

2024년도

졸업작품 최종 보고서

노인분들을 위한 식당 예약 서비스
안드로이드 앱 개발 프로젝트



MA

MOBILE APPLICATION

2201781 정유리 (조장)

2201663 권소희

2201715 변지현

2201761 이현진

목 차

1장 프로젝트 개요

1.1 제안 배경

1.2 기대 효과

1.3 애플리케이션 개요

2장 소프트웨어 아키텍처

2.1 소프트웨어 구성도

2.2 SWOT 테이블 분석

3장 주요기술

3.1 안드로이드 서버 및 데이터베이스 구축

3.2 예약 관리 시스템

4장 테스트 결과

5장 결과 분석

5.1 결과 분석 및 개선점

5.2 참고 문헌

1장 프로젝트 개요

1.1 제안 배경

급격히 증가하는 고령화 인구와 이들이 겪고 있는 일상 생활의 불편함을 해소하는 데 중점을 둔다. 스마트 기술이 발전하는 만큼 고령화 시대에 디지털 환경에 적응하지 못하는 디지털 문맹 세대인 노년층이 증가하고 있다. 현대 사회에서 노인들은 경제적으로 독립적인 생활을 유지하고 있지만, 디지털 격차로 인해 다양한 서비스에 쉽게 접근하지 못하는 경우가 많다. 기술의 발전에 따라 예약 시스템의 편리성에 대한 중요성이 강조되면서 모바일 앱을 통한 예약을 활성화하는 업체가 증가하는 추세다. 앱 이용자가 증가할수록 앱을 통해 노쇼를 방지하고 효율적인 식당 운영을 위해 업체 역시 예약 서비스를 제공하는 모바일 앱을 사용할 수밖에 없기 때문이다.

하지만 현 시점에서 널리 도입된 키오스크의 경우만 보더라도 노년층은 키오스크 이용에 아직 익숙하지 않으며, 기존의 식당 예약 서비스 어플리케이션은 청년층을 주 대상으로 설계되어 있어 노년층이 사용하기에 어려운 부분이 많은 상황이다. 실제 최근 보건복지부가 발표한 ‘2023년 노인실태조사’에 따르면 고령층의 스마트폰 보유율은 76.6%에 이르렀으나 절반을 넘는 비율이 여전히 디지털 사회 적응에 어려움을 느끼고 있는 것으로 파악되었다. 디지털 취약 계층인 고령층 인구가 증가함에 따라 고령자 편의를 위한 앱의 지원이 더욱 필요할 것으로 예측된다. 앞으로 기술의 발전은 고령자를 포함한 전 연령층이 사용할 수 있는지에 따른 고려가 필요한 전망이다.

따라서 노년층을 위해 특별히 개발된 식당 예약 서비스를 제공하는 모바일 앱의 필요성을 느껴 개발하게 되었다.

이 앱은 큰 글씨로 인한 가독성을 고려했으며, 조작이 쉬운 인터페이스, 직관적인 메뉴 등을 갖춘 사용자 친화적 인터페이스를 제공하여 노인들이 디지털 서비스에 대한 거부감을 느끼지 않도록 설계되었다.

따라서 이 앱은 노년층이 쉽게 사용할 수 있도록 직관적인 인터페이스와 간단한 예약 과정을 제공하여 노년층 역시 식당 예약 서비스를 자유롭게 사용하는 데 나아가, 향후 생활 속에 전반적으로 자리잡을 수 있기를 기대한다. 해당 서비스가 디지털 격차를 줄이는 시초가 되기를 소망한다.

1.2 기대효과

노년층을 위한 식당 예약 서비스 어플리케이션이 상용화될 경우, 기대 효과는 다음과 같다. ‘디지털 문맹’에 관련된 문제가 갈수록 심화되고 있어 디지털 기기에 익숙하지 않은 고령층의 소외감이 증폭하는 현 상황에서 노년층이 사용할 수 있도록 배려하여 설계된 간단하고 직관적인 인터페이스는 디지털 기기를 쉽게 다루어 해당 서비스를 사용하는 목적에 쉽게 도달하도록 돕는다. 예약이 우선시되는 업체가 증가하는 추세에 따라 전화 예약조차 어려움을 겪을 수 있는 노년층은 해당 앱 서비스를 이용하여 편리한 온라인 예약 서비스를 경험할 수 있다.

또한, 기존 식당 예약 서비스 앱의 단점인 예약을 하는 시점에서 자신이 원하는 좌석을 선택하지 못하는 부분을 보완시켜 타 서비스로부터 차별화되는 강점을 지니며, 이는 앱을 이용하는 소비자의 구체적인 요구사항을 업체가 받아들임으로써 크게 나아가 노년층 인구가 큰 비중을 차지하는 지역 내 상권 활성화를 기대해볼 수 있다.

1.3 애플리케이션 개요

애플리케이션의 이름은 “food with old”로, 노인들이 외식산업에 쉽게 접근할 수 있다는 의미를 지니고 있습니다.

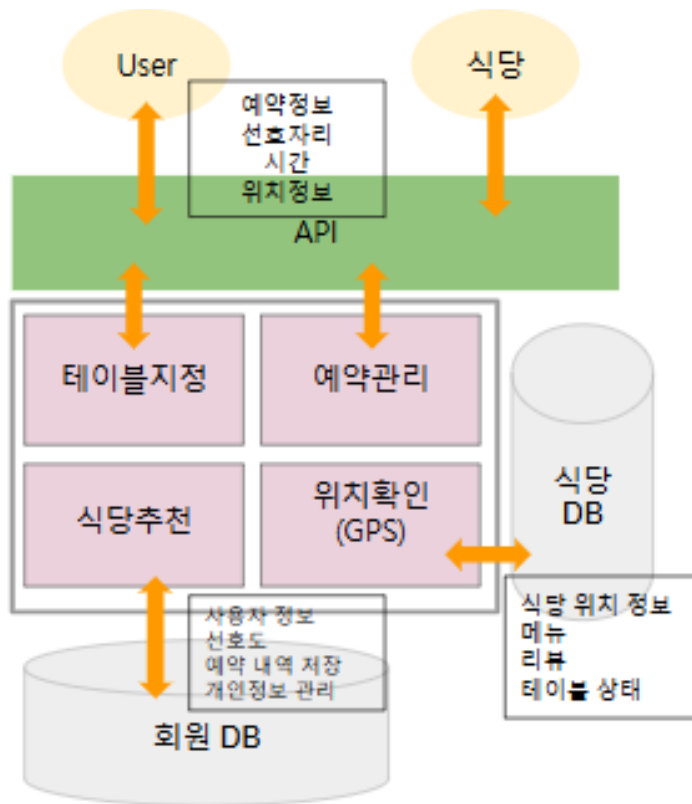
포인트 컬러를 주황색으로 지정한 이유는 노인분들에게 한눈에 잘 보이는 짙은 색감과, 안정감과 편안함을 동시에 제공하기 위한 의도가 담겨있습니다. 서브 컬러 초록색 또한 눈의 피로감을 덜어주는 색으로 앱을 사용하는 동안은 최대한 안정감과 편안함을 제공하고 휴대폰 사용으로 지친 눈의 피로를 덜어줄 수 있습니다.

앱의 로고는 식당을 대상으로 개발한 앱이라는 점을 드러내기 위하여 요리사를 연상하는 조리모를 로고에 드러내어 로고 디자인을 설계하였습니다.



2장 시스템 아키텍처

2.1 소프트웨어 구성도



2.2 SWOT 테이블 분석



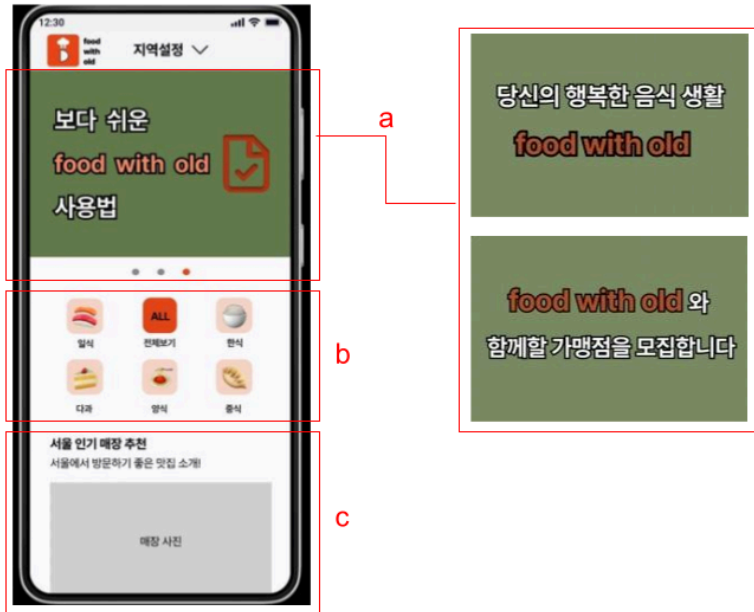
3장 주요 기술

1. 첫 화면



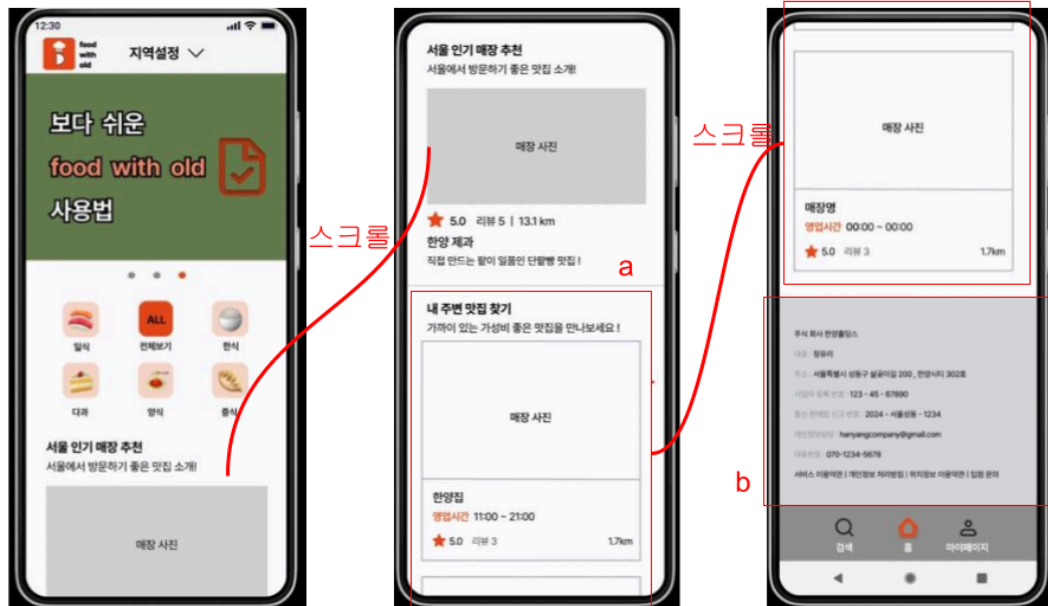
첫 화면은 앱의 로고를 강조하며 시작한다.

2. 홈 화면 1



- a. 광고나 추천 식당 등 여러 배너들을 슬라이드 형식으로 나열하였다.
- b. 식당을 음식 카테고리 별로 나누어 정렬한다. 이때 한 눈에 알아보기 쉽게 이모티콘으로 아이콘 설정을 한다.
- c. 가장 인기있는 식당 등을 추천해주는 배너로 맛집을 더 알기 쉽게 찾고 알아볼 수 있다.

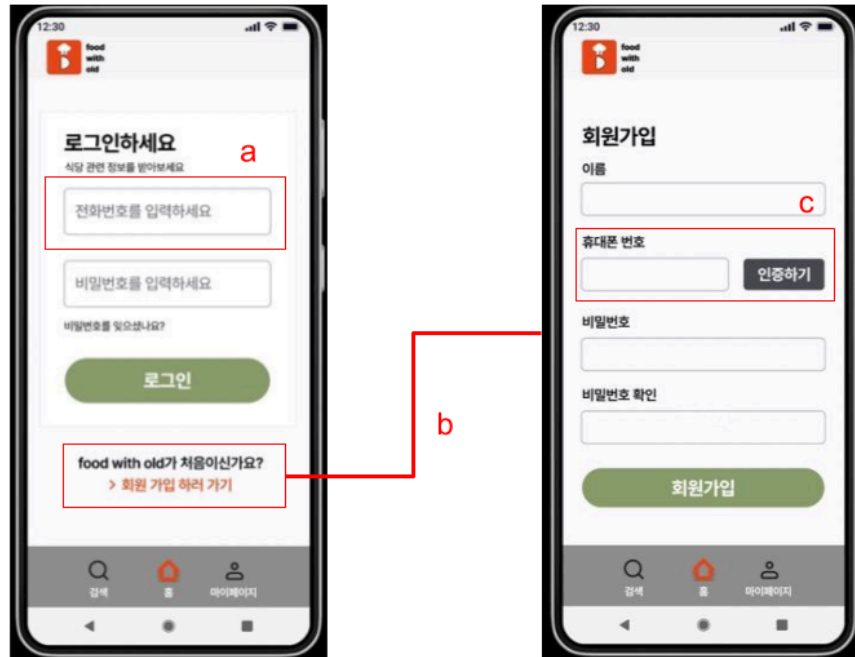
3. 홈 화면2



a. 내 주변의 맛집들을 추천해준다.

b. 스크롤을 마지막까지 하면 앱의 각종 서비스 약관, 개인정보 처리 방침, 위치정보 등이 나타난다.

4. 로그인 및 회원가입



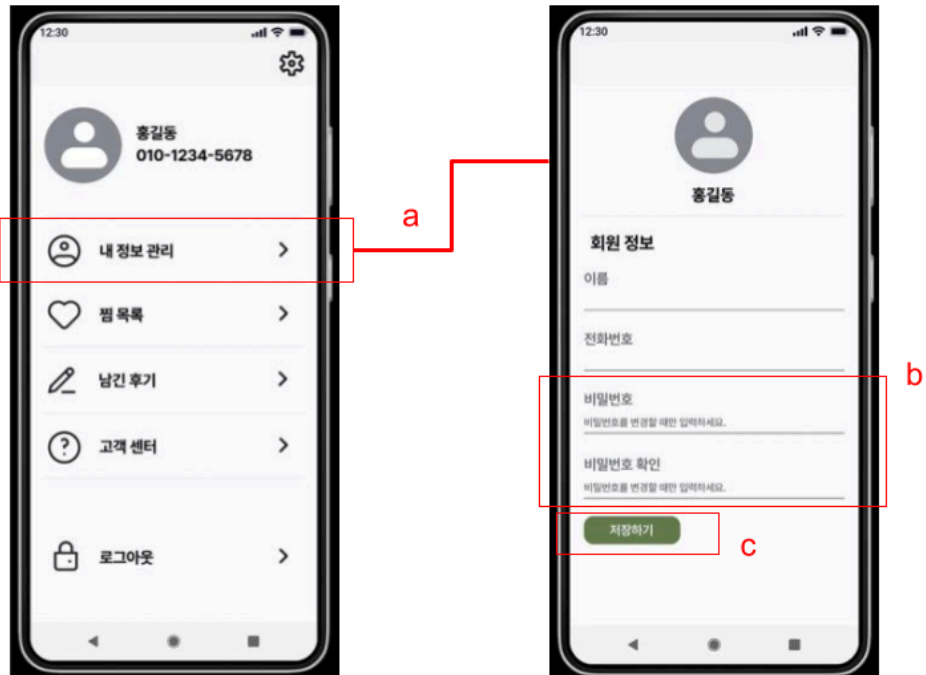
- a. 아이디 대신 전화번호로 로그인하여 정보가입이나 가능한 아이디 확인 등 노인분들이 어려워하시는 단계들을 줄였다.
- b. 앱을 처음 사용하시면 간단한 회원가입을 통해 계정을 생성할 수 있다.
- c. a단계와 마찬가지로 아이디 만드는 칸을 제거하고 휴대폰 번호와 비밀번호만으로 회원가입을 할 수 있다.

5. 검색 화면



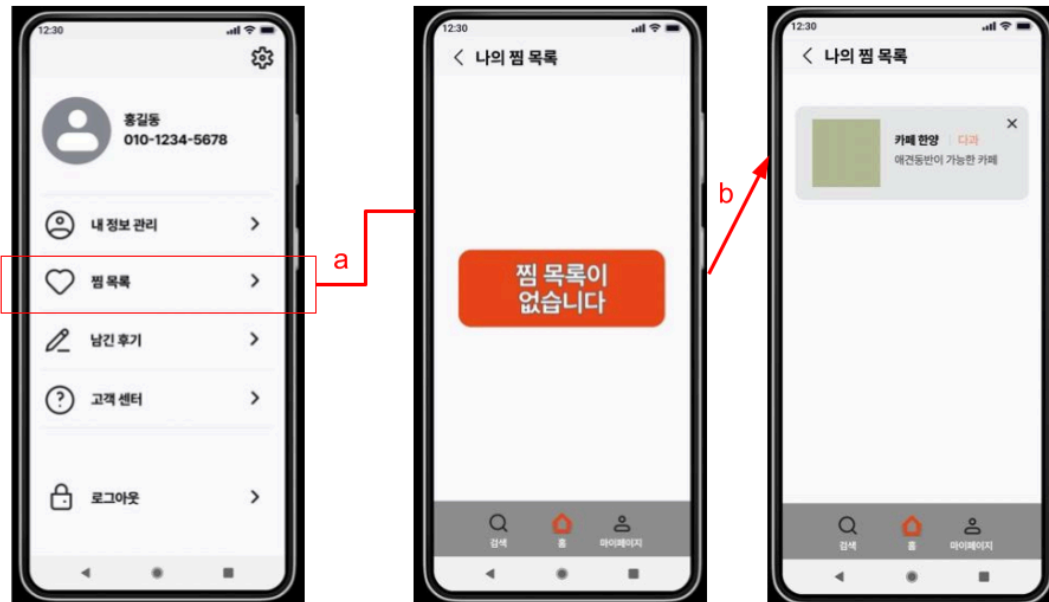
- a. 검색어를 입력하면 관련 식당들이 뜬다. 또, 음성인식 아이콘을 누르면 음성인식으로 편하게 검색 할 수 있다.
- b. 인기 검색어 키워드를 제공해주어 더욱 편리한 검색을 할 수 있다.
- c. 최근 검색한 검색어 들을 기억하여 저장한다.
- d. 검색후 방금 입력한 검색어들을 기억하여 저장한다.

6. 마이 페이지 - 내 정보 관리



- a. 내 정보 관리를 클릭하면 회원 정보를 입력,수정 할 수 있다.
- b. 비밀번호를 변경 할 때에는 한 번더 확인한다.
- c. 정보를 입력, 수정을 마쳤다면 저장을 누르면 저장이 된다.

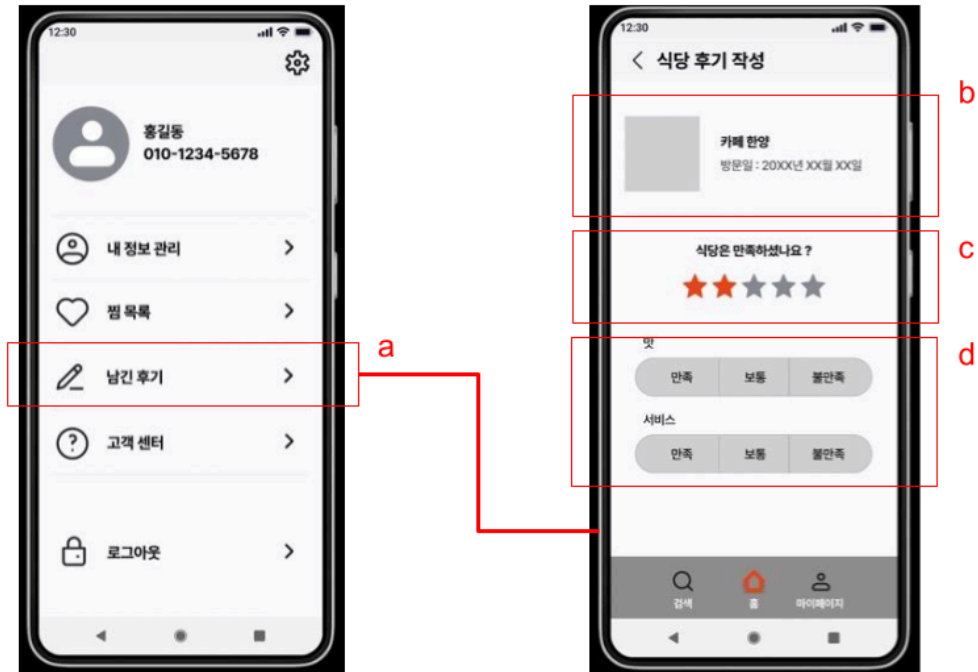
7. 마이 페이지 - 찜 목록



a. 찜 목록을 클릭하면 나의 찜 목록을 볼 수 있다.

b. 지금 까지 찜을 해놓았던 식당들을 한번에 모아볼 수 있다.

8. 마이 페이지 - 남긴 후기



- a. 남긴 후기를 클릭하면 식당 후기 작성을 볼 수 있다.
- b. 내가 방문한 식당과 방문한 날짜를 저장한다.
- c. 식당에 대한 만족도를 별점을 통해 쉽게 남길 수 있다.
- d. 맛과 서비스를 만족, 보통, 불만족으로 나누어 선택할 수 있다.

9. 마이 페이지 - 남긴 후기



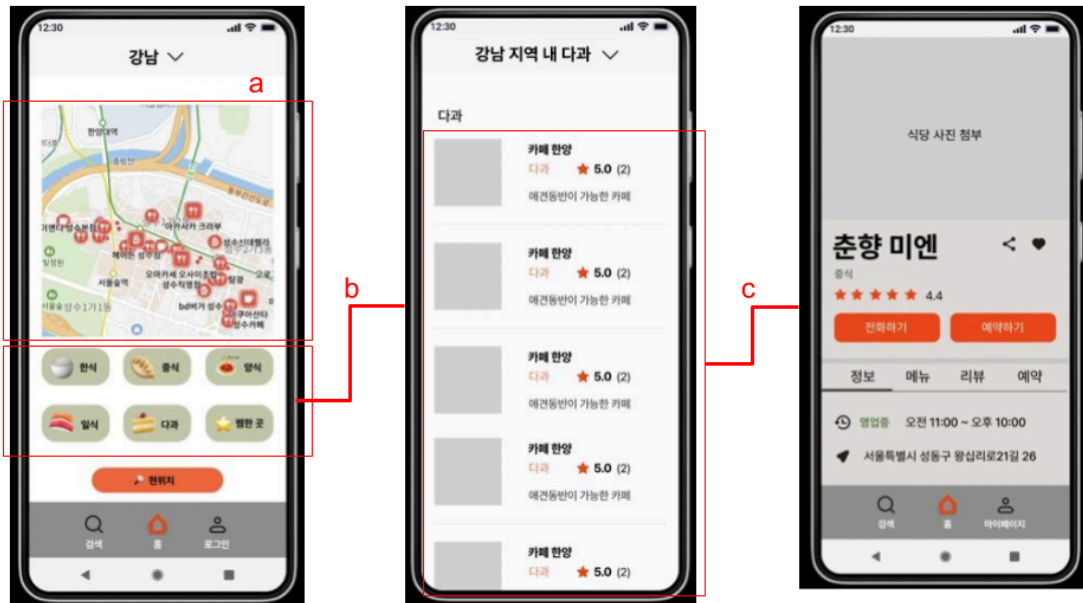
- a. 로그아웃을 클릭하면 “로그아웃 하시겠습니까?”창이 뜨면
“예”, “아니오”로 선택한다.

10. 지역 설정 화면



- a. 지역설정을 누르면 지역 설정하기 페이지가 나온다.
- b. 현위치를 누르면 현재위치에 맞는 식당들이 나온다.
- c. 자신이 가는 지역으로 선택을 한다.
- d. 지역설정하기 버튼을 누르면 선택한 지역으로 설정하는지의 창이 뜬다.

11. 지역설정 -> 식당 선택



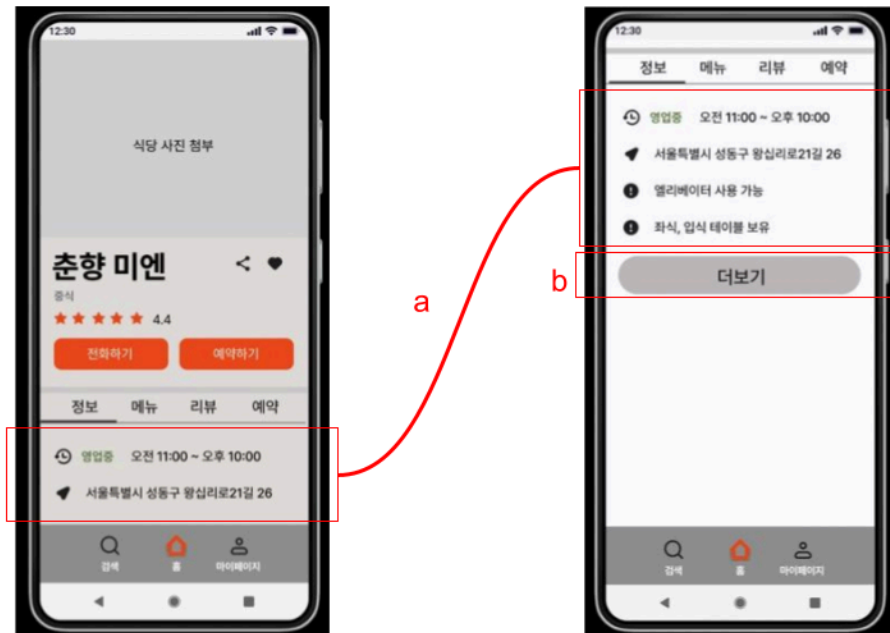
- 설정한 지역의 범위내의 식당들을 한눈에 볼 수 있도록 지도가 나타난다.
- 식당을 음식 카테고리 별로 나누어 정렬한다. 이때 한 눈에 알아보기 쉽게 이모티콘으로 아이콘 설정을 한다.
- 음식 카테고리중 하나를 고르면 그 카테고리에 관련된 식당이 나오고, 그 후에 가고싶은 식당을 선택한다.

12. 식당 페이지 (메뉴)



- a. 식당의 대표사진이나 내부사진, 음식사진등으로 설정한다.
- b. 식당이름과 음식의 종류, 그 식당의 평점을 나타낸다.
- c. 왼쪽 아이콘은 식당의 링크를 공유하고, 오른쪽 아이콘은 찜을 할 수있는 아이콘이다.
- d. “전화하기”버튼을 누르면 바로 전화연결이 된다.
- e. “예약하기”버튼을 누르면 예약창으로 넘어간다.
- f. 식당의 간단한 정보를 볼 수 있다.
- g. “메뉴”를 선택하면 간단한 메뉴를 볼 수 있다.
- h. 더 많은 메뉴를 보고싶다면 “더보기”를 누른다.

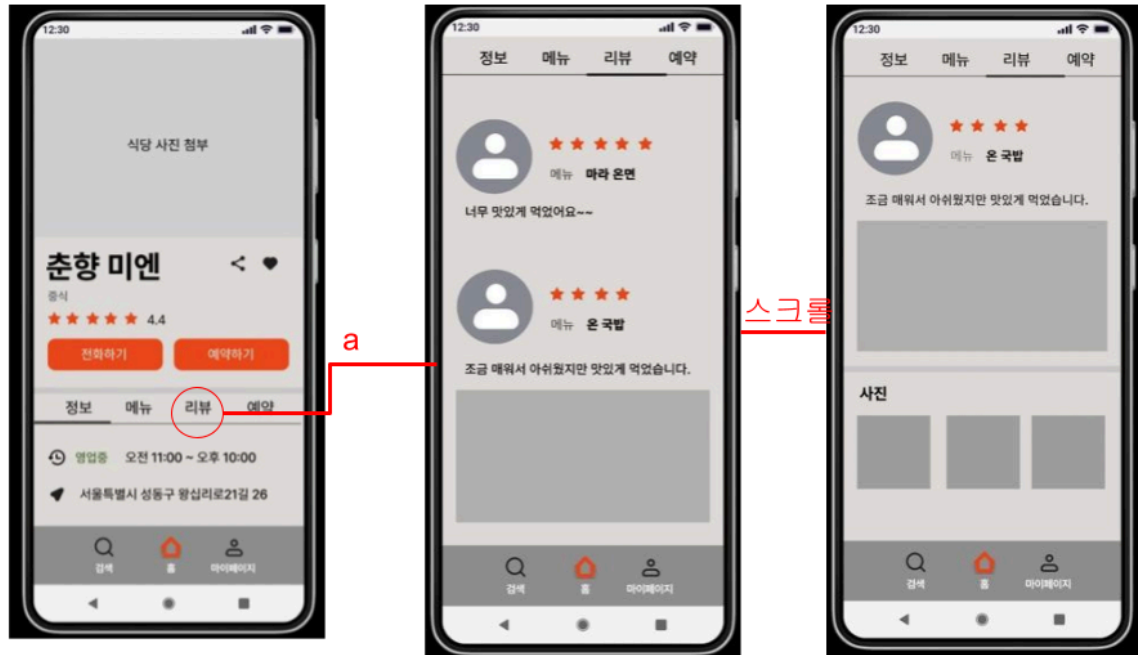
13. 식당 페이지 (정보)



a. 식당의 정보를 기입한다. 특히 엘리베이터 유무, 테이블의 형태 등.

b. 더 많은 정보를 보고싶다면 “더보기”버튼을 누른다.

14. 식당 페이지 (리뷰)



a. “리뷰” 선택하면 식당의 리뷰를 모아볼 수 있다.

스크롤 -> 스크롤하여 밑으로 내려가면 리뷰에 쓰인 사진을 모아볼 수 있다.

15. 식당 페이지 -> 예약 페이지



- “예약”을 선택하면 예약할 수 있는 페이지로 넘어간다.
- 예약 페이지에서 날짜와 인원수, 시간을 선택한다.
- “다음”버튼을 눌러 좌석을 선택할 수 있는 페이지에서 자신이 원하는 좌석을 고른다.
- “확인”버튼을 누르면 날짜, 인원수, 시간을 한 번더 강조하며 예약 의사를 묻는 팝업창이 뜬다.
“예”를 선택하면 예약이 확인됨과 동시에 문자로 예약완료 메시지가 전송된다.
“아니요”를 선택하면 다시 식당 페이지로 돌아간다.

4장 테스트 결과

1. MainActivity 코드



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent">
7
8     <!-- 스크롤 가능한 콘텐츠 영역 -->
9     <ScrollView
10         android:layout_width="match_parent"
11         android:layout_height="match_parent"
12         android:layout_above="@id/bottom_navigation"
13         android:padding="16dp">
14
15         <LinearLayout
16             android:layout_width="match_parent"
17             android:layout_height="wrap_content"
18             android:orientation="vertical">
19
20         <!-- 상단 이미지 영역 -->
```

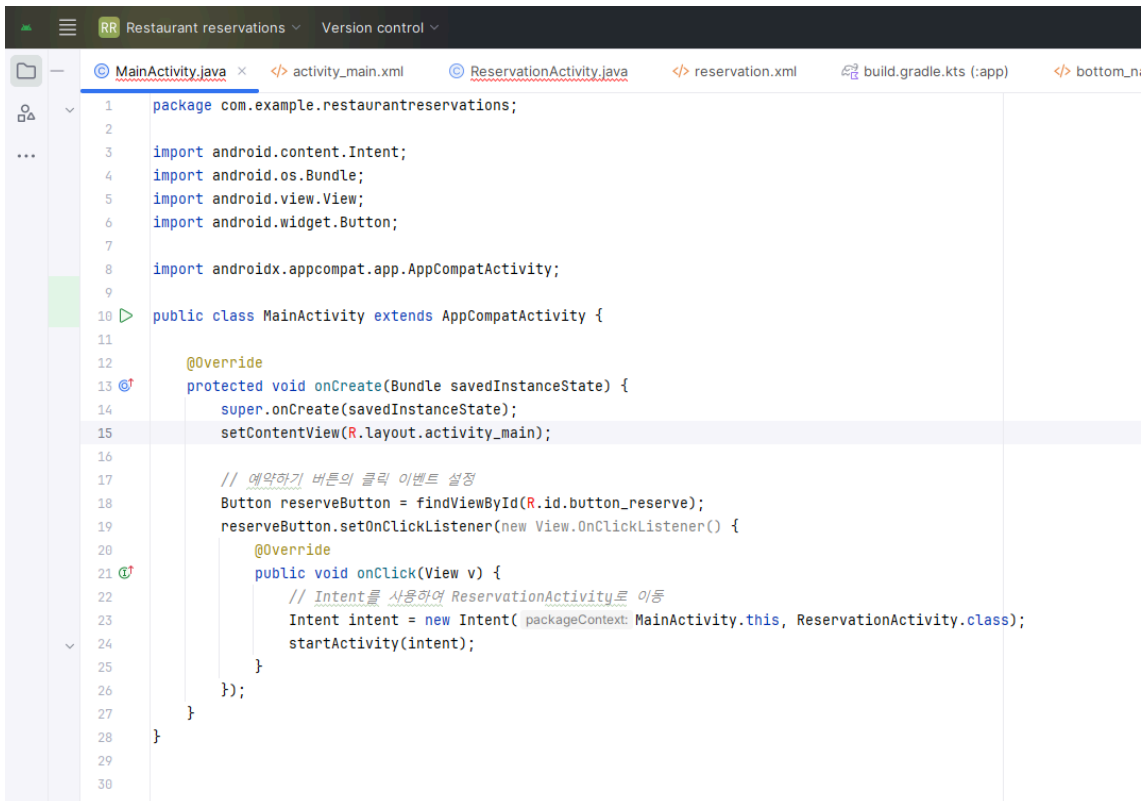
- 스크롤 뷰를 사용하여 밑으로 내리면서 볼 수 있도록 제작하였다


```

<!-- 하단 네비게이션 바 -->
<com.google.android.material.bottomnavigation.BottomNavigationView
    android:id="@+id/bottom_navigation"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    app:menu="@menu/bottom_nav_menu"
    android:background="?android:attr/windowBackground" />
</RelativeLayout>

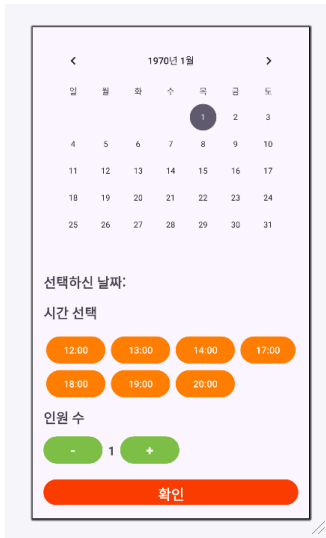
```

- 하단에는 네비게이션 바를 사용하여 검색, 홈, 마이페이지로 바로 갈 수 있도록 하였다



- 예약하기 버튼에 클릭 이벤트를 지정하여 버튼을 누르면 예약 지정 페이지로 넘어가도록 설정하였다

2.Reservation Activity 코드



```
RR Restaurant reservations Version control
© MainActivity.java </> activity_main.xml © ReservationActivity.java </> reservation.xml x bui
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:orientation="vertical"
7     android:padding="16dp">
8
9     <!-- 날짜 -->
10    <CalendarView
11        android:id="@+id/calendarView"
12        android:layout_width="match_parent"
13        android:layout_height="wrap_content"
14        android:layout_marginBottom="10dp"/>
15
```

- 캘린더 뷰를 사용하여 실제 달력이 나올 수 있도록 코드를 작성하였다

```
Restaurant reservations Version control
MainActivity.java activity_main.xml ReservationActivity.java reservation.xml build.gradle.kts (:app)

32
33 // 날짜 선택 리스너
34 calendarView.setOnDateChangeListener((view, year, month, dayOfMonth) -> {
35     String date = year + "/" + (month + 1) + "/" + dayOfMonth;
36     selectedDateText.setText("선택하신 날짜: " + date);
37 });
38
39 // 시간 버튼 설정
40 findViewById(R.id.button_time_12).setOnClickListener(v -> selectTime("12:00"));
41 findViewById(R.id.button_time_13).setOnClickListener(v -> selectTime("13:00"));
42 findViewById(R.id.button_time_14).setOnClickListener(v -> selectTime("14:00"));
43 findViewById(R.id.button_time_17).setOnClickListener(v -> selectTime("17:00"));
44 findViewById(R.id.button_time_18).setOnClickListener(v -> selectTime("18:00"));
45 findViewById(R.id.button_time_19).setOnClickListener(v -> selectTime("19:00"));
46 findViewById(R.id.button_time_20).setOnClickListener(v -> selectTime("20:00"));
47
48 // 인원 수 증감 버튼 설정
49 findViewById(R.id.button_decrease).setOnClickListener(v -> changePeopleCount(delta: -1));
50 findViewById(R.id.button_increase).setOnClickListener(v -> changePeopleCount(delta: 1));
51
52 // 예약 확인 버튼 설정
53 findViewById(R.id.button_confirm).setOnClickListener(v -> confirmReservation());
54 }
```

- 날짜 선택 리스너

calendarView.setOnDateChangeListener : 사용자가 달력에서 날짜를 선택할 때 호출된다. 그 이후 선택된 날짜를 selectedDateText에 표시한다.

```

56 // 시간 선택 함수
57 7 usages
58 private void selectTime(String time) {
59     selectedTime = time;
60     Toast.makeText( context: this, text: "선택한 시간: " + time, Toast.LENGTH_SHORT).show();
61 }
62
63 // 인원 수 변경 함수
64 2 usages
65 private void changePeopleCount(int delta) {
66     if ((peopleCount + delta) > 0) {
67         peopleCount += delta;
68         peopleCountText.setText(String.valueOf(peopleCount));
69     }
70 }
71
72 // 예약 확인 함수
73 1 usage
74 private void confirmReservation() {
75     String date = selectedDateText.getText().toString();
76     if (date.isEmpty() || selectedTime.isEmpty()) {
77         Toast.makeText( context: this, text: "날짜와 시간을 모두 선택해 주세요.", Toast.LENGTH_SHORT).show();
78     } else {
79         String reservationInfo = "예약 정보\n날짜: " + date + "\n시간: " + selectedTime + "\n인원 수: " + peopleCount;
80         Toast.makeText( context: this, reservationInfo, Toast.LENGTH_LONG).show();
81     }
82 }

```

- 시간 선택 기능

findViewById(R.id.button_time_12).setOnClickListener
(v -> selectTime("12:00")); : 버튼에 클릭 리스너를 설정해,
버튼 클릭 시 메서드가 호출되고, 선택된 시간을 selectTime
에 저장하고 Toast 메시지로 표시한다.

- 인원수 조절 기능과 예약 확인 기능도 마찬가지로 동일한 코드를 사용하여 제작하였다.

5장 결과 분석

5.1 결과 분석 및 개선점

기술적인 한계와 리소스 부족 등으로 코딩이 완료 되지 못하였다. 식당페이지에서 예약페이지로 넘어가는 시점에서는 R로 인한 디버깅이 발견되었는데 이를 해결하지 못하였다. 향후에는 미완성된 기능과 코드를 보완하기 위해 노력할 것이며 모든 기능이 구현된 후, 사용자 테스트를 통해 성능 평가를 수행할 예정이다.

5.2 참고문헌

IT CookBook, Android Studio를 활용한 안드로이드 프로그래밍(9판)
한빛 아카데미