는 정보를 뽑아냅니다.

③ 구글API - 구글맵 불러오기, 현재위치확인, 마커찍기

```
protected void onCreate(Bundle savedInstanceState) {
    MapFragment mapFragment =
    (MapFragment) getFragmentManager().findFragmentById(R.id.map);
    // xml의 프래그먼트에 구글맵을 보여주기 위해서
    //mapFragment와 1:1 매칭을 시켜주기 위해서...
    map id가 있는 getFragmentManager()와 매칭을 시켜주기 위함이다.
    mapFragment.getMapAsync(this);
    // 구글맵을 불러왔을때 콜백해 주기 위함 // Async는 비동기식을 뜻함!
    // onMapReady메소드를 자동으로 호출해준다. onMapReady는 지도를 표시해주
    는 역할을 해준다.
    private final LocationListener mLocationListener = new LocationListener() {
        public void onLocationChanged(Location location) {
            //여기서 위치값이 갱신되면 이벤트가 발생한다.
            //값은 Location 형태로 리턴되며 좌표 출력 방법은 다음과 같다.
            longitude = location.getLongitude();
            //경도 : 맨위에 현재를 정해준 것을 여기서 호출해줌
            latitude = location.getLatitude();
            //위도 : 맨위에 현재를 정해준 것을 여기서 호출해줌
        }
        public void onProviderDisabled(String provider) { // 추상메소드
            // Disabled시
            Log.d("test", "onProviderDisabled, provider:" + provider);
        }
        public void onProviderEnabled(String provider) { // 추상메소드
            // Enabled시
            Log.d("test", "onProviderEnabled, provider:" + provider);
        }
        public void onStatusChanged(String provider, int status, Bundle extras) {
            // 추상메소드
            // 변경시
            Log.d("test", "onStatusChanged, provider:" + provider + ", status:" + status + "
            ,Bundle:" + extras);
        }
    };
    /*구글맵을 여기서 불러주고 이 메소드에서 필요한 좌표(위도,경도)에 마커를 찍어주고 표시해줍니
    다.*/
    public void onMapReady(GoogleMap googleMap) {
        gMap = googleMap;
        LatLng positon = new LatLng(mLatitude, mLongitude);
        // 위도 경도의 위치를 여기서 LatLng 객체를 생성해 줍니다.
        googleMap.setMapType(MAP_TYPE_NORMAL);
    }
}
```



```

// 구글맵의 첫 화면을 일반지도로 보여지게 한다.
googleMap.getUiSettings().setZoomControlsEnabled(true);
// 구글맵의 오른쪽 하단에 줌업 +,- 할 수 있도록 한다.
googleMap.moveCamera(CameraUpdateFactory.newLatLng(new LatLng(mLatitude,
mLongitude)));
// 맵이 처음 보여질때 시작좌표를 초기화 해주기 위함이다.
// 안양을 기준으로 위에서 초기값을 지정해주고 호출해주었다.
googleMap.moveCamera(CameraUpdateFactory.newLatLngZoom(positon, 12));
// 위의 시작좌표를 기준으로 크게보기위해서 줌을 지정해 주었다. 이때 줌은 숫자가 작을 수
록 크게 보인다.
// 보통의 줌은 15가 적당하다. 필요해 따라 12로 지정을 해주었다.

if (ActivityCompat.checkSelfPermission
(this, android.Manifest.permission.ACCESS_FINE_LOCATION)
== PackageManager.PERMISSION_GRANTED) {
    gMap.setMyLocationEnabled(true); // 구글맵에서 현재 위치 불러오기
} else {
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
        requestPermissions
        (new String[]{Manifest.permission.ACCESS_FINE_LOCATION},
        MY_PERMISSION_FINE_LOCATION); }
}

public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions,
int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);
    switch (requestCode) {
        case MY_PERMISSION_FINE_LOCATION:
            if (grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                if (ActivityCompat.checkSelfPermission
                (this, android.Manifest.permission.ACCESS_FINE_LOCATION)
                == PackageManager.PERMISSION_GRANTED) {
                    gMap.setMyLocationEnabled(true);
                    // 구글맵에서 현재위치 불러오기 }
            } else {
                Toast.makeText(getApplicationContext(),
                "This App Location",Toast.LENGTH_LONG).show();
                finish(); }
            break; } }

```

- mapFragment()로 구글맵을 매칭시켜주고 onMapReady는 지도를 표시해줍니다. 그리고 OnMapReadyCallback 추상메소드를 implements를 해줘야 합니다.
- 먼저 구글맵을 불러오기 전에 구글API Key번호를 얻어와야 합니다. 그리고 매니페스트에 등록해줍니다.
- 구글맵을 onMapReady메소드에 불러오고 이 메소드에서 필요한 좌표에 마커를 찍어줍니다. Marker로 마커를 찍어주고, moveCamera로 시작지점과 줌크기를 정해줍니다.
- setMyLocationEnabled(true);를 이용하여 구글맵에서 현재 위치를 불러올 수 있도록 합니다.

④ 현재위치(GPS위도경도)와 주말농장위치 거리구하기

```
void getLocation(){
    if(ActivityCompat.checkSelfPermission
        (this, Manifest.permission.ACCESS_COARSE_LOCATION)
        !=PackageManager.PERMISSION_GRANTED&&ActivityCompat.checkSelfPermission
        (this,Manifest.permission.ACCESS_COARSE_LOCATION)
        !=PackageManager.PERMISSION_GRANTED){
        ActivityCompat.requestPermissions(this, new String[]
            {Manifest.permission.ACCESS_FINE_LOCATION},REQUEST_LOCATION);
    } else{
        Location location =
locationManager.getLastKnownLocation(LocationManager.NETWORK_PROVIDER);
        if (location != null){
            double latti=location.getLatitude(); // GPS현재위치 위도
            double longi=location.getLongitude(); // GPS현재위치 경도
            double lat1 = 37.4112; // 지도시작 기준점 위도
            double lng1=126.9625; // 지도시작 기준점 경도
            double LatInRad = latti * (Math.PI/180.0);
            double LongInRad = longi * (Math.PI/180.0);
            double Lat1InRad = 37.4199 * (Math.PI/180.0), Lat2InRad = 37.4112
* (Math.PI/180.0), Lat3InRad = 37.4047 * (Math.PI/2.0);
            //예술공원,관양주말농장,포도원
            double Long1InRad = 126.9178 * (Math.PI/180.0), Long2InRad =
126.9625 * (Math.PI/180.0), Long3InRad = 126.9736
* (Math.PI/180.0); //예술공원,관양주말농장,포도원
            double Latitude1 = Lat1InRad - LatInRad, Latitude2
= Lat2InRad - LatInRad, Latitude3 = Lat3InRad - LatInRad;
            double Longitude1 = Long1InRad - LongInRad, Longitude2
= Long2InRad - LongInRad, Longitude3 = Long3InRad - LongInRad;
            double a1 =
Math.pow(Math.sin(Latitude1/2.0),2.0)+Math.cos(LatInRad)
*Math.cos(Lat1InRad)*Math.pow(Math.sin(Longitude1/2.0),2.0),
            a2 =
Math.pow(Math.sin(Latitude2/2.0),2.0)+Math.cos(LatInRad)
*Math.cos(Lat2InRad)*Math.pow(Math.sin(Longitude2/2.0),2.0),
            a3 =
Math.pow(Math.sin(Latitude3/2.0),2.0)+Math.cos(LatInRad)
*Math.cos(Lat3InRad)*Math.pow(Math.sin(Longitude3/2.0),2.0);
            double c1 = 2.0*Math.atan2(Math.sqrt(a1),Math.sqrt(1.0-a1)),
            c2 = 2.0*Math.atan2(Math.sqrt(a2),Math.sqrt(1.0-a2)),
            c3 = 2.0*Math.atan2(Math.sqrt(a3),Math.sqrt(1.0-a3));
```

```

double KEarthRadiusKms = 6376.5;
String Distance1 = String.format("%.2f", (KEarthRadiusKms * c1)),
        Distance2 = String.format("%.2f", (KEarthRadiusKms * c2)),
        Distance3 = String.format("%.2f", (KEarthRadiusKms * c3));
Location locationA = new Location("A");
locationA.setLatitude(latti);
locationA.setLongitude(longi);
Location locationB = new Location("B");
locationB.setLatitude(lat1);
locationB.setLongitude(lng1);
((TextView)findViewById(R.id.L3tab_content1)).setText(" "+Distance1+"Km");
// 현재위치에서 예술공원주말농장 까지의 거리
((TextView)findViewById(R.id.L3tab_content2)).setText(" "+Distance2+"Km");

```

- 핸드폰 내에 장착된 GPS를 이용하여 얻는 위도와 경도값을 저장할 변수를 더블형으로 선언해줍니다.
- 현재위치위도*($\pi/180$)->LatInRad에 저장 (위도), 현재위치 경도*($\pi/180$)->LongInRad에 저장 (경도)
- 목적지의위도*($\pi/180$)->Lat1InRad에 저장 (위도), 목적지의 경도*($\pi/180$)->Long1InRad에 저장(경도)
- 목적지의 위도 - 현재위치의 위도->Latitude1에 저장
- 목적지의 경도 - 현재위치의 경도->Longitude1에 저장
- $\sin((\text{목적지}-\text{현재위치})/2)^2 + \cos(\text{현재위치} * \pi/180) * \cos(\text{목적지위치} * \pi/180) * \sin((\text{목적지경도}-\text{현재위치 경도})/2)^2$ -> y/x의 아크탄젠트를 라디안 단위로 계산하고 반환, 반환값은 직각 삼각형의 대각에 마주보는 각을 나타냅니다.
- 구한값을 Km단위로 변환해 주기 위해서 선언 : Kms = 6376.5;
- 구한값을 Km단위로 변환하고 Km는 형식 문자열로 변환합니다.
- 현재위치에서 목적지까지의 거리를 화면에 출력해줍니다.

⑤ 뒤로가기 버튼 2번 누르면 App 종료

```
BackPressedHandler backpress;
// 뒤로가기 버튼을 눌렀을때 토스트메세지를 보여주기 위한 라이브러리 선언
/* 핸드폰의 뒤로가기 버튼을 눌렀을때 기능을 수행하기 위해 onBackPressed() 메소드를 오버라이딩 */
public void onBackPressed(){
    backpress.onBackPressed();    }
/*버튼을 한 번 누르면 토스트 메시지가 뜨고 두번 눌러야 꺼지는 기능을
BackPressedHandler라는 라이브러리를 사용해서 구현*/
class BackPressCloseHandler {
    private long backKeyPressedTime = 0; // 뒤로가기 버튼을 눌렀을때의 시작 시
                                       간을 0으로 초기화
    private Toast toast; // 토스트 메시지를 화면에 보여주기 위해서
    private Activity activity;
    BackPressCloseHandler(Activity context) { // 위에 선언된 라이브러리 사용
        this.activity = context;
    }
    void onBackPressed() {
        // 뒤로가기 버튼을 누른 시작시간부터 2초 넘어서 누르면
        // "뒤로가기 버튼을 한번더 누르시면 앱이 종료됩니다"를 출력한다
        if (System.currentTimeMillis() > backKeyPressedTime + 2000) {
            // 만약 2초넘어서 누르면
            //System.currentTimeMillis() : 핸드폰의 뒤로가기 버튼을 눌렀을 때
            // backKeyPressedTime : 누른순간 시작 시작
            backKeyPressedTime = System.currentTimeMillis();
            // 뒤로버튼 누른값은 처음 누른값이 됩니다.
            showGuide(); // 아래의 토스트 메시지를 보여줍니다
            return; }
        // 뒤로가기 버튼 2번 클릭시 종료되는 조건
        // 뒤로가기 버튼을 누른 시작시간부터 재 클릭 시간이 2초 이내 누르면 앱종료
        if (System.currentTimeMillis() <= backKeyPressedTime + 2000) {
            activity.finish(); // 앱을 종료합니다
            toast.cancel(); // 토스트 캔슬        }    }
    private void showGuide() { // 뒤로가기 버튼을 눌렀을때 아래의 토스트텍스트를 화
                               면에 보여준다.
        toast = Toast.makeText(activity, "'뒤로\''버튼을 한번 더 누르시면 종료됩니다.",
                                Toast.LENGTH_SHORT);
        // text의 토스트 메시지를 화면의 가운데 아래에 보여주도록 설정합니다.
        toast.show(); // 토스트 메시지를 화면에 보여줌    }
    void showGuide_1() { // 뒤로가기 버튼을 눌렀을때 아래의 토스트텍스트를 화면에 보여준다.
        Toast.LENGTH_SHORT); // text의 토스트 메시지를 화면의 가운데 아래에 보여주도록 설정합니다.
        toast.show(); // 토스트 메시지를 화면에 보여줌    }
```

- 뒤로가기 버튼을 눌렀을 때 토스트메세지를 보여주기 위해서 라이브러리를 선언해 줍니다.
- 핸드폰의 뒤로가기 버튼을 눌렀을 때 기능을 수행하기 위해 onBackPressed() 메소드를 오버라이딩 해 줍니다.
- 버튼을 한 번 누르면 토스트메세지가 뜨고 두 번 눌러야 꺼지는 기능을 BackPressCloseHandler라는 라이브러리를 사용해서 구현해줍니다.
- 우선 버튼을 눌렀을 때 시작시간을 0으로 초기화해줍니다. 그리고 선언된 라이브러리를 사용하여 if문을 이용하여 메시지를 보여줍니다.
- 버튼을 2초이내에 또 누르면 어플리케이션은 종료되고, 2초 넘어서 다시 누르면 시간을 초기화하고 해당 토스트메세지를 보여줍니다.

● 작물별.java (총 21개)

```
public class PotatoActivity extends Activity {
    ImageButton heartbtn;
    int i = 1;
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.potato);
        TextView textView1 = (TextView) findViewById(R.id.textv1); ~textView6;
        ImageButton btn = (ImageButton) findViewById(R.id.back);
        btn.setOnClickListener(new View.OnClickListener() {
            public void onClick(View view) {
                finish();
            }
        });
        textView1.setText("① 재배일정: 내용");~textView6.setText("⑥ 수확하기: 내용");
        heartbtn = (ImageButton) findViewById(R.id.heartbtn);
        heartbtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (i % 2 == 1) {
                    heartbtn.setImageResource(R.drawable.heart_fill);
                    i++;
                } else {
                    heartbtn.setImageResource(R.drawable.heart);
                    i--;
                }
            }
        });
    }
}
```

- 각 작물마다 키우는 방법을 작성해 줍니다. 이때 키우는 방법 테마마다 지정 id가 있습니다.
- 그 xml에 있는 아이디와 일치시켜 해당 텍스트를 보여줍니다.
- 버튼을 눌렀을 때 이벤트를 주어 뒤로가기 버튼을 누르면 intent를 이용하여 화면전환을 해줍니다. 또한 하트버튼을 누를때는 색을 변경하여 선택한 것처럼 보이는 효과를 줍니다.
- 총 21개의 작물을 위와 같은 방법으로 구현하였습니다.

8) 개발 결과

① 어플 아이콘



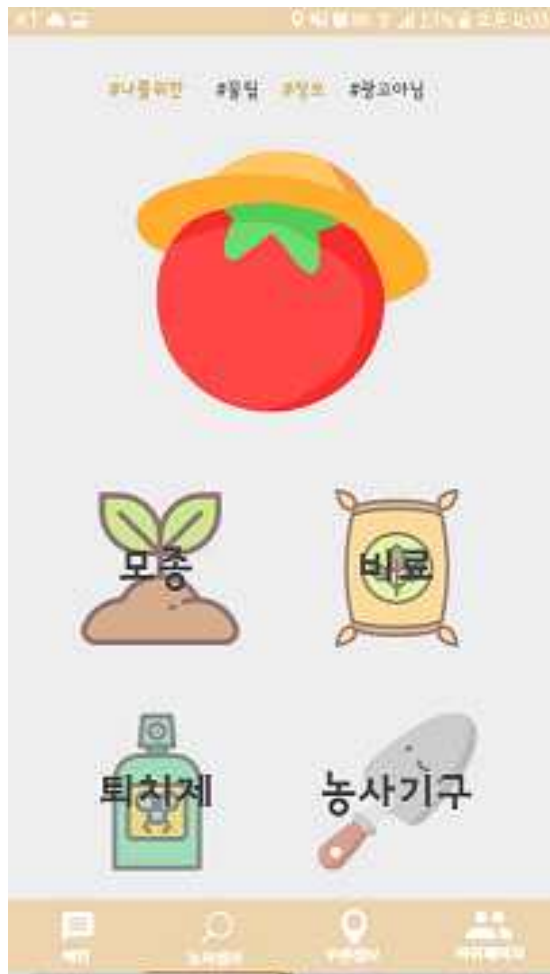
② Intro 로딩 화면



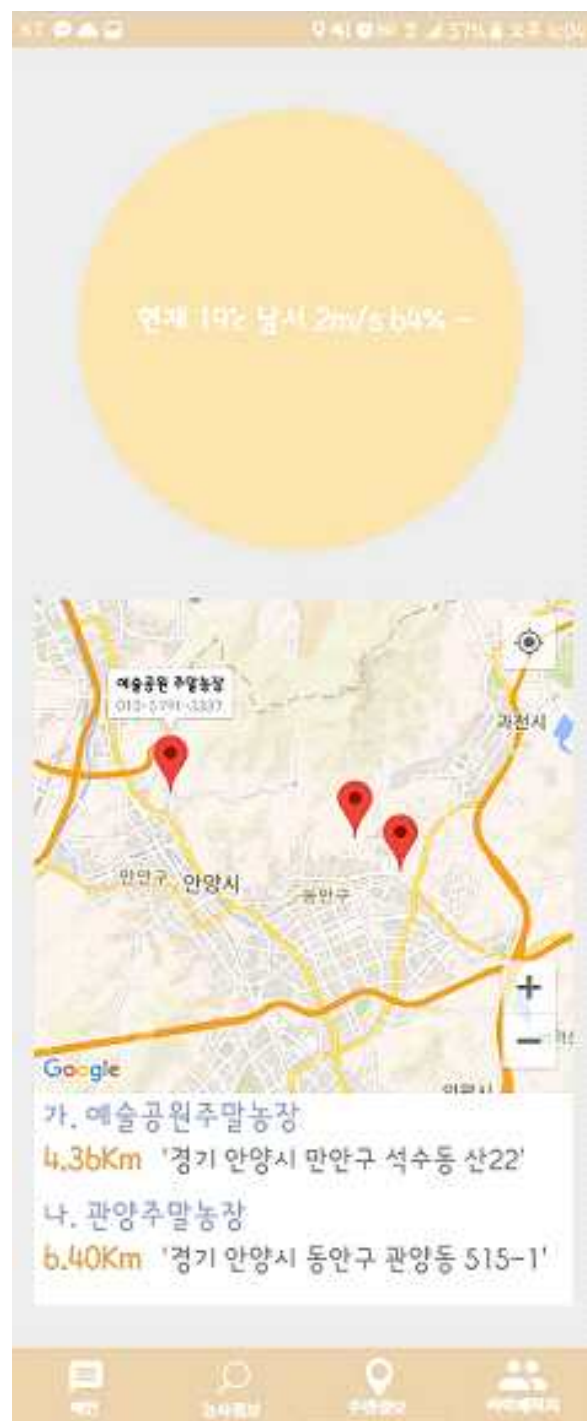
③ 메인화면



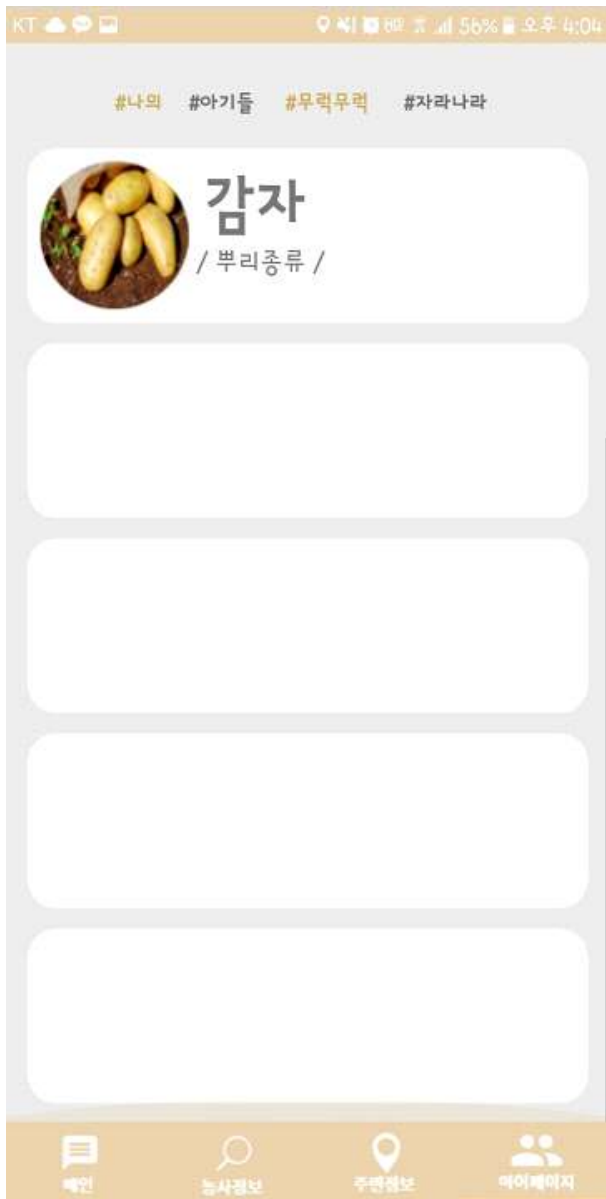
④ 농사정보 화면



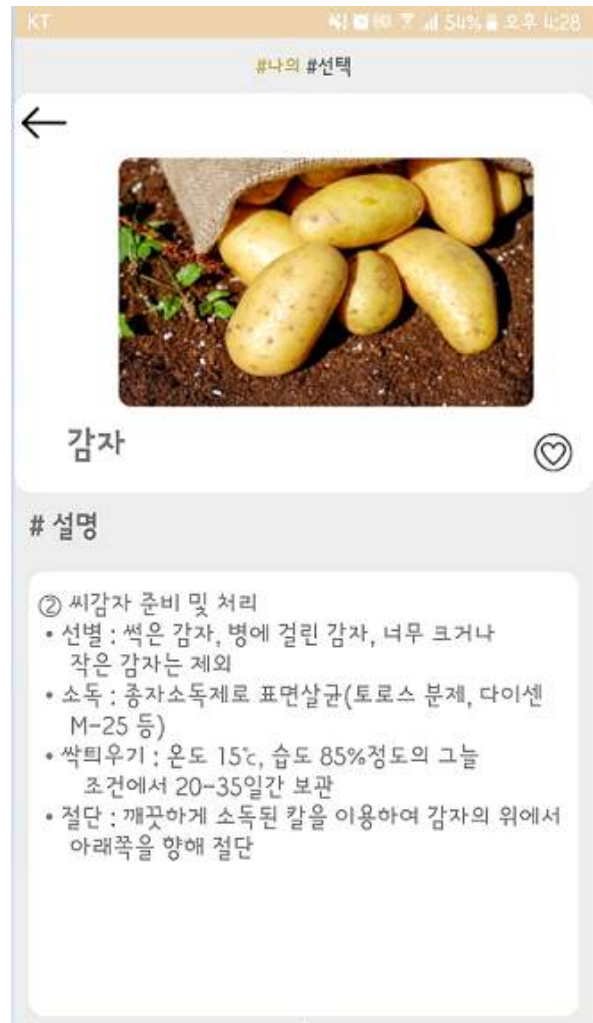
⑤ 주변정보 화면



⑥ 마이페이지 화면



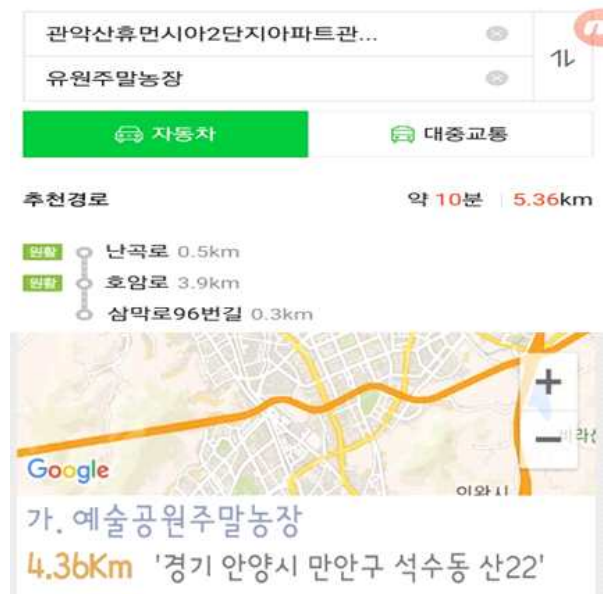
⑦ 작물을 눌렀을 때 화면



⑧ 날씨 파싱데이터 확인

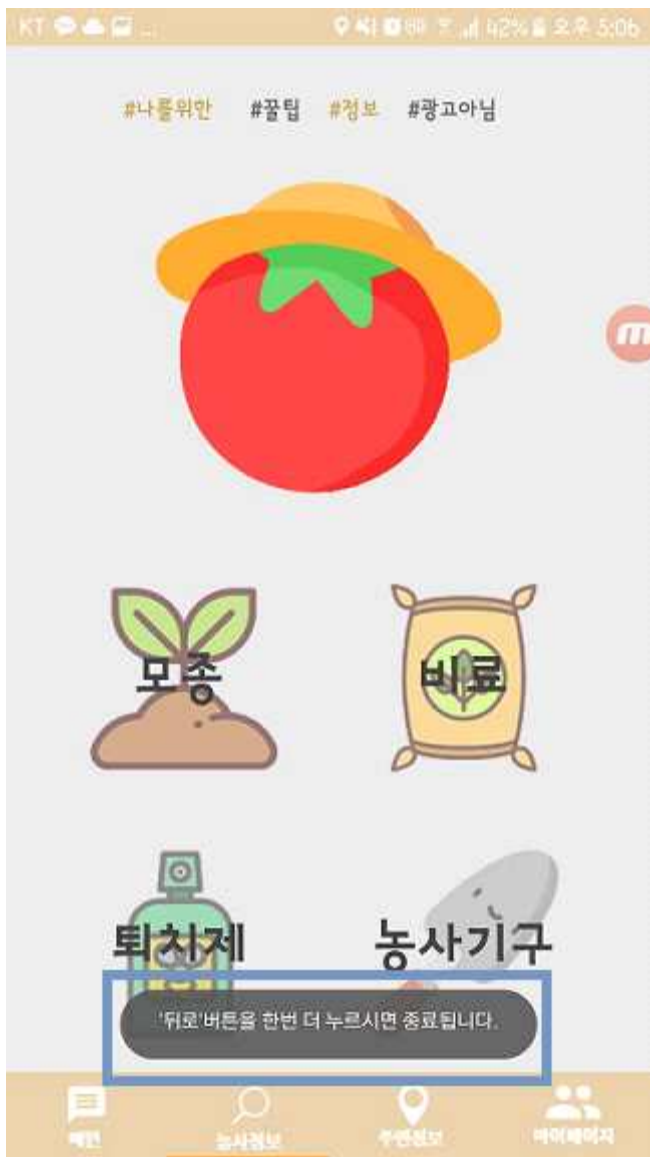


⑨ 현재위치 거리 확인



* 거리의 약간의 오차는 조금씩 있을 수 있습니다.

⑩ 뒤로가기 버튼시 토스트메세지



9) 달성도

NO.	요구사항	내용	설명	우선 순위	달성도
1	뒤로가기 기능	화면을 전환한다.	뒤로 가기 버튼을 눌렀을 때 화면이 전환되면서 보여준다.	3	100%
2	스크롤 기능	화면을 위아래, 좌우로 움직인다.	화면을 위아래, 좌우로 움직여 원하는 내용을 찾을 수 있다.	2	100%
3	데이터베이스	농사에 관한 정보를 저장한다.	원하는 작물을 누르고 마음에 들면 마이페이지에 따로 저장했다가 본인이 선택했던 농작물만 확인할 수 있다.	6	0%
4	파싱	외부에서 정보를 불러온다.	기상청에서 날씨 정보를 실시간으로 불러와 앱에 표시한다.	5	100%
5	구글API	구글맵에 안양시 주변 주말농장을 표시하고 현재위치에서 거리를 보여준다.	안양시 주변에 있는 주말농장위치와, 이름, 그리고 GPS상태일 때 현재위치와 주말농장사이의 거리를 보여준다.	4	100%
6	UI 구성	앱의 전체적인 디자인을 구성한다.	로딩화면, 테마, 카테고리 등을 디자인하고 테마는 뿌리, 열매, 잎줄기로 나누어 선택할 수 있도록 한다.	1	90%

10) 향후계획

수행 내용		일정 (2주 단위)						
		1,2	3,4	5,6	7,8	9,10	11,12	13
목표와 기준 설정	• 주제 및 목표 설정							
	• 자료조사 및 정리							
	• 소프트웨어 요구사항 파악							
합성	• 기능 블록 구성							
	• 기능별 구현방법 결정							
	• UI 설계							
분석	• 적용할 이론 및 기술							
	• 목표달성가능성 확인							
	• 순서도 작성							
제작	• 프로그램 코딩							
	• UI 구현							
시험/평가	• 기능 시험 및 검증							
	• 오류 수정							
결과	• 결과보고 및 시연							

11) 소감

이름/학번	소감
이보미/ 20161015	어플을 만들때는 기능구현보다는 디자인구현에서 더 시간이 많이 걸림을 알게되었습니다. 더 섬세하고 어떻게 편리성을 제공하는 어플을 만들지에대해서 많은 고민을 하여야했었고, 팀프로젝트에서는 팀원들이 각자 맡은 역할 수행을 잘 수행해주어서 너무 고맙습니다. 시험기간과 다른 설계와의 겹침에도 소홀하지않고 다들 열심히 해주어서 너무 고맙고 지금 까지 이렇게 진행하는 것도 모두다 팀원들 덕분입니다. 앞으로도 마지막까지 열심히 해서 다들 좋은결과 있었으면 좋겠습니다.
이지원/ 20161023	수업에서만 하는 실습으로는 부족하다 생각했는데 팀 프로젝트를 통해서 많이 배웠다. 수업 때 배운 내용을 활용 또는 심화하고 배우지 않은 것들을 안드로이드 developer 사이트를 통해 배웠으며 어플이 쉽게 만들어지지 않는다는 생각을 일깨워주는 보람찬 프로젝트였다.
고정연/ 20160970	앱을 제작하기 전에는 과연 잘할 수 있을까? 라는 생각이 들곤 했지만 팀원들과 자기가 잘할 수 있는 부분들을 각자 맡아서 맡은 부분들을 열심히 하면서 점차 완성되어가는 앱을 보니 뿌듯함을 느꼈고, 모르는 것이 생겨 물어보면 항상 친절하게 다 대답해주고 열심히 노력해준 팀원들에게도 정말로 고마운 마음이 들었습니다. 경진 대회까지의 디자인은 끝났지만 최종 앱 제작까지 같이 힘내서 했으면 좋겠고, 열심히 한만큼 좋은 앱이 나오리라는 기대를 해보게 됩니다.

12) 참고문헌

참고	제공 사이트
농사 정보	<ul style="list-style-type: none"> • 농촌진흥청 국립원예특작과학원 • "텃밭 가꾸기" • "지성아빠의 블로그" • http://www.nihhs.go.kr
이미지	<ul style="list-style-type: none"> • https://pixabay.com/ (저작권 무료)
아이콘	<ul style="list-style-type: none"> • https://www.flaticon.com/ (저작권 무료) • https://thenounproject.com/ (저작권 무료) • https://thenounproject.com/ (저작권 무료)
사진편집	<ul style="list-style-type: none"> • https://pixlr.com/editor/ • 어도비포토샵
코딩기술	<ul style="list-style-type: none"> • https://www.youtube.com/watch?v=U_dEeFGJt00 • https://www.youtube.com/watch?v=NHXa96-r8TY
책	<ul style="list-style-type: none"> • 안드로이드스튜디오를 활용한 실전 앱 만들기, 조상철 저, 영진닷컴 • 초보자를 위한 안드로이드 스튜디오, 김성재 역, 한빛미디어 • Android Studio를 활용한 안드로이드 프로그래밍, 우재남 저, 한빛아카데미

부록) 회의록

회 의 록				
회의명	모바일 프로그래밍 다섯 번째 회의			
일 시	2018.04.05 목요일		장 소	설계실
의 제	어플의 주제 수정			
토의 내용	저번 발표 후에 아이디어는 좋으나 주제에 안양 지역과의 연계성이 나타나면 좋겠다는 것과 귀농 귀촌자를 위한 어플이지만 좀 더 범위를 좁혀서 정했으면 좋겠다는 피드백을 받아 이를 바탕으로 토의를 하였습니다.			
결의 사항	안양 지역과 연계성이 있었으면 좋겠다는 점과 귀농 귀촌자들을 위한 어플이지만 범위를 조금 더 좁혔으면 좋겠다는 교수님의 피드백을 듣고 고민을 한 결과 안양 지역과 연계성이 있고 범위도 좁힌 안양 지역의 주말 농장 경작자들을 위한 어플 제작하기로 어플의 주제를 약간 수정하기로 하였습니다. 수정을 함으로써 기존의 “영농 고수”라는 어플의 이름과 주제가 맞지 않는다는 생각을 하게 되어 다음 회의까지 안양 지역의 주말 농장 경작자들을 위한 어플과 맞는 이름을 생각해 모이기로 하였습니다.			
이견 사항	어플의 변경된 주제와 알맞는 어플의 이름을 생각해 보기로 하였고, 어플을 제작할 때 필요한 정보 자료 조사를 하여 다음 회의 때 모이기로 하였습니다.			
참석 현황	대상	성명	학번	서 명(자필)
	참석자	이보미 이지원 고정연	20161015 20161023 20160970	
	불참자			
불참 사유				
작성자	고정연	작성일	2018년 04월 05일	

회 의 록				
회의명	모바일 프로그래밍 여섯 번째 회의			
일 시	2018.04.12 목요일	장 소	설계실	
의 제	어플의 이름 정하기			
토의 내용	지난 회의 때 정하지 못한 변경된 주제의 어플 이름에 대하여, 어플을 제작할 때의 필요한 정보들이 무엇 인지에 대하여, 어플의 디자인에 대하여 토의를 하였습니다.			
결의 사항	저번 회의 때에 변경된 주제에 따른 어플의 이름을 생각해 오기로 하였는데 주말 농장과도 관련이 있고 주말에 시간을 내어 농사를 한다는 것을 주말의 시간을 팔아 농사를 한다는 뜻을 가지는 주말팜(주말farm)이라는 이름으로 정하였습니다. 어플의 필요한 정보로는 기르는 방법 그 작물의 사진들을 저작권에 걸리지 않도록 조사하기로 하였고, 디자인에 대해서는 조금 더 이야기를 해보기로 하였습니다.			
이건 사항	경진대회를 위해서는 디자인이 중요하기 때문에 디자인에 대하여 더욱더 고민을 해보기로 하였습니다.			
참석 현황	대상	성명	학번	서 명(자필)
	참석자	이보미 이지원 고정연	20161015 20161023 20160970	
	불참자			
불참 사유				
작성자	고정연	작성일	2018년 04월 12일	

회 의 록				
회의명	모바일 프로그래밍 일곱 번째 회의			
일 시	2018.04.19 목요일		장 소	설계실
의 제	어플의 디자인			
토의 내용	전반적인 어플의 디자인과 어플의 내용에 따른 기능들에 맞는 디자인에 대하여 토의를 하였고, 임의로 구상을 해보았습니다.			
결의 사항	“주말팜”이라는 어플에 맞는 시그니처 마크를 보통 쉽게 텃밭 작물이나 주말 농장을 생각하면 쉽게 떠오르는 작물이 토마토라고 생각하여 토마토를 가지고 만들었습니다. 또한 어플의 카테고리는 추천 작물과 계절에 따라 분류된 작물을 확인할 수 있는 메인, 농사를 할 때 필요한 도구, 퇴치제, 비료, 씨앗/모종 등을 구매할 수 있고, 교육 사이트로 연결해주는 농사 정보, 날씨와 현재 위치와 주변에 있는 주말 농장 위치를 알려주는 주변 정보, 자신이 키우고자 하는 작물을 담은 마이페이지 등 5개의 카테고리로 나누기로 하였습니다.			
이견 사항	디자인에 대해서는 수정해야 할 부분이 계속 있을 수 있어 꾸준히 확인을 해보기로 하였고, 어플을 제작하기에 필요한 기술들을 공부하기로 하였습니다.			
참석 현황	대상	성명	학번	서 명(자필)
	참석자	이보미 이지원 고정연	20161015 20161023 20160970	
	불참자			
불참 사유				
작성자	고정연	작성일	2018년 04월 19일	

회 의 록				
회의명	모바일 프로그래밍 여덟 번째 회의			
일 시	2018.04.23 월요일	장 소	성결관 201호	
의 제	피드백에 관하여			
토의 내용	중간발표를 하면서 받은 교수님의 피드백 내용에 관련하여 제작하려고 하는 어플리케이션의 보완해야 될 부분에 대하여 토의를 하였습니다.			
결의 사항	교수님께서 저희가 농사를 잘 알지 못하니 얼마 전에 씨를 뿌리고 오신 윤기호 교수님께 주말 농장과 관련하여 여쭙보라고 하셔서 조만간 윤기호 교수님께 상담 신청을 하기로 하였고, 메인 카테고리에서 작물을 봄, 여름, 가을, 겨울 계절 별로 분류했었는데, 계절이 조금 시기가 애매해 절기로 나누면 좋을 것 같다고 피드백을 해주셨는데 절기가 24절기로 너무 많아 작물들을 분류하는 것에 대하여 조금 더 고민을 해보고 분류하기로 하였습니다. 또한 재배 방법에 대하여 더 상세하게 설명해 주어 주말 농장에 실질적인 도움을 주었으면 좋겠고 하셔서 재배 방법에 대해서는 더욱 자세한 자료 조사를 하기로 하였습니다. 또한 불필요한 기능은 빼라고 하셔서 물주기 알람기능을 빼기로 하였습니다.			
이건 사항	메인 카테고리의 작물을 무엇을 기준으로 나누면 좋을 것 인가 대하여 더욱 생각해 보도록 하였습니다.			
참석 현황	대상	성명	학번	서 명(자필)
	참석자	이보미 이지원 고정연	20161015 20161023 20160970	
	불참자			
불참 사유				
작성자	고정연	작성일	2018년 04월 23일	

회 의 록				
회의명	모바일 프로그래밍 아홉 번째 회의			
일 시	2018.04.26 목요일		장 소	설계실
의 제	작물 분류 기준			
토의 내용	어플의 메인 카테고리의 작물 분류에 대하여, 기능적인 부분에 대하여 토의를 하였습니다.			
결의 사항	저번에 메인 카테고리의 계절에 따른 작물 분류를 절기에 따라 분류하면 좋을 것 같다는 피드백을 받아 생각을 해보았지만 우리나라의 절기는 24절기로 그에 따라 분류를 하면 절기에 해당하는 작물이 균일하게 분포하지 않을 수 있고, 24 개라는 수가 너무 많아 뿌리 채소, 잎/줄기 채소, 열매 채소로 나누면 어떨까? 라는 의견이 나왔습니다. 그 의견에 따른 반응도 나쁘지 않았습니다. 또한 불필요한 기능들을 제거할 필요가 있다고 말씀하셔서 그 의견에 따라 어플을 제작하면서 조금씩 수정하기로 하였습니다.			
이건 사항	뿌리 채소, 잎/줄기 채소, 열매 채소에 따른 작물을 더욱 조사해 보리고 하였고, 어플의 기능적인 부분을 더욱더 생각해 보기로 하였습니다.			
참석 현황	대상	성명	학번	서 명(자필)
	참석자	이보미 이지원 고정연	20161015 20161023 20160970	
	불참자			
불참 사유				
작성자	고정연	작성일	2018년 04월 26일	

회 의 록				
회의명	모바일 프로그래밍 열 번째 회의			
일 시	2018.05.03 목요일		장 소	설계실
의 제	윤기호 교수님 상담			
토의 내용	중간 발표 하였을 때에 교수님께서 저희가 농사를 잘 알지 못하니 얼마 전에도 주말 농장에 씨를 뿌리고 오셨다는 윤기호 교수님께 찾아가서 주말 농장과 관련해 여러가지를 여쭙봤으면 좋겠다는 피드백을 받아서 이번에 윤기호 교수님께 찾아가서 여러가지 조언들을 구했고, 메인 카테고리의 계절 별로 나뉘던 작물들을 다시 어떤 기준으로 나눠야 할지 토의를 하였습니다..			
결의 사항	주말 농장 특성 상 매일 가서 작물을 관리하지 못하는 부분이 있어 작물들을 손이 많이 가지 않는 물을 매일 주지 않아도 쉽게 자랄 수 있는 작물들 예를 들어 말하자면 토마토, 상추, 감자 등 손이 많이 가지 않는 작물들 위주로 하면 좋을 것 같다고 말씀을 해주셨고, 농사와 관련된 "지성아빠의 블로그"라는 블로그도 한번 참고해보면 좋을 것 같다고 추천해주셨습니다. 또한 윤기호 교수님이 작물을 계절 별로 분류하면 좋을 것 같다고 제안해 주셨지만 그 부분은 수정하기로 한 부분이었기 때문에 조금 더 고민해 보기로 하였습니다.			
이건 사항	각자 맡은 역할을 열심히 수행하고, 다음 회의 때 각자 한 일들을 가져와 서로 확인하기로 하였습니다.			
참석 현황	대상	성명	학번	서 명(자필)
	참석자	이보미 이지원 고정연	20161015 20161023 20160970	
	불참자			
불참 사유				
작성자	고정연	작성일	2018년 05월 03일	

*윤기호 교수님과 상담 인증샷



회 의 록				
회의명	모바일 프로그래밍 열한 번째 회의			
일 시	2018.05.10 목요일		장 소	설계실
의 제	디자인 작업			
토의 내용	확정 짓지 못했던 메인에서의 작물 분류에 대하여 토의를 하였고, 모여서 각자 자신의 맡은 부분에 대해서 수행을 하였습니다.			
결의 사항	예전에 확정 짓지 못했던 메인에서의 작물 분류를 뿌리 채소, 잎/줄기 채소, 열매 채소 이 3가지 형태로 분류하기로 하였고, 그 외에 농사 정보, 주변 정보, 마이페이지 까지 총 4가지의 카테고리의 형태로 디자인을 하였고, 안에 기능이나 내용들은 중간 발표를 했을 때와 같습니다. 또한 모여서 디자인을 하면서 서로 계속 확인을 해주고, 일을 수행하면서 모르는 일이 있으면 서로 물어보며 알려주었습니다..			
이건 사항	자신이 맡은 부분의 일을 열심히 수행하기로 하였습니다.			
참석 현황	대상	성명	학번	서 명(자필)
	참석자	이보미 이지원 고정연	20161015 20161023 20160970	
	불참자			
불참 사유				
작성자	고정연	작성일	2018년 05월 10일	

회 의 록				
회의명	모바일 프로그래밍 열두 번째 회의			
일 시	2018.05.12 토요일	장 소	설계실	
의 제	디자인 마무리 작업			
토의 내용	경진대회에 최종 디자인을 제출하기 위해 모여서 각자 할 일을 분담하여 수행하면서 토의 하였습니다.			
결의 사항	이 전에 디자인 틀은 완성했었기 때문에 남은 일을 분담하여 메인, 농사 정보,마이페이지 등을 채워나가며 디자인 마무리 작업을 하였습니다. 또한 경진대회 최종 발표에 필요한 보고서 작성, ppt 제작, 조사한 자료 정리 등 여러가지 일을 나눠서 수행하였고, 지금까지 수행된 것들에 대하여 계속 수정하면서 검토하였습니다.			
이견 사항	자신이 맡은 부분의 일을 열심히 최선을 다하기로 하였습니다.			
참석 현황	대상	성명	학번	서 명(자필)
	참석자	이보미 이지원 고정연	20161015 20161023 20160970	
	불참자			
불참 사유				
작성자	고정연	작성일	2018년 05월 12일	