

종합설계 프로젝트 수행 보고서

| | |
|-------|--|
| 프로젝트명 | 유통기한 어플리케이션 |
| 팀번호 | S4-12 |
| 문서제목 | 수행계획서() 2차발표 중간보고서() 3차발표 중간보고서() 4차발표 중간보고서() 최종수행 보고서(O) |

2020.11.29

팀원 : 2014154049 지기용(팀장)
2016150007 김규진(팀원)
2015154053 전병관(팀원)

지도교수 : 교수 (인)
지도교수 : 교수 (인)

문서 수정 내역

| 작성일 | 대표작성자 | 버전(Revision) | 수정내용 | |
|-------------|---------|--------------|-----------|---------|
| 2020.02.22 | 전병관(팀원) | 1.0 | 수행계획서 | 최초작성 |
| 2020.04.08 | 지기용(팀장) | 2.0 | 2차 발표자료 | 설계서추가 |
| 2020.05.02 | 지기용(팀장) | 2.1 | 2차 재심발표자료 | 지적사항 추가 |
| 2020.06.27 | 김규진(팀원) | 3.0 | 3차 발표자료 | 시험결과추가 |
| 2020.07.12. | 지기용(팀장) | 4.0 | 4차 발표자료 | 시험결과 수정 |

문서 구성

| 진행단계 | 프로젝트 계획서 발표 | 중간발표1 (2월) | 중간발표2 (4월) | 학기말발표 (6월) | 최종발표 (10월) |
|------------|---|---|---|---|------------------------|
| 기본양식 | 계획서 양식 | 계획서 양식 | 계획서 양식 | 계획서 양식 | 계획서 양식 |
| 포함되는 내용 | I . 서론 (1~6) II . 본론 (1~3) 참고자료 | I . 서론 (1~6) II . 본론 (1~4) 참고자료 | I . 서론 (1~6) II . 본론 (1~5) 참고자료 | I . 서론 (1~6) II . 본론 (1~7) 참고자료 | I II III |

이 문서는 한국산업기술대학교 컴퓨터공학부의
 “종합설계” 교과목에서 프로젝트 “유통기한 어플리케이션” 을
 수행하는
 (S 4 - 12, 지기용, 김규진, 전병관)들이 작성한 것으로
 사용하기 위해서는 팀원들의 허락이 필요합니다.

목 차

I. 서론

1. 작품선정 배경 및 필요성
2. 기존 연구/기술동향 분석
3. 개발 목표
4. 팀 역할 분담
5. 개발 일정
6. 개발 환경

II. 본론

1. 개발 내용
2. 문제 및 해결방안
3. 시험시나리오
4. 상세 설계
5. Prototype 구현
6. 시험/ 테스트 결과
7. Coding & DEMO

III. 결론

1. 연구 결과
2. 작품제작 소요재료 목록

참고자료

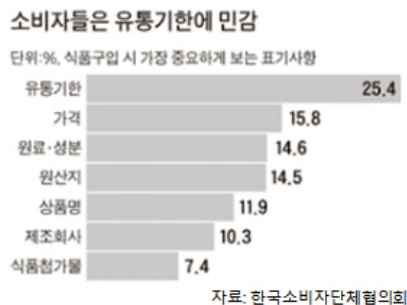
I. 서론

1. 작품선정 배경 및 필요성

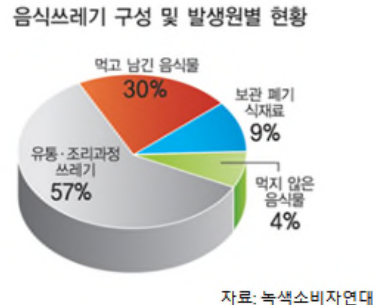
1.1 배경

한국소비자단체 협의회에서 통계를 낸 자료에 의하면 소비자들은 식품의 유통기한에 대해 제일 민감해 한다. 하지만 소비자가 구매한 식품의 유통기한을 지속적으로 인지하기란 어렵다. 녹색 소비자 연대에서 발표한 통계에는 전체 음식물 쓰레기 비율 중 유통과 조리 과정에서 발생한 쓰레기의 비율이 반 이상을 차지한다.

<표1> 소비자들은 유통기한에 민감



<표2> 음식쓰레기 구성 및 발생원별 현황



1.2 필요성

소비자가 구매한 식품들의 유통기한을 계속해서 상기시키거나 따로 기록을 해두지 않으면 관리가 어렵다. 하지만 대중에게 많이 보급된 스마트폰을 통해 식품의 유통기한 관리를 도와주는 애플리케이션이 있다면, 소비자들이 유통기한에 신경을 덜 쓰지만 안전한 식품을 섭취 가능하고 무의미하게 버려지는 음식물의 양을 줄일 수 있다.

2. 기존 연구/기술동향 분석

2.1 시중에 유통된 애플리케이션

2.1.1 유통기한 언제까지, BEEP

바코드를 스캔하여 제품의 이름을 등록해서 유통기한을 관리하는 애플리케이션이다. 단점으로는 바코드의 가독성은 떨어진다. 또한 바코드 정보는 공공데이터가 아닌 민간 기업의 정보여서 개인이 오픈 API를 받아오기 어렵다.



<그림1> 언제까지



<그림2> BEEP

2.1.2 박스히어로

사용자가 식품에 QR코드 라벨링을 추가적으로 붙여 유통기한을 관리하는 애플리케이션이

다. 단점으로는 QR코드 라벨링을 제품마다 수동으로 붙여줘야 한다. 이를 위해 유통·제조업체에게 QR코드 라벨링 협조가 있어야 하며 추가비용이 발생한다.



<그림3> 박스히어로

2.1.3 유통기한 언제지

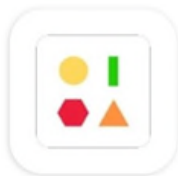
OCR기술을 이용하여 식품명을 받아들이고 레시피를 제공하는 앱이다. 단점으로는 소비자가 등록한 식품으로 레시피를 추천하지 않고 미리 저장된 레시피만 볼 수 있다.



<그림4> 유통기한 언제지

2.1.4 밥타임

사용자가 등록한 식품을 토대로 레시피를 제공한다. 단점으로는 사용자가 수동으로 식품을 등록해야 하며 유통기한이 지나도 사용자에게 알림이 오지 않는다.



<그림5> 밥타임

2.2 시중의 제품들과의 차별점

<표3>과 같이 만들고자 하는 애플리케이션은 식품을 등록할 때 사용자가 편리한 방법인 OCR이나 음성인식을 통해 등록한다. 등록된 식품의 유통기한이 임박한 경우는 사용자에게 알림을 제공한다. 또한 여유 식품을 푸드뱅크나 커뮤니티를 통하여 나눔을 할 수 있다.

<표3> 개발 중인 애플리케이션과 시중에 유통된 애플리케이션의 차별점

| | Our App | 언제까지, Beep | 박스 히어로 | 유통기한 언제지 | <u>밥타임</u> |
|-------------|---------|---------------|--------|-------------|------------|
| 알림 | ○ | ○ | X | ○ | X |
| 텍스트 외 기능 | ○ | ○ | ○ | ○ | X |
| <u>레시피</u> | ○ | X | X | △ | ○ |

3. 개발 목표

만들고자 하는 애플리케이션은 사용자가 식품명과 유통기한을 직접 입력하는 것뿐만 아니라 OCR, 음성인식 기술을 통해서도 등록할 수 있게 한다. 유통기한이 임박한 식품은 알림을 통하여 사용자에게 공지한다. 애플리케이션은 사용자가 등록한 식품을 토대로 만들 수 있는 레시피를 제공한다. 여유 식품이 있을 경우 사용자들에게 식품 나눔 커뮤니티를 제공하여 나눌 수 있게 하며 푸드뱅크의 위치 정보를 알려주어 사용자가 기증을 할 수 있게 한다.

4. 팀 역할 분담

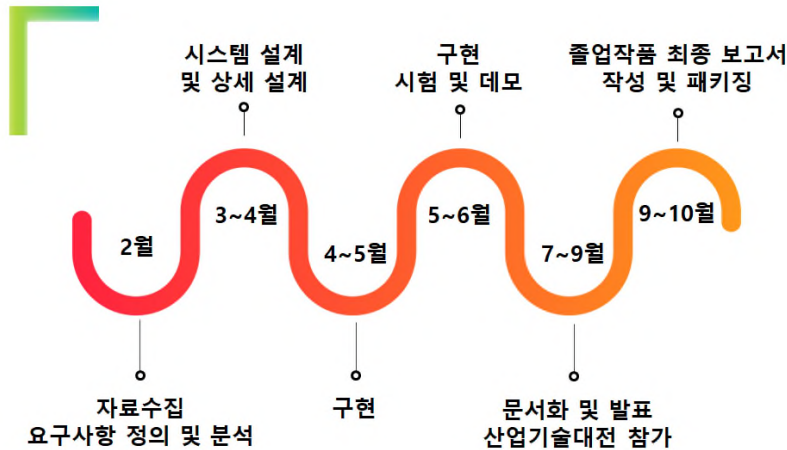
OCR, 음성인식 기술, 푸드뱅크 정보, 사용할 Open API 와 서버연동에 필요한 자료 수집은 모든 팀원이 같이 진행한다. 김규진은 OCR과 음성인식, 유통기한 정보에 관한 App설계와 DB설계를 진행하며 구현 및 연동을 진행한다. 지기용은 서버 설계 및 DB를 설계하며 서버 구현 및 애플리케이션에서 서버DB를 사용하는 로그인과 게시판 화면을 구현한다. 전병관은 푸드뱅크 설계와 서버를 테스트해보며 애플리케이션에서 동작하는지 테스트한다. 설계한 것을 바탕으로 구현 및 연동하여 App에서의 작동 및 제어테스트를 한 후, 통합테스트를 거쳐 유지보수를 끝마친다.

<표4> 역할 분담

| | 김규진 | 지기용 | 전병관 |
|------|-------------------------------------|-----------------------------|----------------------|
| 자료수집 | OCR, 음성인식 기술, 푸드뱅크, 사용할 오픈API, 서버연동 | | |
| 설 계 | Main App 설계 및 DB 설계 | Server 설계 및 DB 설계 | Sub App 설계 |
| 구 현 | Main App 구현 및 DB 연동 | Server 구현 및 App에서 연동될 화면 구현 | Server 테스트 App 동작테스트 |
| 테스트 | App에서의 작동 및 제어테스트 통합테스트 및 유지보수 | | |

5. 개발 일정

자료수집과 요구사항 정의 및 분석은 2월에 진행한다. 시스템 설계 및 상세 설계는 3월과 4월에 진행한다. 애플리케이션의 구현은 4월과 5월에 진행한다. 구현은 계속해서 진행하며 시험 및 데모를 5월과 6월에 마친다. 문서화 및 발표와 산업기술대전에 참가하는 7월에서 9월 사이에 진행한다. 졸업 작품 최종 보고서 작성과 패키징은 9월과 10월에 걸쳐 진행한다.



<그림6> 개발 일정

<표5> 개발 일정 세부사항

| 항목 | 추진 사항 | 2월 | 3월 | 4월 | 5월 | 6월 | 7월 | 8월 | 9월 | 10월 |
|-------------------|---|----|----|----|----|----|----|----|----|-----|
| 자료수집 | - 주제에 따른 사전조사, 자료수집 - 제안서작성 및 발표 | | | | | | | | | |
| 요구사항 정의 및 분석 | - 요구사항 분석 - 요구사항 정의 | | | | | | | | | |
| 시스템 설계 및 상세 설계 | - Application 화면 설계 - 서버 설계 | | | | | | | | | |
| 구현 | - Application 및 Server 코딩 - 시스템 개발 | | | | | | | | | |
| 시험 및 데모 | - Application과 시스템 통합 - 전체 시스템 테스트 | | | | | | | | | |
| 문서화 및 발표 | - 졸업작품 중간 보고서 작성 - 발표 및 산업기술 대전 참가 | | | | | | | | | |
| 최종보고서 및 패키징 | - 졸업작품 최종보고서 작성 - CD패키징(사용법, 프로그램 등) | | | | | | | | | |

6. 개발 환경

앱에 사용되는 라이브러리들 때문에 안드로이드는 최소 API 21 이상을 지원해야 하므로 Android 5.0 Lollipop으로 구현한다. 구현 시에는 Galaxy S6로 테스트한다. 애플리케이션 개발에 사용할 언어는 자바이며 'Android Studio 3.5'를 사용한다. 서버는 닷홈 사이트에서 무료 호스팅을 받아 사용한다. 서버에서는 PHP 언어를 사용하여 DB와 연동한다. 애플리케이션의 기능인 OCR, 음성인식, 위치정보는 Kakao에서 제공하는 API를 사용한다.



<그림7> Galaxy S6 사양



<그림8> 서버 호스팅 받는 곳



<그림9> 애플리케이션 개발할 환경



<그림10> 사용할 언어

II. 본론

1. 개발 내용

1.1 애플리케이션에서의 개발

애플리케이션에서는 OCR, 음성인식 활용해서 새로운 식품 정보를 입력 받는다. 애플리케이션의 로컬 DB에 식품 정보와 푸드뱅크 위치와 연락처를 저장한다. 사용자에게 유통기한 알림은 Notification Manager System을 이용하여 식품의 유통기한이 다가올 때 알림이 가 개끔 만든다. 로그인을 시도할 경우 서버에 저장된 데이터베이스의 아이디와 비밀번호가 일치하는지 확인한다. 또한 사용자에게 저장된 식품 정보를 기반으로 레시피를 추천 해준다. 여유 식품의 경우 사용자가 커뮤니티에 게시글을 작성할 수 있게 하여 나눔 커뮤니티를 형성한다.

1.2 서버에서의 개발

서버는 닷홈 사이트에서 무료 호스팅을 받는다. 관리자는 서버에 USER 테이블을 만들어 두어 애플리케이션에 회원 등록하는 유저들을 관리할 수 있다. 또한 사용자들이 작성한 게시판을 서버와 연동한다. 관리자는 myadmin 뿐만 아니라 핸드폰에서도 admin으로 접속하여 회원리스트를 보며 삭제가 가능하다.

2. 문제 및 해결 방안

2.1 OCR

OCR 기능으로는 사용자가 제공하는 사진에서 텍스트 인식이 가능하다. 사용자는 식품명과 유통기한을 얻을 수 있다. 다이얼로그로 카메라나 갤러리로 가져올 것인지 물어본다. 사용자는 누른 결과에 맞는 카메라나 갤러리 Intent로 이미지를 가져온다. Android-Image-Cropper로 사용자가 상표나 유통기한 이미지를 Crop한다. Kakao vision의 형식에 맞게 이미지 scale과 파일크기를 조정한다. retrofit으로 Kakao vision에 이미지를 전송하고 결과를 확인한다. 사용되는 API는 Kakao vision, ArthurHub/Android-Image-Cropper가 있다.

2.2 음성인식 기능

Kakao newtone sdk API를 사용한다. 이를 통해 사용자가 제공하는 음성을 인식하여 텍스트로 변환해준다. 사용자는 식품명과 유통기한을 얻을 수 있다. 음성인식에서 알고리즘으로는 다음의 과정을 걸친다. SERVICE_TYPE_DICTATION용 STT로 설정한다. 유저는 사전 설정을 거친다. sdk에서 제공하는 SpeechRecognizeListener로 구현한다. 인식 성공 시에 비슷한 후보 단어를 최대 5개까지 보여준다.

2.3 푸드뱅크 위치 정보

Kakao 지도 API를 사용해서 사용자에게 주변 푸드뱅크를 지도에 보여준다. 하루 300,000건이 무료이다. REST API가 제공되고 라이선스는 이용허락범위 제한이 없다. API를 사용할 때 다음의 과정을 걸친다. 첫째, Mapview 객체를 생성하여 Activity의 content-view에 삽입하여 지도화면으로 구현한다. 둘째, Mapview 클래스가 제공하는 메소드를 이용하여 지도의 중심점과 확대, 축소 레벨을 변경한다. 마지막으로 Mapview.PollItemEventListener Interface를 구현하여 지도에 마커를 생성한다.

2.4 레시피

저장되어있는 완성요리를 사용자에게 추천한다.. 만약 사용자가 거부한 요리의 경우는 제외하여 추천을 해준다. 사용자가 원하는 요리를 선택할 경우 해당 요리를 조리하는데 필요한 재료들과 레시피를 보여준다.

2.5 알람

유통기한이 임박한 식품이 있으면 알려준다. 또한 사용자가 식품을 등록 시에 자동으로 알람을 등록한 후 식품이 사용되면 알람을 취소한다. 알고리즘으로는 다음과 같다. 안드로이드의 AlarmManager로 알람을 설정한다. 알람은 BroadcastReceiver로 받아서 안드로이드 Notification에 등록한다. 안드로이드 리부트 시에는 AlarmManager에 설정된 알람이 지워진다. 알람을 다시 설정할 때는 ACTION_BOOT_COMPLETE 브로드 캐스트를 받는다. API는 public void setAlarm(int id)로 알람을 설정하고 public void deleteAlarm(int id)로 알람을 지운다.

2.6 로컬DB

로컬DB에서 다루는 정보는 식품 즐겨찾기, 유통기한 정보, 푸드뱅크 정보, 사용자가 거부한 레시피, 레시피 기본정보, 레시피 재료정보, 레시피 과정정보가 있다. 사용되는 API로는 다음과 같다. 안드로이드에서 제공하는 Room Library로 구현한다. Room에서 제안한 Entity, Dao, Repository로 구현한다. Room에서 Entity는 SQLite에서 테이블로 생성해준다. Dao로 실행할 SQLite문을 정의한다. Repository로 다른 모듈에서 호출할 API를 선언한다.

2.7 서버 DB

서버 DB의 기능으로는 애플리케이션에 회원가입한 사용자들의 리스트를 볼 수 있다. 그리고 사용자가 올린 글을 저장하며 다른 사용자들이 게시판에 올린 글 또한 볼 수 있다. 관리자의 경우 사용자들을 차단하거나 글을 임의로 삭제가 가능하다.

2.8 DataSync

DataSync 기능으로는 로컬 DB와 서버 DB를 연동한다. 그리고 푸드뱅크의 최신 정보가 있으면 가져온다. Sync Adapter로 구현 시 Content Provider와 유저 인증 시스템을 연동한다. 또한 로컬 DB 사용하는 Content Provider로 생성하며 유저인증을 연동한다.

2.9 보기 화면

앱 Fragment의 보기화면은 다음의 기능을 가지고 있다. 입력된 식품들을 보여주고 네 개의 탭에서 유통기한이 지난 상품들을 보여준다. 식품은 냉장식품, 냉동식품, 상온식품으로 분류하여 보여주며 식품명의 이름순과 유통기한 순으로 정렬이 가능하다. 이것은 ViewPager2와 탭 레이아웃으로 구성한다. 앱은 사용자에게 RecyclerView를 활용해서 보여주며 검색하면 검색결과만 보여준다.

2.10 즐겨찾기

사용자가 즐겨찾기한 식품의 목록을 보여주며 새로운 즐겨찾기를 추가 가능하다. 새로운 식품을 추가할 때는 OCR, STT, 수동 입력방식을 통하여 추가할 수 있다. 사용자가 OCR과 STT 버튼을 클릭할 경우에는 startActivityForResult로 해당 액티비티가 시작된다. 결과를 확인 다이얼로그로 보여주며 장바구니에 추가된다. 수동 입력이나 즐겨찾기 추가 시에는 입력 다이얼로그를 보여주고 장바구니에 추가된다.

2.11 장바구니

사용자가 입력한 유통기한 정보를 가지고 있다. 장바구니의 아이템 클래스는 따로 없으며 식품 클래스에서 장바구니 아이템인지 확인하는 boolean를 활용한다. 추가 버튼을 누르면 boolean 값을 변경해서 저장한다.

3. 시험 시나리오

3.1 로그인

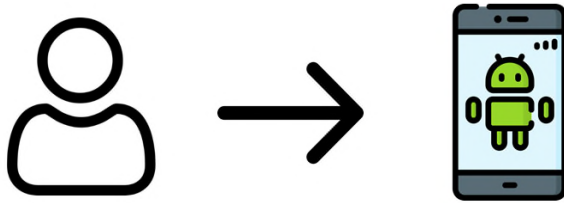
사용자는 애플리케이션에 로그인을 진행한다. 서버 데이터베이스에 저장된 회원 정보일 경우 로그인이 성공하지만 등록된 회원이 아닐 경우 회원 가입을 요청한다. 로그인을 진행한 회원은 게시판에 글을 작성할 수 있으며 작성한 글은 서버의 데이터베이스에 저장된다.



<그림11> 로그인

3.2 사용자가 애플리케이션에 입력

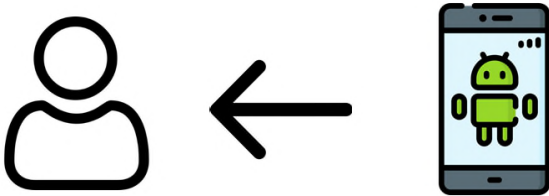
사용자는 식품명을 OCR이나 음성인식을 통하여 등록한다. 인식을 하지 못할 경우에는 텍스트로 입력을 하게끔 한다. 사용자가 자주 구매하는 식품의 경우 즐겨찾기 기능으로 저장 가능하다.



<그림12> 사용자가 애플리케이션에 입력

3.3 사용자가 애플리케이션을 통해 받는 정보

애플리케이션은 사용자가 등록한 식품의 유통기한을 알람을 통해 공지한다. 사용자는 레시피를 검색이 가능하며 필요한 재료와 레시피 검색이 가능하다. 또한 사용자 근처에 있는 푸드뱅크의 위치 정보를 검색할 수 있다.



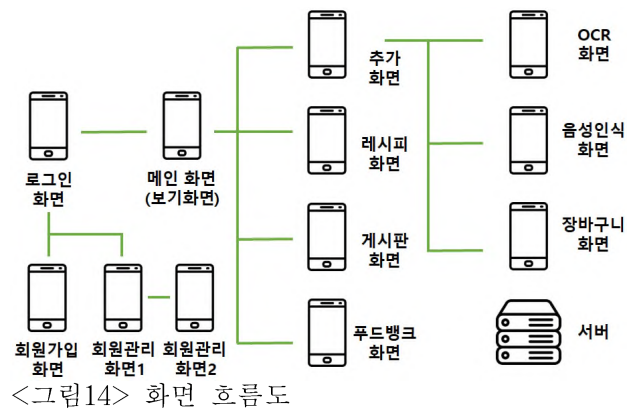
<그림13> 애플리케이션을 통해 받는 정보

3.4 서버 관리자

서버 관리자는 사용자들의 계정 관리와 게시판 관리를 한다. 관리자는 관리자 계정을 생성해 두어 등록된 회원들의 리스트를 보며 관리 가능하다. 그 후 사용자들의 계정 정보를 볼 수 있고 등록된 회원을 삭제할 수 있다.

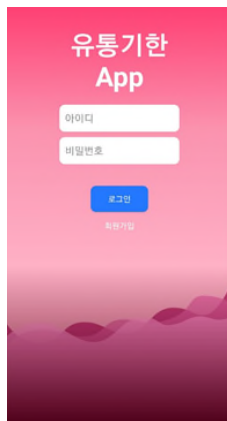
4. 상세 설계

4.1 화면 흐름도

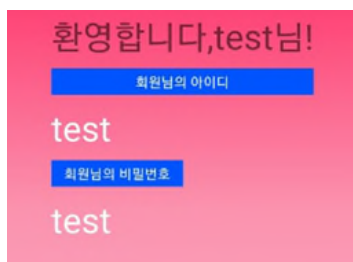


4.2 로그인 화면

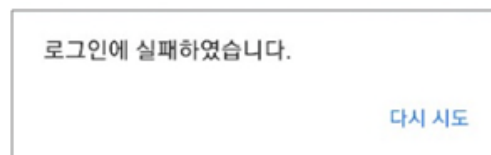
애플리케이션을 처음 작동 시킬 때 <그림15>의 로그인 화면을 사용자에게 보여준다. 로그인 화면의 구성은 상단에 서비스를 제공하는 애플리케이션의 이름을 표시하고 중단에 아이디와 비밀번호 입력을 받는다. 마지막으로 하단에서 로그인버튼과 회원가입 버튼이 있다. 로그인에 성공할 경우 <그림16>의 화면이 출력된다. 잘못된 정보를 입력하고 로그인 버튼을 클릭할 경우 <그림17>과 같이 로그인 실패화면이 뜬다.



<그림15> 로그인 화면



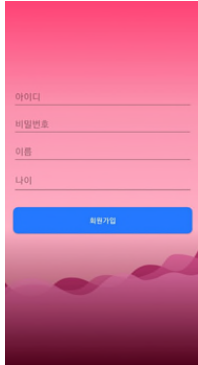
<그림16>로그인 성공화면



<그림17>로그인 실패화면

4.3 회원가입 화면

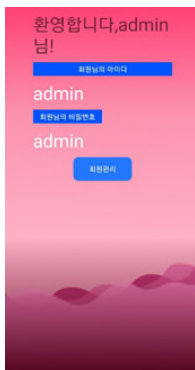
계정이 없는 경우 회원가입 버튼을 클릭하면 <그림18>의 회원가입화면으로 넘어가며 아이디, 비밀번호, 이름, 나이를 입력하여 계정을 생성한다.

A mobile app screen for registration. It has a pink gradient background with a DNA helix at the bottom. The form includes fields for '아이디' (ID), '비밀번호' (Password), '이름' (Name), and '나이' (Age), each with a corresponding label and a blue underline. Below the fields is a blue button labeled '회원가입' (Sign Up).

<그림18> 회원가입 화면

4.3 관리자 화면

로그인한 사용자가 관리자일 경우 <그림19>와 같은 화면이 뜨며 일반사용자와 달리 회원관리 버튼이 있다. 회원관리 화면을 들어가면 등록된 회원리스트가 나오며 회원의 이름을 검색할 수 있으며 삭제가 가능하다.

A mobile app screen for the admin interface. It has a pink gradient background with a DNA helix at the bottom. The text '환영합니다,admin님!' (Welcome, admin!) is at the top. Below it is a blue button labeled '회원님의 아이디' (Your ID). Then, the text 'admin' is displayed, followed by a blue button labeled '회원님의 비밀번호' (Your Password). Below that, the text 'admin' is displayed, followed by a blue button labeled '회원관리' (Manage Members).

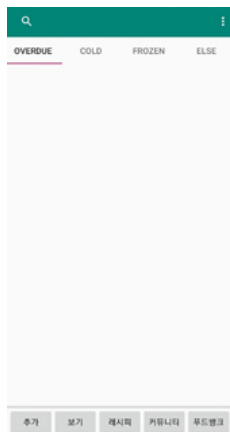
<그림19> 관리자 화면

A mobile app screen showing a list of members and a search bar. The search bar is at the top with the text 'te'. Below it are two rows of member information. Each row has a blue button labeled '삭제' (Delete) and a text label. The first row shows 'test' in blue, 'test' in red, 'jiki' in green, and '20' in purple. The second row shows 'test2' in blue, 'test2' in red, 'ho' in green, and '34' in purple.

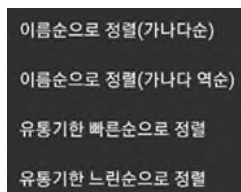
<그림20> 회원리스트 및 검색

4.4 메인 화면

로그인 후 처음에 보여주는 화면이다. 메인 화면에서는 유통기한이 지났는지 확인이 가능하며 등록한 식품이 냉장, 냉동, 상온으로 분류된 것을 볼 수 있다. 사용자는 검색을 하여 등록한 식품을 찾을 수 있으며 이때 이름순이나 유통기한 순으로 식품을 정렬할 수 있다. 그리고 하단부에 버튼을 통하여 다른 화면으로 이동이 가능하다.



<그림21> 메인 화면



<그림22> 등록된 식품 검색 시 정렬 기능

4.5 추가 화면

이 화면에서는 즐겨찾기 버튼과 OCR, STT화면으로 이동이 가능하다. 또한 즐겨찾기를 통하여 사용자가 자주 구매하는 식품을 저장하여 다시 등록하지 않게 한다. 즐겨찾기 버튼을 클릭 시 <그림24>의 다이얼로그가 뜬다. 다이얼로그에서 즐겨찾기 된 식품이 아직 남아 있다면 사용버튼을 클릭하여 식품의 잔여 유무를 선택할 수 있다. 화면 하단부에 있는 장바구니는 OCR과 STT를 통해 식품의 정보를 받을 시 그 정보가 넘어가는 곳이다.



<그림23> 추가 화면



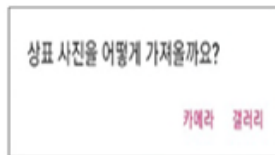
<그림24> 즐겨찾기 다이얼로그

4.6 OCR 화면

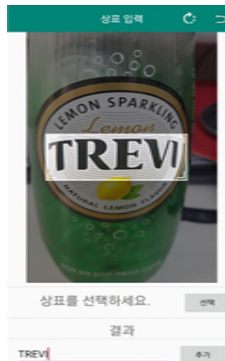
사용자가 OCR 화면에 들어가면 상품명을 카메라 또는 갤러리를 통해 가져올지 묻는 <그림26>의 다이얼로그가 뜬다. 유통기한을 가져오는 다이얼로그도 이와 동일하다. 그 후 <그림27>의 모습이 뜨며 OCR 화면 지정 틀이 나오게 된다. 지정 틀에서는 가져온 사진을 회전하거나 재지정 할 수 있다. 결과를 다 입력한 후에는 <그림28>의 다이얼로그가 떴서 최종확인을 묻는다.



<그림25> OCR 화면



<그림26> OCR에서 출력되는 다이얼로그



<그림27> OCR 지정 틀

4.7 STT 화면

사용자가 STT를 사용할 때 <그림28>에서 녹음 상태를 나타내는 문구로는 '대기 중', '듣는 중', '들음'으로 표시된다. STT사용 후 <그림29>와 같이 뜨며 '이걸 말했나요?'의 문구와 함께 인식된 결과와 비슷한 단어를 보여준다.



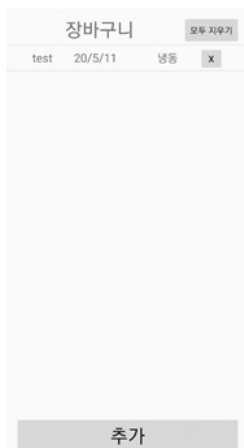
<그림28>STT 화면



<그림29> 인식된 결과와 비슷한 단어 출력

4.8 장바구니 화면

OCR, STT를 통해 들어온 식품 정보는 장바구니 화면으로 이동하게 된다. 추가 화면에서 텍스트로 식품정보를 적을 때도 정보를 입력하면 장바구니 화면으로 이동하게 된다. 장바구니에 담긴 식품은 추가/삭제가 가능하다. 추가 버튼을 클릭하게 되면 핸드폰에 <그림31>과 같은 유통기한 알림이 온다.



<그림30>장바구니 화면



<그림31> 유통기한 알림

4.9 레시피 화면

사용자에게 추천 메뉴를 보여준다. 검색 기능을 제공하고 있으며 원하는 식품을 클릭 시에 해당 메뉴에 필요한 재료들과 레시피를 보여준다.



<그림32> 레시피 화면

4.10 커뮤니티 화면

사용자들이 글을 작성하며 소통할 수 있는 게시판이다. 사용자는 제목과 내용을 적어 글을 작성할 수 있으며 작성한 글에는 다른 사용자가 댓글을 달 수 있다.



<그림33> 게시판 화면



<그림34> 글 작성 화면

4.11 푸드뱅크 화면

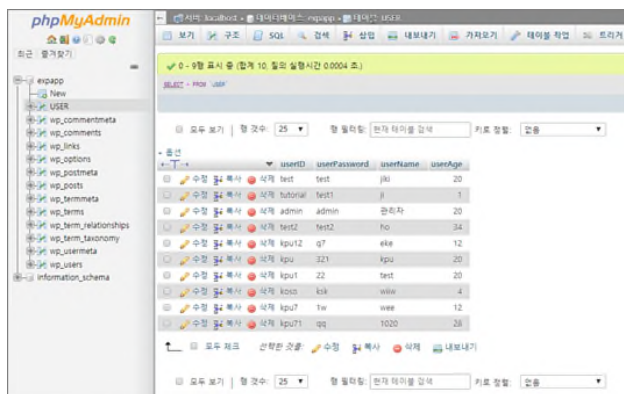
사용자의 위치 기반으로 근처 푸드뱅크를 보여준다. 지점의 위치는 검색을 통하여 알 수 있으며 푸드뱅크 소개 글과 정보들을 알 수 있다.



<그림35> 푸드뱅크 화면

4.12 서버 화면

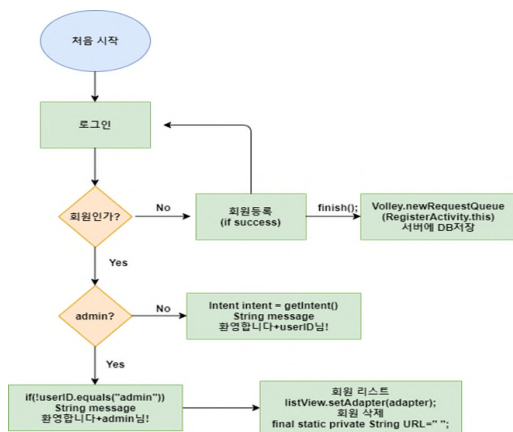
종합설계 프로젝트에 쓰일 서버 환경으로 닷홈에서 무료호스팅을 하여 진행하였다. 현재 만들 애플리케이션에 등록된 유저들 목록을 볼 수 있다. 'USER' 테이블에는 유저ID, 비밀번호, 이름, 나이가 저장된다.



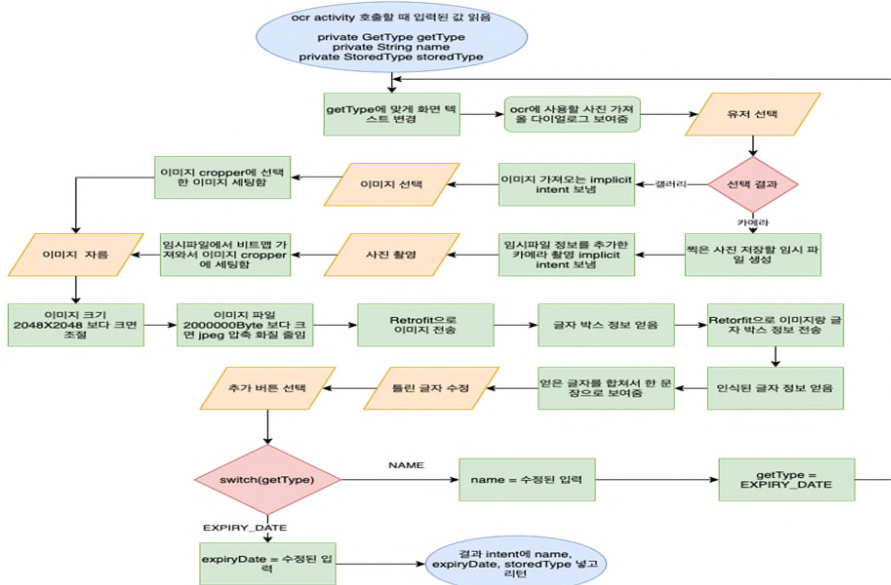
<그림36> 서버 화면

4.13 화면 설계

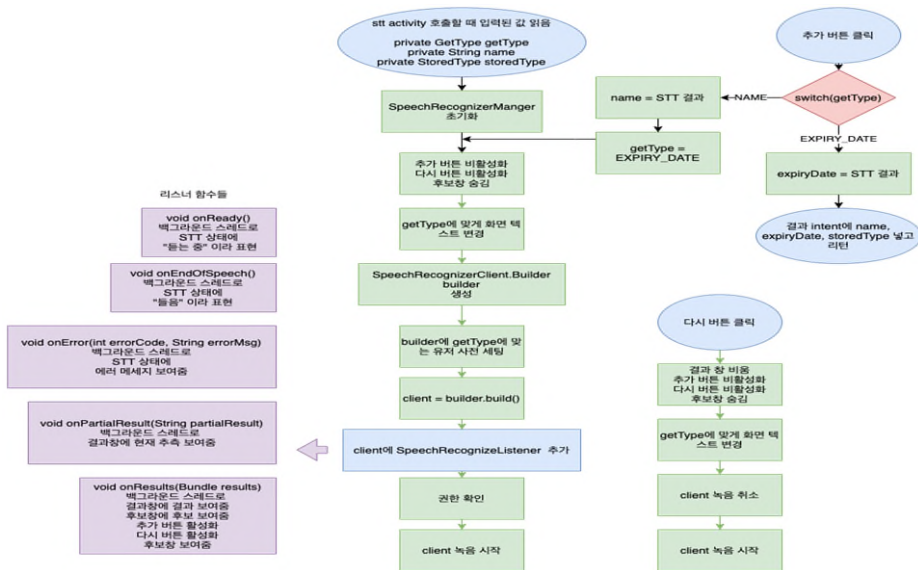
아래의 그림들은 현재 사용되는 화면들의 알고리즘으로 개발하는 애플리케이션의 로직을 알 수 있다.



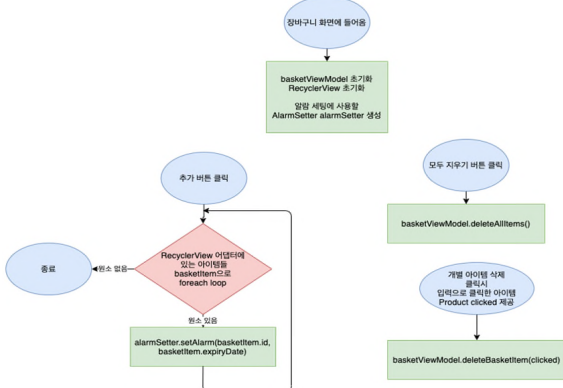
<그림37> 로그인 화면 설계



<그림38> OCR 화면 설계



<그림39> STT 화면 설계



<그림40> 장바구니 화면 설계

4.14 로컬 DB 테이블

로컬 DB에서 데이터 테이블은 <표6>, <표7>, <표8>을 통해 알 수 있다. 테이블에 따로 별표를 통해 표시해둔 것은 Primary Key를 표시해둔 모습이다.

<표6> 식품 정보, 푸드뱅크 DB 테이블

| Favorite | |
|------------|-------------|
| *String | name |
| StoredType | stored |
| Int | expiryAfter |

| FoodBankPlace | |
|---------------|---------|
| *int | id |
| String | name |
| String | address |
| String | phone |

| Product | |
|------------|------------|
| *int | id |
| String | name |
| LocalDate | expiryDate |
| StoredType | stored |
| boolean | inBasket |

<표7>레시피 DB 테이블1

| RecipeInfo | |
|------------|-----------------|
| *int | recipeCode |
| String | recipeName |
| String | recipeBasicDesc |
| Integer | cuisineCode |
| String | cuisineDesc |
| Integer | foodTypeCode |
| String | foodType |
| String | cookTime |
| String | calories |
| String | servings |
| String | skillLevel |
| String | ingredientType |
| String | costToMake |
| String | mainImgUrl |
| String | detailUrl |

<표8>레시피 DB 테이블2

| DislikedRecipe | |
|----------------|----|
| *int | id |

| RecipeProgress | |
|----------------|----------------|
| *int | recipeCode |
| *int | recipeOrder |
| String | recipeDesc |
| String | progressImgUrl |
| String | tip |

| RecipeIngredient | |
|------------------|--------------------|
| *int | recipeCode |
| *int | ingredientOrder |
| String | ingredientName |
| String | ingredientAmount |
| int | ingredientType |
| String | ingredientTypeName |

4.15 레시피 DB

레시피 정보 관련 DB함수와 레시피 과정 관련 DB함수를 <표9>를 통해 알 수 있다.

<표9> 레시피 DB

| 형식 | RecipeInfoRepository |
|----|---|
| 설명 | 레시피 정보 관련 DB 함수 |
| 예시 | <pre>public LiveData<List<RecipeInfo>> getRecommendation() // 현재 가지고 있는 상품 기반으로 레시피 추천 public LiveData<List<RecipeInfo>> getSearched(String searchWord) // 검색된 레시피 정보 보여줌</pre> |
| 형식 | RecipeProgressRepository |
| 설명 | 레시피 과정 관련 DB 함수 |
| 예시 | <pre>public LiveData<List<RecipeProgress>> getRecipe(int recipeCode) // 특정 레시피코드를 갖는 레시피 과정을 가져옴</pre> |

4.16 API 함수

구현한 API 함수에 대해 표를 만들었다. OCR API 함수는 <표10>, 즐겨찾기 API 함수는 <표11>, 장바구니 API 함수는 <표12>, 상품 알람 API 함수는 <표13>, 서버에 저장시킬 정보 API 함수는 <표14>를 통해 확인할 수 있다.

<표10> OCR API 함수

| 형식 | OcrActivity-void onExtractTextArea |
|----|--|
| 설명 | OCR 텍스트 지정틀 |
| 예시 | <pre>public void onExtractTextAreaSuccess(Call<TextExtractionResponse> call, Response<TextExtractionResponse> response) // OCR 텍스트 사각형 성공시 호출 public void onExtractTextAreaFailure(Call<TextExtractionResponse> call, Throwable t) // OCR 텍스트 사각형 실패시 호출</pre> |
| 형식 | OcrActivity-void onRecognize Text |
| 설명 | OCR 텍스트 인식 |
| 예시 | <pre>public void onRecognizeTextSuccess(Call<TextRecognizeResponse> call, Response<TextRecognizeResponse> response) // OCR 텍스트 인식 성공시 호출 public void onRecognizeTextFailure(Call<TextRecognizeResponse> call, Throwable t) // OCR 텍스트 인식 실패시 호출</pre> |
| 형식 | OcrActivity-void ocrQuery(Bitmap, croppedBitmap) |
| 설명 | OCR 쿼리에 필요한 전처리과정(용량,크기 조정) |
| 예시 | <pre>private void ocrQuery(Bitmap croppedBitmap) //크기 조정 후 OCR 쿼리 요청하는 함수</pre> |

<표11> 즐겨찾기 API 함수

| 형식 | FavoriteRecyclerViewAdapter |
|----|---|
| 설명 | 즐거찾기에서 호출되는 인터페이스 |
| 예시 | <pre>void onDeleteClicked(Favorite clickedFavorite) // 즐겨찾기 삭제 클릭시 호출하는 인터페이스 void onStoredChanged(Favorite changedFavorite) // 즐겨찾기에서 저장된 공간 변경시 호출하는 인터페이스 void onDefaultExpiryDateChanged(Favorite changedFavorite) // 즐겨찾기에서 기본 유통기한 변경시 호출되는 인터페이스</pre> |

<표12> 장바구니 API 함수

| 형식 | BasketActivity-void onDeletedClicked |
|----|--|
| 설명 | 장바구니 삭제 클릭 시 호출하는 인터페이스 |
| 예시 | void onDeletedClicked(Product clicked) |
| 형식 | void setFilterString |
| 설명 | 검색어 설정한 후 해당 정보를 리사이클뷰 어댑터에 전달할 때 사용하는 함수 |
| 예시 | void setFilterString(String newFilterString) |

<표13> 상품 알람 API 함수

| 형식 | AlarmSetter-void setAlarm/cancelAlarm |
|----|--|
| 설명 | 상품 알람 설정 및 삭제 |
| 예시 | <pre>public void setAlarm(int productId, LocalDate date) // 알람 설정할 상품 id랑 설정할 시간을 입력으로 알람 설정 public void cancelAlarm(int productId) // 해당 상품 id의 알람 취소</pre> |
| 형식 | NotificationSetter-void setNotification/createNotification |
| 설명 | 상품 정보 알람 설정 |
| 예시 | <pre>void setNotification(Product product) // 상품 정보를 가지고 Notification 설정 public void createNotificationChannel() // 안드로이드 O 이상에서 필요한 notification channel 생성</pre> |

<표14> 서버에 저장시킬 정보 API 함수

| 형식 | db-void downloadData() |
|----|--|
| 설명 | 서버에 저장시킬 정보 |
| 예시 | <pre>boolean isThereFoodBankListUpdate() // 서버에 새로운 푸드뱅크 정보있는지 확인 void downloadNewFoodBankList() //서버에서 새로운 푸드뱅크 정보 다운 int checkServerSyncNumber() // 서버에 저장된 싱크로 숫자 확인 void sendData() // 앱 정보 서버에 전송 void downloadData() // 서버에서 저장된 정보 다운로드</pre> |

4.17 서버에 구축한 테이블

서버에 'USER' 데이터베이스 테이블을 구축했으며 테이블 안에는 유저ID, 비밀번호, 이름, 나이 정보가 담긴다.

<표15> USER 테이블

| 형식 | USER 테이블 |
|----|---|
| 설명 | USER 데이터베이스 테이블 구축 |
| 예시 | <pre>CREATR TABLE 'USER' (userID VARCHAR(20) NOT NULL; userPassword VARCHAR(20) NOT NULL; userName VARCHAR(20) NOT NULL; userAge INT NOT NULL; PRIMARY KEY(userID)); //아이디, 비밀번호, 이름은 문자형으로 받고 나이는 인트형으로 받음</pre> |

4.18 서버와 연동할 때 쓰인 PHP 구문

애플리케이션과 서버의 연동을 위해 PHP언어를 사용하였으며 이때 사용한 구문은 <표 16>, <표17>, <표18>을 통해 알 수 있다. <표18>은 관리자가 회원리스트를 보며 삭제가 가능하도록 한다.

<표16> 로그인 php

| 형식 | Login.php |
|----|--|
| 설명 | DB에 저장되어 있는 회원 정보를 불러온 후 동일할 시 로그인됨 |
| 예시 | <pre> <?php \$con = mysqli_connect("localhost", "expapp", "tksrleo12!", "expapp"); //expapp은 서버 데이터베이스 아이디, tksrleo12!는 데이터베이스 비밀번호 \$stmt = mysqli_prepare(\$con, "SELECT * FROM USER WHERE userID = ? AND userPassword = ?"); //특정 아이디와 특정 비밀번호에 해당하는 회원이 존재하는지 검색 while(mysqli_stmt_fetch(\$stmt)) { \$response["success"] = true; \$response["userID"] = \$userID; \$response["userPassword"] = \$userPassword; \$response["userName"] = \$userName; \$response["userAge"] = \$userAge; } echo json_encode(\$response); ?> //그 결과가 각각의 response로 담겨서 안드로이드 프로그램에서 보게됨 </pre> |

<표17> 회원 등록 php

| 형식 | Register.php |
|----|--|
| 설명 | 회원등록 |
| 예시 | <pre> \$userID = \$_POST["userID"]; \$userPassword = \$_POST["userPassword"]; \$userName = \$_POST["userName"]; \$userAge = \$_POST["userAge"]; //각각의 회원 정보를 받음 \$stmt = mysqli_prepare(\$con, "INSERT INTO USER VALUES (?, ?, ?, ?);"); //회원 정보를 USER 테이블에 삽입 \$response = array(); \$response["success"] = true; //response에 성공 여부가 담긴 true값을 넣어 줌으로써 안드로이드 프로그램에서 회원 정보가 등록된 것을 볼 수 있음 </pre> |

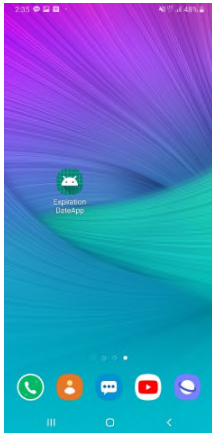
<표18> 회원 리스트 및 삭제 php

| 형식 | List.php |
|----|---|
| 설명 | 데이터베이스 접속한 이후에 모든 유저를 가져옴 |
| 예시 | <pre> while(\$row = mysqli_fetch_array(\$result)){ array_push(\$response, array("userID"=>\$row[0], "userPassword"=>\$row[1], "userName"=>\$row[2], "userAge"=>\$row[3])); } //유저 정보를 가져올 때 ID, Password, 이름, 나이 순으로 차례대로 뽑아옴 echo json_encode(array("response"=>\$response)); //뽑아온 정보를 response라고 이름붙여서 가져옴 </pre> |
| 형식 | Delete.php |
| 설명 | 데이터베이스에서 특정 회원 삭제 |
| 예시 | <pre> \$userID = \$_POST["userID"]; //매개변수로 넘어온 userID값을 받음 \$stmt = mysqli_prepare(\$con, "DELETE FROM USER WHERE userID = ?"); //해당 userID를 가진 회원을 삭제 \$response["success"] = true; //success라는 이름의 response변수는 true값을 가지고 반환됨 </pre> |

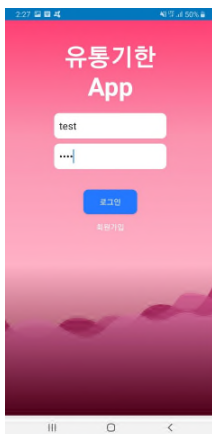
5. Prototype 구현

5.1 로그인 진행

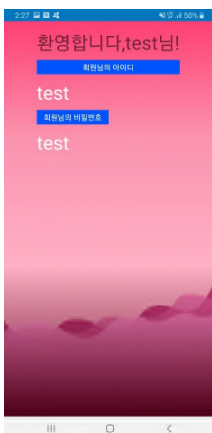
<그림41>의 애플리케이션 아이콘을 누르면 <그림42>와 같은 화면이 나온다. 여기에 test라는 회원을 임의로 만들어서 로그인을 진행 <그림43>과 같은 화면이 출력된다.



<그림41> 애플리케이션 아이콘



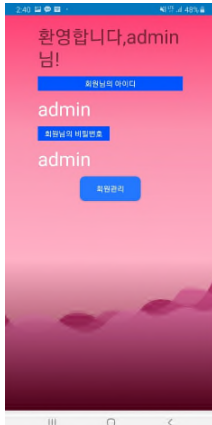
<그림42> 로그인 일반유저 화면



<그림43> 로그인 시 출력되는 화면

5.2 관리자의 회원관리

관리자의 경우는 ID를 admin으로 설정을 해두었다. admin으로 진입할 경우 <그림43>과는 다르게 화면에 회원관리 버튼이 출력됨을 볼 수 있다. 회원관리 버튼을 누르면 <그림 45>와 같은 화면이 나오며 여기서는 애플리케이션에 회원가입 된 유저들의 아이디, 비밀번호, 이름, 나이를 볼 수 있으며 유저를 검색하여 삭제가 가능하다.



<그림44> 관리자 로그인 시 출력되는 화면



<그림45> 회원 관리화면

5.3 메인 화면

로그인을 진행 후 유저가 처음 보게 되는 화면이다. 이곳에서는 유통기한이 지난 식품을 확인 가능하며 냉장, 냉동, 식품, 상품분류가 가능하다. 또한 등록된 식품이 많을 경우 찾기 쉽게 화면 상단부에 검색 기능을 넣었다. 화면 상단 오른쪽 부분을 클릭하게 되면 유저가 등록한 식품들을 이름순과 유통기한순으로 정렬이 가능하다. 화면 하단부에 보이는 '추가', '레시피', '커뮤니티', '푸드뱅크' 버튼을 클릭하여 다른 화면으로의 이동이 가능하다. 메인화면의 명칭은 '보기'로 하였다.



<그림46> 메인 화면

5.4 추가 화면과 장바구니 화면

추가화면은 사용자가 자주 등록하는 상품의 경우 번거롭게 다시 등록하는 것을 방지하고 자 즐겨찾기 기능이 있다. 즐겨찾기 된 식품이 아직 있는 경우 사용버튼을 클릭하여 주고 식품을 섭취하였으면 없음 버튼으로 바꾸어주면 된다. 사용자는 OCR이나 음성인식을 통하여 상품을 등록할 수 있는데 이때 추가화면에 있는 버튼들을 통하여 이동 가능하다. 입력이 잘 이루어지지 않을 경우 텍스트를 통해 입력해도 등록이 이루어지게 구현하였다. 화면 하단부에 보이는 빨간색 장바구니 모양의 버튼은 OCR과 음성인식을 통하여 저장한 정보가 담기는 곳이다. 즉, 장바구니화면은 사용자가 상품을 등록한 정보가 맞는지 화면으로 이곳에서 최종적으로 추가를 하여야 상품이 등록된다. 최종적인 추가가 이루어지면 <그림31>처럼 애플리케이션에 사용자에게 등록한 상품의 유통기한 알림이 온다.

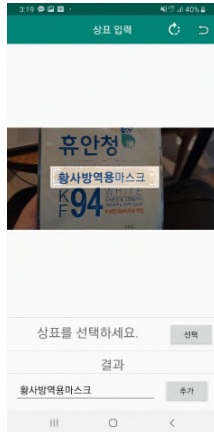


<그림47> 즐겨찾기 버튼 클릭 시

5.5 애플리케이션에 추가한 기능을 이용한 상품 등록

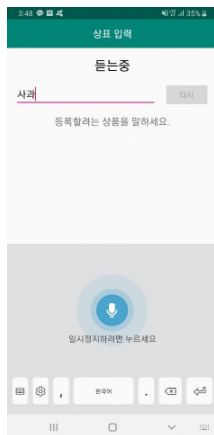
OCR을 이용할 경우 사용자가 등록을 원하고자 하는 상품을 카메라로 촬영하거나 이미 사진에 저장되어있다면 갤러리에서 가져올 수 있다. <그림48>은 상품을 사진으로 촬영한 모습이다. 이 화면에서 OCR화면 지정 틀을 조정하여 상품의 이름이 있는 부분을 인식하게끔 한다. 유통기한 입력도 이와 동일하게 이루어지며 추가를 하면 등록한 정보가 장바구니

로 이동하게 된다. <그림48>은 평평한 상품의 글씨를 인식한 것이지만 상세설계 시 <그림 27>처럼 굴곡진 페트병에 영어로 적힌 글자의 경우에도 제대로 인식하는 모습을 확인하였다.



<그림48> OCR 화면에서 지정틀 사용

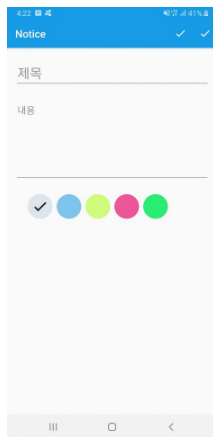
음성인식을 이용할 경우에는 <그림49>를 통해 확인할 수 있다. 화면 상단부에 ‘듣는 중’ 표시를 통해 지금 음성인식을 하는 중이라는 것을 알 수 있다. 이 경우도 OCR과 같이 상품명 등록한 후 저장된 정보가 장바구니화면으로 넘어가게 된다.



<그림49> 음성인식중인 모습

5.6 커뮤니티 화면

사용자는 <그림50>의 모습처럼 제목과 내용을 적어 게시글을 적을 수 있다. 글을 적고 나면 서버에 게시글이 저장되는 모습을 <그림51>의 화면을 통해 알 수 있다. 게시글을 적을 때 5가지 색상 중 하나를 선택하면 게시되는 글의 테두리와 내부가 그 색상으로 기재된다. 서버에서 글의 제목과 내용, 어떤 색상을 사용하여 저장하였는지, 언제 작성하였는지를 볼 수 있다.



<그림50> 게시글 화면

| + 옵션 | | | | | | |
|--------------------------|----|----|----|-------|------------|------------|
| | | | id | title | note | color |
| <input type="checkbox"/> | 수정 | 복사 | 삭제 | 1 | 오우 | 테스트12 |
| <input type="checkbox"/> | 수정 | 복사 | 삭제 | 2 | 오우 | 테스트12 |
| <input type="checkbox"/> | 수정 | 복사 | 삭제 | 3 | 오우 | 테스트 |
| <input type="checkbox"/> | 수정 | 복사 | 삭제 | 4 | Title | 오우 |
| <input type="checkbox"/> | 수정 | 복사 | 삭제 | 5 | Test | 오우 |
| <input type="checkbox"/> | 수정 | 복사 | 삭제 | 6 | Test | 오우 |
| <input type="checkbox"/> | 수정 | 복사 | 삭제 | 7 | 아으 | 오우 |
| <input type="checkbox"/> | 수정 | 복사 | 삭제 | 8 | 아으 | 오우 |
| <input type="checkbox"/> | 수정 | 복사 | 삭제 | 9 | 아으 | ㅇㅇㅇ |
| <input type="checkbox"/> | 수정 | 복사 | 삭제 | 10 | 어어 | 어어 |
| <input type="checkbox"/> | 수정 | 복사 | 삭제 | 11 | ㅇㅇ | ㅇㅇㅇ |
| <input type="checkbox"/> | 수정 | 복사 | 삭제 | 12 | ㅋㅋㅋ | ㅋㅋㅋ |
| <input type="checkbox"/> | 수정 | 복사 | 삭제 | 13 | fgdfg | sdfasdf |
| <input type="checkbox"/> | 수정 | 복사 | 삭제 | 14 | Does it go | does it go |

<그림51> 서버에 게시글 저장

5.7 레시피 화면

레시피 화면에는 <그림52>와 같은 모습이 출력된다. 레시피를 알고 싶은 음식사진을 클릭하면 주재료와 만드는 방법을 사용자에게 알려준다. 그리고 검색 기능을 넣어 원하는 음식을 보다 빠르게 찾을 수 있다.



<그림52> 음식 리스트

6. 시험/ 테스트 결과

6.1 테스트 대상 내용 소개

5장 프로토타입 파트에서 이미 구현을 마치고 동작결과를 보여 5장을 기반으로 테스트 수행 내용을 기반으로 작성한다.

| 단계 | 시나리오 |
|---------|---|
| 1(로그인) | 사용자는 로그인을 진행한다. |
| 2(유통기한) | 상품명과 유통기한을 OCR과 STT로 받아들이 저장하면 사용자에게 유통기한 알림이 온다. |
| 3(즐거찾기) | 즐거찾기된 식품목록을 볼 수 있다. |
| 4(게시판) | 사용자 게시판 등록 및 삭제를 진행한다. |
| 5(레시피) | 레시피 정보를 확인한다. |
| 6(푸드뱅크) | 사용자 주변 푸드뱅크 위치 확인을 해본다. |
| 7(서버) | 서버에 사용자들의 회원가입 정보와 게시글이 올라와있는지 확인해본다. |

6.2 기능별 테스트 결과

| 수행 단계 1(로그인) | 결과 |
|--------------|---|
| 1. 회원가입 진행 | 아이디, 비밀번호, 이름, 나이를 적으면 회원가입이 완료된다. |
| 2. 로그인 | 1. 아이디나 비밀번호가 틀릴 경우, 로그인에 실패하였다는 문구가 출력된다. 2. 로그인 성공 시 환영한다는 문구가 출력된다. |
| 3. 관리자 로그인 | 관리자는 로그인 시 미리 만들어 둔 admin계정으로 접속한다. |
| 4. 회원관리 | 1. 관리자로 로그인될 경우에만 회원관리 테이블이 나온다. 2. 이곳에서 등록된 회원들을 검색 및 삭제가 가능하다. |

| 수행 단계2(유통기한) | 결과 |
|--------------|--|
| 1. OCR 사용 | 1. 식품명과 유통기한을 애플리케이션에서 안내하는 단계에 맞춰 촬영을 하거나 저장된 사진이 있다면 갤러리에서 불러온다. 2. OCR 화면 지정틀로 인식을 원하는 부분을 지정한다. 3. 모든 인식을 완료하면 최종확인용으로 다이얼로그가 출력되는데 '추가'를 누르면 된다. 4. 장바구니 화면으로 가서 OCR을 통해 담긴 정보를 추가/삭제가 가능하며 '추가'할 시 사용자에게 유통기한 알림이 온다. |
| 2. STT 사용 | 1. 음성인식 설정 허용한 후 등록하고자 하는 식품명을 말한다. 2. 모든 인식을 완료하면 최종확인용으로 다이얼로그가 출력되는데 '추가'를 누르면 된다. 3. 장바구니 화면으로 가서 STT를 통해 담긴 정보를 추가/삭제가 가능하며 '추가'할 시 사용자에게 유통기한 알림이 온다. |
| 3. 텍스트 입력 | OCR과 STT를 이용하기 번거로울 시 텍스트로 저장도 가능하다. |
| 4. 저장된 식품 정보 | 저장된 식품 정보는 냉장, 냉동, 그 외 테이블에서 볼 수 있다. |

| 수행 단계3(즐거찾기) | 결과 |
|--------------|---|
| 1. 즐거찾기 추가 | 1. 사용자가 자주 구매하는 식품의 경우 계속해서 정보를 입력하면 번거로우므로 즐거찾기를 통해 저장 가능하다. 2. 사용/없음 버튼이 있어 등록한 식품의 재고 여부를 확인하게끔 할 수 있다. |

| 수행 단계4(게시판) | 결과 |
|-------------|--|
| 1. 글 작성 | 1. 사용자는 게시글의 제목과 내용을 기재한다. 2. 기재한 글의 바탕색을 회색, 파랑색, 노랑색, 빨간색, 초록색으로 지정하여 설정할 수 있다. |
| 2. 글 목록 | 1. 게시글을 작성하고 나면 게시한 내용과 날짜가 적힌 모습으로 게시 테이블이 입력된다. |

| 수행 단계5(레시피) | 결과 |
|-------------|--|
| 1. 레시피 검색 | 원하는 레시피를 검색해본다. |
| 2. 정보 | 1. 원하는 레시피의 사진을 찾으면 눌러본다. 2. 주재료와 만드는 방법이 사용자에게 출력된다. |
| 3. 레시피 연동부분 | 유통기한이 임박한 재료 순서로 레시피를 보여준다. |

| 수행 단계6(푸드뱅크) | 결과 |
|--------------|--|
| 1. 사용자 권한 | 사용자에게 권한 받는 알림을 띄운 후 허락을 받는다. |
| 2. 푸드뱅크 위치 | 1. 현재 위치를 기반으로 푸드뱅크를 보여준다. 2. 푸드뱅크의 경우 지점이 많지 않아 미리 마커로 지정해두었다. 3. 사용자 주변에 마커된 지점을 클릭하면 지점명과 전화번호를 알 수 있다. |

| 수행 단계7(서버) | 결과 |
|------------|--|
| 1. 회원가입 정보 | 1. 회원가입 시 사용자들이 입력한 정보가 제대로 들어와 있는지 확인을 해본다. 2. 회원가입된 유저를 삭제가 가능하다. |
| 2. 게시글 정보 | 1. 사용자들이 입력한 글 정보를 확인 가능하다. 2. 게시글의 내용을 수정할 수 있으며 게시글의 바탕색 수정도 가능하다. 3. 게시글을 삭제할 수 있다. |

7. Coding & DEMO

7.1 애플리케이션에 사용된 코드

아래의 깃허브 주소를 들어가면 각 기능마다 사용된 코드를 확인할 수 있다.

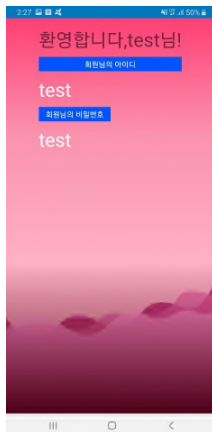
<https://github.com/jiki94/CapstoneDesign/tree/master/app/app/src/main/java/com/example/expirationdateapp>

7.2 데모 진행 절차

상품을 하나 구매하여 다음의 순서를 통해 진행해보도록 한다.

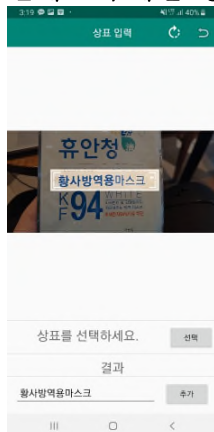
1. 앱에 로그인을 한다.
2. 구매한 상품을 OCR과 STT를 통해 저장한다.
3. 즐겨찾기 기능을 통해 따로 정보를 받아들이지 않고도 저장됐는지 확인해본다.
4. 유통기한 알림이 오는지 확인해본다.
5. 게시판에 글 작성이 가능한지 확인해본다.
6. 레시피를 확인해본다.
7. 사용자 주변 푸드뱅크를 확인해본다.
8. 서버에 회원정보와 게시글이 저장됐는지 확인해본다.

단계1. 앱에 로그인을 한다.

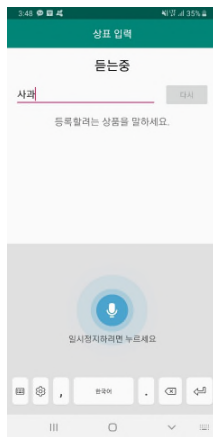


<그림43> 로그인 시 출력되는 화면

단계2. 구매한 상품을 OCR과 STT를 통해 저장한다.



<그림48> OCR 화면에서 지정틀 사용



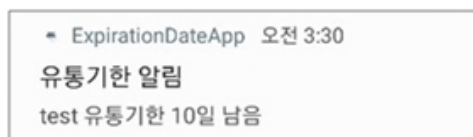
<그림49> 음성인식중인 모습

단계3. 즐겨찾기 기능을 통해 따로 정보를 받아들이지 않고도 저장됐는지 확인해본다.



<그림23> 추가 화면

단계4. 유통기한 알림이 오는지 확인해본다.



<그림31> 유통기한 알림

단계5. 게시판에 글 작성이 가능한지 확인해본다.



<그림50> 게시글 화면

단계6. 레시피를 확인해본다.



<그림52> 음식 리스트



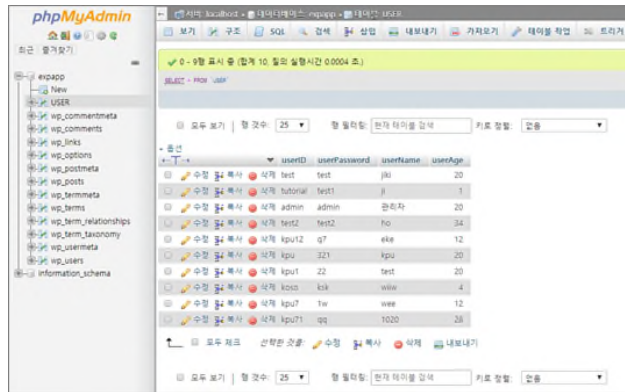
<그림54> 만드는 방법

단계7. 사용자 주변 푸드뱅크를 확인해본다.



<그림55> 마크 표시한 푸드뱅크 지점

단계8. 서버에 회원정보와 게시글이 저장됐는지 확인해본다.



<그림36> 서버 화면



<그림51> 서버에 게시글 저장

III. 결론

1. 연구 결과

여러 마트에서 각종 관측 행사를 진행하는 지금, 소비자들은 처음 구매하고자 했던 계획보다 조금 더 많은 상품을 구매하는 경우가 생기게 된다. 이런 일들이 발생하면 상품의 유통기한을 관리하기가 다소 어렵다. 또한 개발 과정에서 알게 된 점은 소비자들이 상품의 유통기한에 상당히 민감하다는 점이었다. 하지만 상품의 유통기한을 일일이 확인하기는 어렵고 결국 유통기한이 지나 버리는 상품들이 생긴다. 이를 방지하기 위해 유통기한 관리 어플리케이션이 필요하다는 것에 초점을 맞추었고 개발을 진행하였다. 기존 시장에 있던 어플리케이션은 상품을 어플리케이션에 입력하는 방법이 다양하지 않았고, 단순히 유통기한을 입력 후 알람을 주는 방식에 그쳤다. 단순히 유통기한 관리만을 위한 어플리케이션이라면 사용자에게 별로 매력적이지 않을 것이라고 판단을 하였고, 여러 기능을 추가하여 사용자에게 유익하며 재미있는 기능들을 추가하여 진행하였다. 이를 토대로 개발된 어플리케이션은 상품을 입력할 때 텍스트를 통한 입력과 더불어 OCR방식과 STT방식을 제공한다. 추가 기능으로 여유 식품이 발생하여 이를 처리하는 방법을 식품 커뮤니티를 만들어 사용자간 소통을 하게 만들었고, 푸드뱅크의 위치를 제공하여 기증을 하는 방법을 제시하였다. 또한 식품을 이용한 각종 레시피를 제공하여 사용자가 무엇을 먹을지 고민하는 것을 다소 해결하였다.

마지막으로 개발한 어플리케이션은 사용자들이 유통기한 관리 기능만 있다면 사용하는 이

유가 없다고 판단하여 여러 기능을 추가하였지만, 좀 더 매력적인 기능이 추가될 필요가 있다. 애플리케이션에 등록된 상품을 커뮤니티 나눔을 통해 교환을 하거나, 푸드뱅크에 기증할 때마다 인증을 하면 포인트를 주는 식으로 해서 사용자에게 애플리케이션을 사용하면 이득이 되는 시스템을 개발하면 보다 발전된 형태로 나아갈 것으로 보여진다. 이러한 기능들을 추가하여 향후 연구개발을 진행할 예정이다.

2. 작품제작 소요재료 목록

어플리케이션 개발에 있어 따로 재료를 구입하거나 SW를 구매하는데 발생하는 비용은 없습니다.

참고자료

1. Android

URL: <https://developer.android.com/>

2. MySQL

URL: <https://dev.mysql.com/doc/>

3. PHP

URL: <https://www.php.net/docs.php>

4. OCR, STT

URL: <https://developers.kakao.com/>