

Kotlin 안드로이드 프로그래밍 최종 보고서

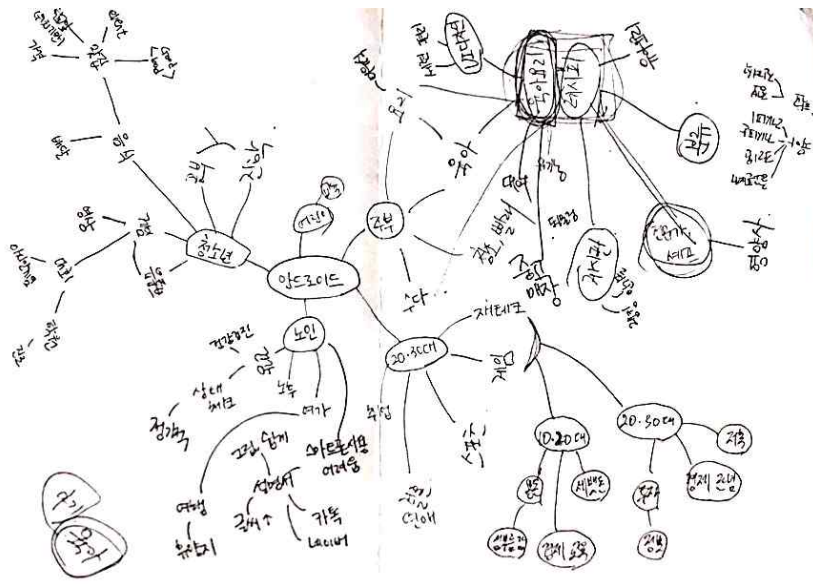
1. 프로젝트 요약문(연구목표 및 최종결과물)

구 분	내용
과제 개요	<p>스마트폰 시장의 규모가 커지면서 스마트폰 플랫폼을 사용하는 다양한 서비스 또한 생겨나고 있다. 현재 시장에는 육아를 막 시작한 초보 엄마, 아빠들을 위한 앱이 많이 없다고 느껴 오로지 육아 정보만을 담고 있는 간편하고 이용하기 쉬운 앱 제작을 기획하였다.</p> <p>육아에 필요한 중고 물품들을 사고 팔 수 있는 마켓, 시기별 이유식의 요리 정보를 공유하는 레시피, 그리고 각종 육아 정보와 팁을 서로 공유할 수 있는 톡톡을 게시판 형태로 구현한다.</p> <p>안드로이드 스튜디오 프로그램을 사용하고 Kotlin 언어 기반으로 한 앱을 제작하고자 한다.</p>
최종 목표	<p>사용자는 회원가입과 로그인을 한 후에 자신의 게시글을 업로드하고 다른 사용자들이 업로드한 게시글을 자유롭게 이용할 수 있다.</p> <ul style="list-style-type: none"> - Handler를 사용하여 delay를 줘서 앱을 처음 시작할 때 로딩화면이 잠깐 나타나게 구현 - 메인화면에 로그인, 회원가입, 마이페이지, 톡톡, 레시피, 마켓 버튼 구현

- 톡톡, 레시피, 마켓 버튼을 누르면 팝업창을 띄워 로그인과 회원가입을 해야 앱을 이용할 수 있도록 함
- 회원가입을 하면 데이터베이스에 아이디와 비밀번호와 추가정보로 생년월일, 성별, 전화번호가 저장되도록 하고 로그인을 할 수 있게 함
- 로그인을 하면 메인페이지 아래에 아이디가 표시되도록 함
- 게시 글을 작성하여 등록하면 Gridview (or Listview) 를 통해 화면에 보임
- 작성한 게시 글을 항목 별로 데이터베이스에 저장되도록 함
- 스피너와 라디오버튼으로 카테고리를 지정하여 카테고리별로 구분할 수 있게 함
- Layout Inflater를 사용하여 카테고리별로 레이아웃만 바꿔주도록 구현
- drawer layout으로 메뉴바 구현
- 마이페이지에서 회원 정보 수정 가능하게 구현

2. 과제 세부 설명

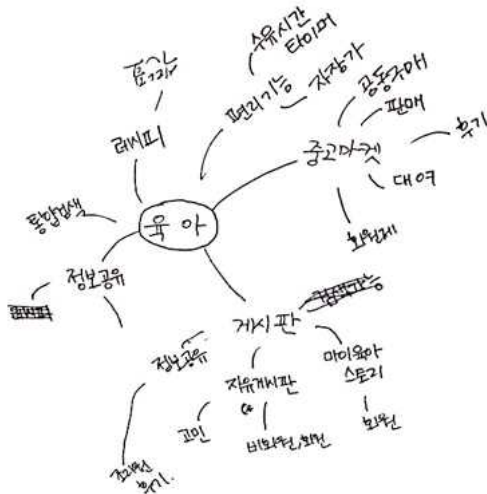
이 프로젝트의 가장 첫 시작은 마인드맵을 통하여 우리 팀이 만들 어플리케이션의 주제를 정하는 것이었다. 다음은 우리 팀의 첫 번째 마인드맵이다.



우리 팀은 '안드로이드' 라는 키워드를 중간에 잡고, '어떤 대상을 위한 어플을 만들 것인가?' 라는 내용으로 다음 가지를 분류하였다. 청소년, 노인, 어린이, 주부 그리고 20:30대로

다음 가치를 뽑고, 각 대상들을 위하여 어떤 기능을 가진 어플이 유용할 것인지 중점을 두고 나머지 가치를 뽑어나갔다.

이 후에 각각 하나의 대상을 선정하여 각자 마인드맵을 그려왔고, 팀원들과 토의를 한 결과 우리 팀의 주제는 '주부를 위한 육아어플리케이션'으로 선정되었다.



그렇다면 우리의 육아 어플리케이션에는 어떤 기능이 들어가면 좋을까?

세 번째 시간에는 위 질문에 대해 토의하였고, 마인드맵을 통하여 육아 어플의 세부기능을 정했다. 그 결과 레시피, 중고마켓, 게시판, 마이육아스토리 등을 구현하기로 했다.

그 후에는 레시피, 중고마켓, 게시판, 마이육아스토리 중 각자 세부적으로 어떤 기능을 구현하면 좋을지에 대한 내용을 핀 카드로 작성했다. 다음은 핀카드 중 마켓에 관련된 부분이다.

핀카드2

2. 중고마켓에서 원하는 것을 내놓은 구매자와 1:1 매칭

- A라는 엄마가 만약 아이가 스드래 1 이상 필요 없었기 같아 중고마켓에 올리려고 할 때, 자신이 물리고픈 물건이 만약 B라는 또다른 엄마가 구매를 원하는 다른 게시판을 올렸을 때, 둘은 올릴 때 자동으로 그 게시판과 연결을 꼭, 매칭을 해주는 기능 추가.

구매자가 원하는 물품을 빠르게 찾아 줄수있어 좋은 기능인것 같아요 5

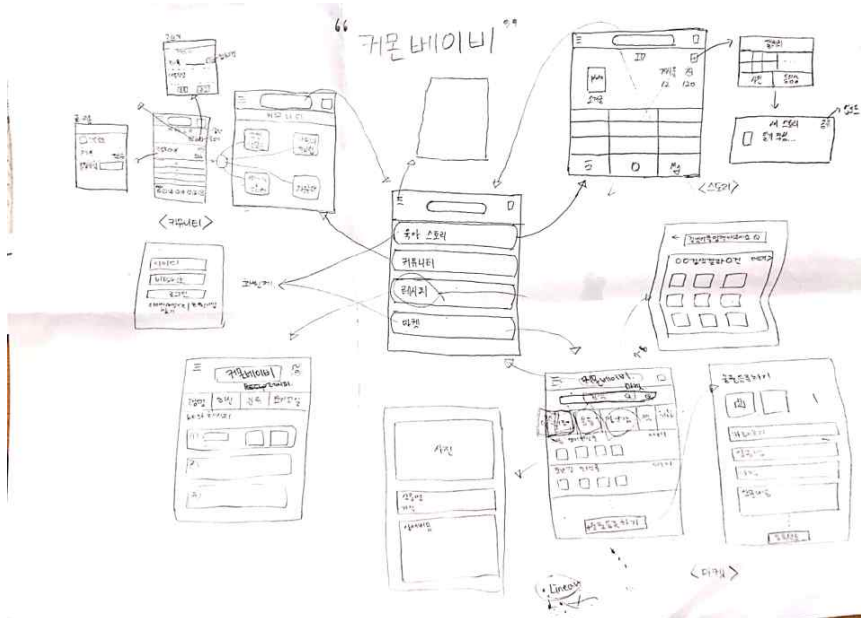
좋은 기능이지만 현실적으로 조금 어려울수도 있을것 같고

판매자나 구매자가 물건 구매 보고 직접 연락하는것이

더 나은 방법을 찾을수 있을것 같다!

이와 같이 세부기능에 대한 내용을 뒷장에 서로 피드백 해주며 어플리케이션의 기능들을 더욱 구체화시켰다.

마지막으로 각자 레시피, 중고마켓, 게시판, 마이육아스토리에 대한 어플리케이션 화면을 그려온 후에 다음과 같이 통합하는 과정을 거쳤다.



이 후에 게시판에 있는 수다방, 육아tip 등을 통하여 마이육아스토리 기능을 대체할 수 있다고 판단하여 육아스토리를 제외한 레시피, 마켓, 게시판(톡톡)으로 “커몬베이비”라는 우리의 어플을 구현해 나가기 시작하였다.

세부적인 어플리케이션 구현 계획을 살펴보면,

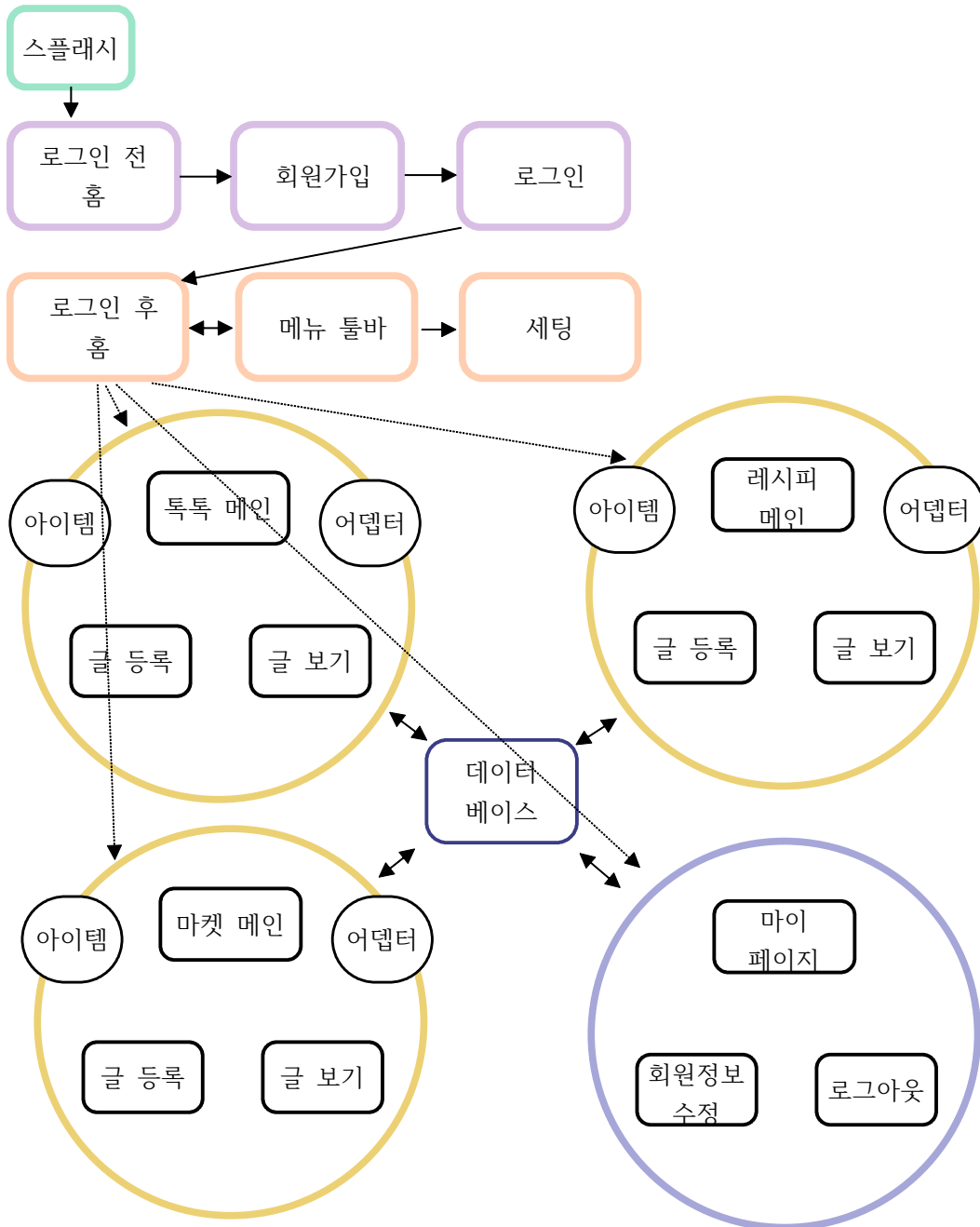
- 1) 디자인은 파란색으로 포인트를 주어 깔끔하고 모던한 느낌으로 계획하였다. 또한 우리만의 로고를 만들어 어플 메인페이지와 상단 부분에 표시하기로 한다.
- 2) 회원제로 운영하여 회원가입 및 로그인을 한 후 어플을 이용할 수 있도록 한다. 이때, 로그인 했을 때와 안했을 때를 구분하여 페이지를 만든다.(후에 다이얼로그를 활용하여 로그인을 하지 않았을 경우 접근이 불가능 하도록 함)
- 3) 게시판을 구현한다. 톡톡에는 제목과 내용을, 마켓과 레시피에는 제목과 내용 그리고 사진까지 등록을 할 수 있도록 한다. 이 때, 사진은 갤러리 접근을 통하여 불러올 수 있도록 한다.
- 4) 글을 등록하면 카테고리 별로 글이 차례대로 등록되도록 한다.(후에 리스트뷰와 그리드뷰를 활용함)
- 5) 메뉴바를 넣어 클릭시 왼쪽에서 슬라이딩 되는 메뉴바가 나오도록 한다.
- 6) 스플래시 화면을 넣어 처음에 약 3초 후 메인 페이지가 나오도록 한다.

※ 2),3),4) 는 데이터베이스를 활용하여 구현하도록 한다. (레시피, 마켓, 톡톡 각각 테이블을 만듦)

이와 같은 계획을 가지고 본격적으로 안드로이드 스튜디오를 활용하여 “커몬베이비” 어플을 만들기 시작했다.

3. 과제 기능별 설명

1) 전체 클래스 다이어그램



2) 세부 클래스 분류 및 진행 상황


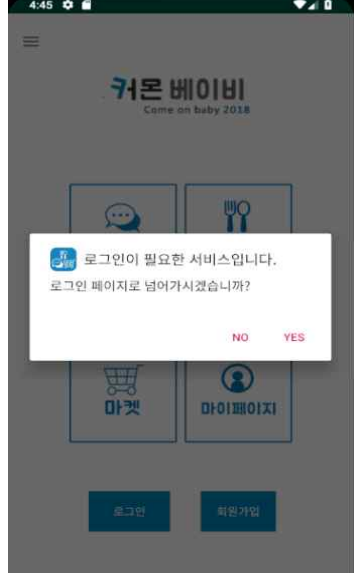
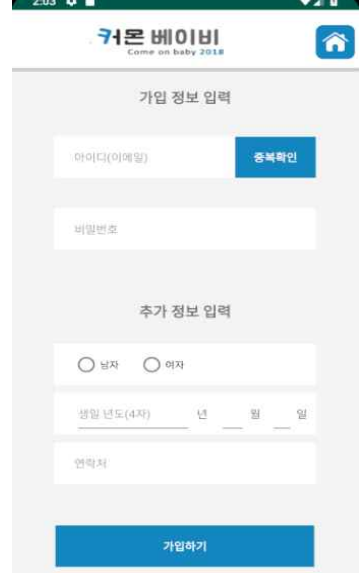

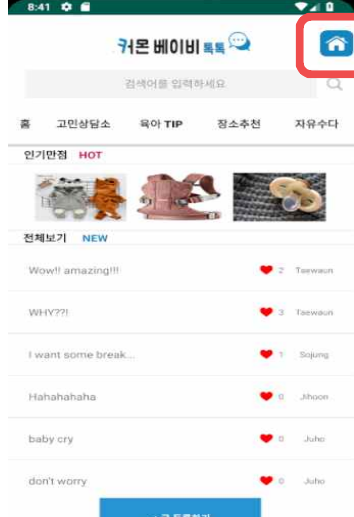
세부내용	구 분	완성도										사유	진척도 (%)
		1	2	3	4	5	6	7	8	9	10		
스플래시 화면 기능	계획											.	100
	진도												
아이디 조회 기능	계획											.	100
	진도												
회원가입 기능	계획											.	100
	진도												
로그인&자동 로그인 기능	계획											.	100
	진도												
회원 정보 수정 기능	계획											.	100
	진도												
메뉴 툴바 기능	계획											.	100
	진도												
리스트뷰 & 그리드뷰 기능	계획											.	100
	진도												
인플레이터 기능	계획											.	100
	진도												
게시글 등록 기능(사진,스피너,버튼)	계획											.	100
	진도												
게시글 보기 기능(하트클릭, 사진)	계획											.	98
	진도												

4. 프로그램 구성

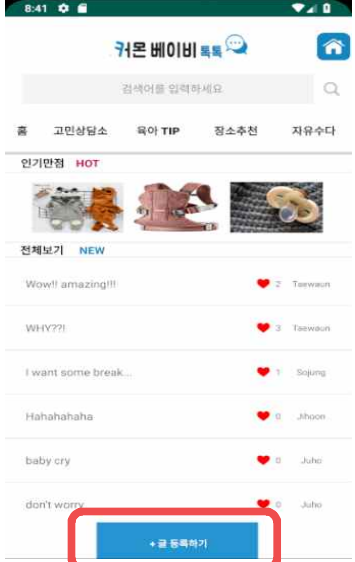

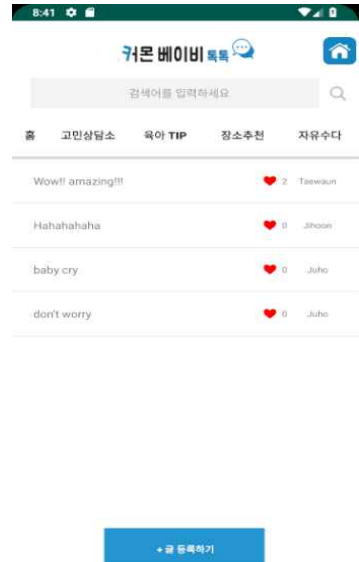

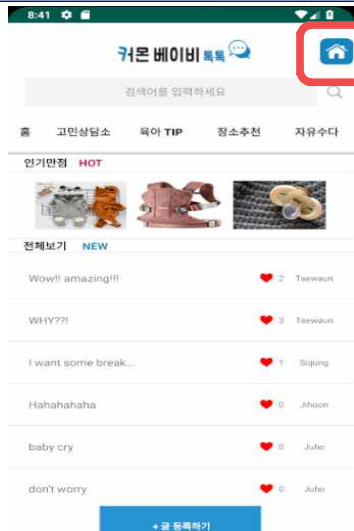
1) 메뉴 구성도

가. 시작화면	나. 로그인 화면	다. 톡톡 게시판
		
<p>3개의 종류의 게시판과 마이 페이지, 로그인, 회원가입 메뉴로 구성</p>	<p>3개의 게시판으로 들어가기 위한 로그인 실행, 회원가입이 필요하면 회원가입 가능</p>	<p>고민 상담, 육아 tip, 장소 추천, 자유 수다 4개로 구성된 게시판으로 다른 사용자들과 정보를 공유</p>
라. 레시피 게시판	마. 마켓 게시판	바. 마이 페이지
		
<p>초기 이유식, 중기 이유식, 후기 이유식, 유아 요리 4가지로 구성된 게시판으로 사용자들이 본인만의 레시피를 등록 가능</p>	<p>의류, 용품, 장난감, 서적, 기타 5가지로 구성된 게시판으로 상품을 등록하여 사용자와 물품을 구매하고 판매 가능</p>	<p>로그인된 사용자의 회원 정보를 볼 수 있으며 정보를 변경하고 로그아웃 가능</p>
<p>총 3개의 게시판이 존재하며 로그인 없이는 게시판을 이용할 수 없음. 게시판은 각각의 카테고리를 구성하고 있으며 다른 카테고리나 다른 게시판으로 쉽게 넘어갈 수 있음. 게시판에서 타인의 글을 볼 수 있고 자신의 게시글을 등록할 수 있도록 함.</p>		


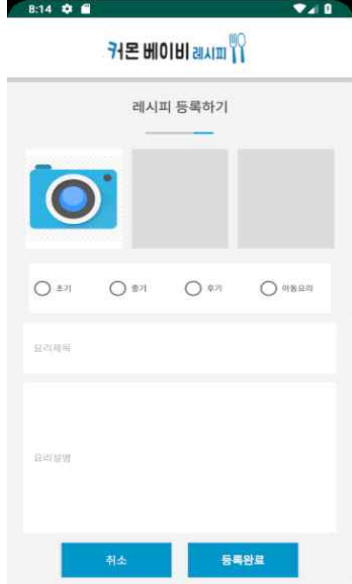

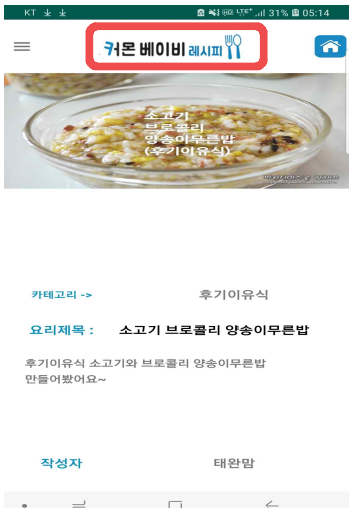
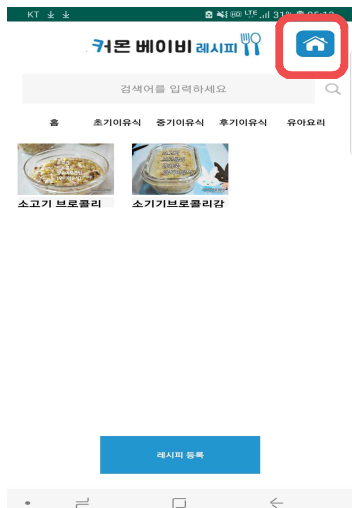
2-1) 사용 방법(회원가입&로그인)

1. 로그인 안 된 홈 화면	2. 비 로그인	3. 회원가입
		
<p>로그인이 되지 않은 상태의 홈 화면</p>	<p>로그인이 되지 않은 상태에서 게시판을 눌렀을 때 다이얼로그를 띄어 로그인 필요함을 알리고 로그인 화면으로 넘어감</p>	<p>아이디, 패스워드, 부가 정보 등을 입력한 후 회원가입 가능. 아이디 중복확인 가능</p>
4. 로그인	5. 로그인 후 홈 화면	
		
<p>회원가입 후 로그인을 할 수 있으며 회원 정보가 등록이 안 되어 있으면 로그인 실패라고 뜨며 회원 정보가 있다면 홈 화면으로 넘어감</p>	<p>로그인 후 홈 화면으로 넘어오면 로그인 완료된 ID가 화면에 보이고 게시판에 들어갈 수 있음</p>	

2-2) 사용 방법(톡톡 게시판)

1. 톡톡 게시판 메인	2. 게시글 등록	3. 게시글 화면에 표시
		
<p>홈 화면에서 톡톡 게시판을 눌렀을 때 나오는 톡톡 게시판 메인화면</p>	<p>글 등록하기를 눌렀을 때의 화면. 제목, 카테고리, 내용을 등록할 수 있음</p>	<p>게시글을 등록하면 해당 카테고리에 맞게 제목과 하트 수, ID가 게시글로 표시</p>
4. 게시글 내용 보기	5. 홈 화면으로 가기	
		
<p>화면에 표시된 게시글을 클릭하면 게시글 내용을 볼 수 있음. 게시글이 마음에 들면 하트를 클릭. 상단의 톡톡 로고를 클릭하면 톡톡 메인 화면으로 돌아감</p>	<p>상단에 홈 그림의 버튼을 클릭하면 홈 화면으로 갈 수 있음</p>	





2-3) 사용 방법(레시피 게시판)

1. 레시피 게시판 메인	2. 게시물 등록	3. 게시물 화면에 표시
		
<p>홈 화면에서 레시피 게시판을 눌렀을 때 나오는 레시피 게시판 메인화면</p>	<p>글 등록하기를 눌렀을 때의 화면. 요리 이름, 카테고리, 요리설명, 사진을 등록할 수 있음</p>	<p>게시글을 등록하면 해당 카테고리에 맞게 사진과 제목, ID가 게시물로 표시</p>
4. 게시물 내용 보기	5. 홈 화면으로 가기	
		
<p>화면에 표시된 게시글을 클릭하면 게시물 내용을 볼 수 있음. 상단의 레시피 로고를 클릭하면 레시피 메인화면으로 돌아감</p>	<p>상단에 홈 그림의 버튼을 클릭하면 홈 화면으로 갈 수 있음</p>	



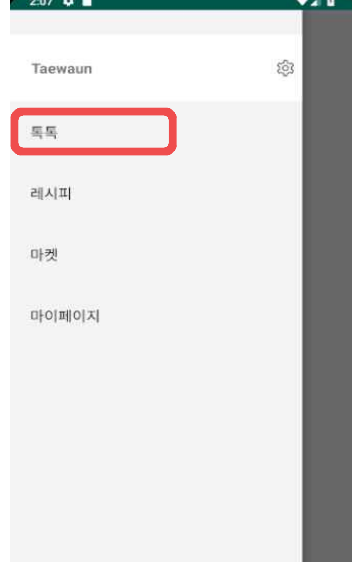

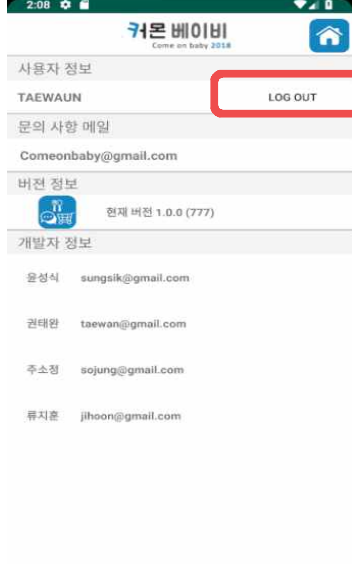
2-4) 사용 방법(마켓 게시판)

1. 마켓 게시판 메인	2. 게시물 등록	3. 게시물 화면에 표시
		
<p>홈 화면에서 마켓 게시판을 눌렀을 때 나오는 마켓 게시판 메인화면</p>	<p>글 등록하기를 눌렀을 때의 화면. 상품 이름, 카테고리, 가격, 사진, 내용을 등록할 수 있음</p>	<p>게시글을 등록하면 해당 카테고리에 맞게 사진과 제목, ID가 게시물로 표시</p>
4. 게시물 내용 보기	5. 홈 화면으로 가기	
		
<p>화면에 표시된 게시글을 클릭하면 게시물 내용을 볼 수 있음. 상단의 마켓 로고를 클릭하면 마켓 메인화면으로 돌아감</p>	<p>상단에 홈 그림의 버튼을 클릭하면 홈 화면으로 갈 수 있음</p>	

2-5) 사용 방법(마이 페이지)

1. 마이 페이지 화면	2. 회원 정보 수정	3. 비밀번호 변경
		
홈 화면에서 마이 페이지를 눌렀을 때 나오는 화면으로 회원 정보수정, 비밀번호 변경, 로그아웃을 할 수 있음	회원 정보 수정을 눌렀을 때 나오는 화면으로 정보를 확인하고 연락처를 변경할 수 있음	비밀번호 변경을 눌렀을 때 나오는 화면으로 사용자의 비밀번호를 변경할 수 있음
4. 로그아웃		
		
현재 로그인 되어 있는 사용자가 로그아웃 할 수 있음. 로그아웃이 되면 로그아웃된 홈 화면으로 넘어감		

2-6) 사용 방법(메뉴 톨바)

1. 홈 화면	2. 메뉴 톨바	3. 톨바에서 클릭 시
		
<p>홈 화면에서 상단에 메뉴 톨바 표시</p>	<p>홈 화면에서 메뉴 톨바를 눌렀을 때 나오는 화면 각 게시판들과 마이 페이지로 구성</p>	<p>메뉴 톨바에서 클릭 시 해당 게시판으로 넘어감</p>
4. 설정 버튼 클릭	5. 로그아웃	
		
<p>메뉴 톨바에서 설정 버튼 클릭하면 사용자와 앱 정보를 확인 가능</p>	<p>로그아웃을 클릭하면 로그아웃된 홈 화면으로 넘어감</p>	

5. 필수 적용 사항

1. DB 적용

우리 앱인 Come On Baby는 육아를 테마로 한 정보 공유 앱이다. 따라서 세 가지의 각 세부 항목인

톡톡, 레시피, 마켓이 존재한다. 이러한 각 항목에서 글을 쓰거나 원하는 내용을 올리기 위해서 회원들은 먼저 자신만의 회원 정보를 갖고 있어야 한다. 이러한 이유로 회원 가입 클래스를 우선 필수적으로 구현하였다. 이 부분에서 우리 조의 핵심 적용 사항인 데이터 베이스(이하 DB)를 구현하였다. 우선적으로 수업 시간에 실습한 DB를 기본으로 구현하였다.

*DB Helper

DB Helper라는 DB관련 코틀린 파일을 생성하여 각 내가 만들고 싶은 테이블을 생성한다. 테이블에는 예를 들어 사용자 정보에 관한 테이블을 만들었다면 내가 저장하고 싶은 사용자 정보들의 이름을 선언한다. 그런 다음 DB를 실행시켜 최종적으로 DB를 사용할 수 있게 한다.

이 앱에서 사용하는 DB테이블은 총 3가지이며, 사용자 정보, 톡톡, 레시피, 마켓 테이블이다.

*DB에 있는 정보를 찾기 위한 Cursor

안드로이드에서는 DB에 저장되어있는 데이터를 처리하기 위해서 Cursor(이하 커서)라는 인터페이스를 제공해준다. 이 인터페이스는 기본적으로 DB에서 값을 가져와서 마치 실제 테이블의 한 행, 한 행을 참조하는 것처럼 사용 할 수 있게 해준다. 실제 커서가 행을 참조하기 때문에 커서의 위치를 바꿔주는 방법들을 이용하여 원하는 데이터가 있는 행으로 이동할 수 있게 한다. 이러한 방법 이용하여 최종적으로 DB에 생성된 테이블의 해당 필드의 데이터를 얻어 올 수 있다.

*사용자 정보 DB 테이블

사용자 정보는 크게 아이디, 비밀번호, 성별, 생년월일, 전화번호가 있다 아래부터는 사용자 DB정보를 가지고 수행하는 모든 부분의 설명이다.

*톡톡 정보 DB 테이블

톡톡 정보는 크게 제목, 카테고리, 글의 내용, 사용자 아이디, 좋아요(하트 모양)가 있다 아래부터는 사용자 DB정보를 가지고 수행하는 모든 부분의 설명이다.

***레시피 정보 DB 테이블**

레시피 정보는 크게 제목, 카테고리, 글의 내용, 사진, 사용자 아이디가 있다 아래부터는 사용자 DB정보를 가지고 수행하는 모든 부분의 설명이다.

***마켓 정보 DB 테이블**

마켓 정보는 크게 제품 이름, 가격, 사진, 카테고리, 글의 내용, 사용자 아이디가 있다 아래부터는 사용자 DB정보를 가지고 수행하는 모든 부분의 설명이다.

- DB관련 데이터를 저장하고 찾는 과정이 적용된 부분 설명

(1) 회원 가입

회원 가입	사용자의 정보를 생성해 저장하는 역할
왜 필요한가?	앱을 사용하기 위해서는 자신만의 사용자 정보가 필요하다. 이 정보를 바탕으로 각 세부적인 앱 항목들인 톡톡, 레시피, 마켓, 마이페이지에 접근 권한이 주어진다. 결국 앱을 사용하기 위한 아주 기초적인 단계이므로 핵심적인 필수 사항이다.
적용 방법	DB helper를 통해 사용자 정보 테이블을 만들어 놓았다면 그것을 이용하여 원하는 정보를 쓸 수 있게 만든다. 그런 다음 아이디, 비밀번호, 성별, 생년월일, 전화번호와 같이 필수적으로 담아야 할 정보들을 스트링 값으로 가져와서 변수에 저장한 다음, 저장한 변수들을 하나의 배열로 저장한다. 그런 다음 사용자 정보 테이블과 내가 넣고 싶은 사용자 정보들의 배열 이 두 가지를 가지고 DB를 실행시켜 저장해준다. 이제 모든 정보를 저장하였다면 최종적으로 DB를 닫아줌으로써 회원가입을 완료한다.
회원 가입 관련 if문	회원 가입 시, 같은 아이디가 현재 DB에 존재하는지 확인 합니다. 이것을 통해 같은 아이디 정보를 가진 사용자 생성을 막아줍니다. 이 부분을 구현하기 위해서 DB에 저장되어있는 아이디 정보와 현재 입력한 아이디를 비교하여 중복 체크를 해줍니다. DB helper를 이용하여 사용자 정보 테이블을 연 다음, 커서를 이용하여 찾고자 하는 정보의 행으로 이동합니다. 그 후, 행에 저장되어있는 아이디를 가져와서 비교합니다. 만약 비교했을 때, 같다면 다른 아이디를 입력해야 된다는 정보를 알려주고 같지 않다면 회원 가입이 가능하도록 코드를 구성하였습니다.

(2) 로그인

로그인	생성한 사용자 정보를 가지고 앱을 사용할 수 있는 단계로 넘어 가게 하는 역할
적용 방법	한 사용자의 정보가 사용자 정보 테이블을 저장되었다면 그것을 이용하여 사용자가 앱을 사용할 수 있는 단계로 넘어가게끔 해줘야한다. 이 부분이 바로 로그인이다 커서를 이용하여 사용자 정보 테이블을 검색한다. 만약 현재 로그인 하고 싶은 유저가 자신의 아이디와 비밀번호를 입력한다면 이 입력한 정보를 가지고 사용자 테이블에 저장되어있는 정보와 비교한다. 사용자 정보 테이블에 존재하는 아이디와 그와 같은 행에 만들어진 비밀번호 데이터가 현재 사용자의 입력 정보와 같다면 로그인을 하여 앱의 다른 부분을 사용할 수 있게 만들어 주고 같지 않다면 로그인을 하지 못 한다.
이전 로그인 정보 저장	DB의 테이블에서 원하는 데이터를 가져오는 방식도 있지만 로그인 정보 저장이라는 것은 모든 사용자 정보가 필요하지 않고, 최신의 정보 하나만 저장하고 있으면 되기 때문에 값 저장의 간단한 방식인 SharedPreferences를 사용하였다. 이 방법은 데이터가 파일 형태로 앱에 저장이 되는 방식이고 또한 앱을 삭제하기 전까지 값을 저장하고 있기 때문에 하나의 작은 DB라고 봐도 무방하다. 현재 로그인 된 정보라는 의미를 가진 SharedPreferences를 생성하여 로그인 시 입력한 아이디와 비밀번호(현재 로그인된 정보)를 저장한다. 이 방법을 이용하여 로그아웃 하고 나서 다시 로그인을 할 경우 먼저 getSharedPreferences라는 구문을 이용하여 저장 되어 있던 제일 최신 로그인 정보를 먼저 띄워 주어서 사용자가 로그인 창을 켜올 경우, 다시 자신의 아이디와 비밀번호를 입력하지 않아도 로그인을 할 수 있게 하였다.

(3) 톡톡의 글 등록

<p>톡톡의 글 등록 DB적용 방법</p>	<p>우선적으로 글 등록 시, 톡톡 테이블 생성한 필드들의 각 항목들을 사용자의 입력을 받아 넣고 DB 필드 중 좋아요는 우선 0으로 초기화 시켜, 위의 DB 저장 과 같은 방법으로 톡톡 테이블에 데이터를 저장한다.</p> <p>이 때, 인텐트를 통해서 넘겨 받은 현재 로그인 된 아이디 값을 톡톡의 테이블이 넣는다. 예를들어 인덱스 0번에 a라는 사용자의 톡톡 정보가 담기고, 로그 아웃한 뒤, b라는 사용자가 로그인하여 톡톡에 글을 등록할 시 b라는 아이디 정보를 가진 같은 행 인덱스 1에 저장 이 되는데 각 사용자의 정보를 이용하여 톡톡 테이블의 정보를 접근 하므로 각각의 사용자에게 관해서 테이블이 만들어졌다고 볼 수 있다.</p>
<p>톡톡의 등록 된 글 화면 DB적용 방법</p>	<p>글 등록이 되었다면 DB에 한 사용자가 등록한 톡톡 테이블의 정보가 저장될 것이다. 그것을 이용하여 사용자가 등록된 글을 볼 수 있어야 한다. 커서를 이용하여 톡톡 테이블을 검색하여, 지금 사용자가 클릭한 글의 내용을 찾아 보여준다.</p> <p>이 때, 톡톡에서 다른 사용자가 좋아요 하트 버튼을 누른다면 toggleButton.setOnClickListener을 통해 만약 하트 모양의 버튼이 체크가 되었다면 DB에 저장된 좋아요 +1 하여 필드의 값을 수정한다. 또한 체크가 해지 된다면 DB에 저장된 좋아요 -1 하여 필드의 값을 수정한다.</p>

(4) 레시피, 마켓의 글 등록

<p>레시피, 마켓의 글 등록 DB적용 방법</p>	<p>우선적으로 글 등록 시, 레시피, 마켓의 테이블 생성한 필드들의 각 항목들을 사용자의 입력을 받아 넣고 위의 DB 저장 과 같은 방법으로 레시피, 마켓의 테이블에 데이터를 저장한다.</p>
----------------------------------	--

<p>레시피, 마켓의 등록 된 글 화면 보기 DB적용 방법</p>	<p>글 등록이 되었다면 DB에 한 사용자가 등록한 레시피, 마켓의 테이블의 정보가 저장될 것이다. 그것을 이용하여 사용자가 등록된 글을 볼 수 있어야 한다. 커서를 이용하여 레시피, 마켓의 테이블을 검색하여, 지금 사용자가 클릭한 글의 내용(마켓: 제품명, 제품 가격, 제품 상세 내용, 전화번호)(레시피: 요리명, 요리 재료, 요리 상세 과정)을 찾아 보여준다. 이 때, 사진을 가져올 수 있어야 하므로 아래와 같은 방법을 통해 구현했다.</p> <p>** 사진 가져오는 방법은 위 밑에 그 밖의 기능 구현에서 설명.</p> <p>-사진의 URL 정보 저장 후 찾는 과정</p> <p>만약 글을 등록할 때는 원하는 사진의 URL을 받아서 DB의 사진 필드에 URL을 저장한다.</p> <p>그런 다음, 이제 글 등록 된 것을 클릭하여 보려고 할 때는 setImageURI이라는 메서드를 이용하여서 지금 띄워야하는 글의 내용과 같은 열의 필드에 있는 사진 URL를 받아와서 변수에 저장한 후, DB를 실행 시켜 글의 내용 뿐만 아니라 사진 파일 또한 띄울 수 있게 하였다.</p>
<p>레시피, 마켓의 글 등록 DB와 그리드 뷰 연동 관련 설명</p>	<p>우선 글 등록을 완료하면 마켓과 레시피 홈과 각 카테고리에 맞는 곳에 글을 띄워준다. 이 때, 각 카테고리의 포지션이 예를 들어 옷이라는 카테고리에 생성된 포지션이 0번부터 시작되어 그리드 뷰에 보여진다면 홈에서는 똑같이 추가가 되는데, 이 경우에 홈의 포지션이 각 카테고리에서 생성된 글의 포지션과 매치가 안 될 경우가 발생한다. 이런 경우를 바로 잡기 위해서 각 카테고리 마다의 포지션을 숫자를 각기 다른 변수로 설정하여 1부터 시작하게끔 초기화 해준다. 그리고 글을 등록 할 시, 어떤 카테고리에 넣을지, 카테고리가 선택되면 어느 번째 포지션을 갖도록 할지를 설정해준다. 이 과정을 거치고 나서 카테고리 명, 카테고리의 포지션, 카테고리의 데이터가 모두 내가 보여주고 싶은 것과 일치하면 DB에 저장된 맞는 데이터들을 가져와서 보여줍니다.</p>

- DB관련 테이블에 저장된 데이터 수정하는 과정의 부분 설명

(5) 마이페이지

회원 정보 수정 및 비밀 번호 수정	사용자가 정보를 수정할 수 있게 도와주는 역할
회원 정보 수정 적용 방법	<p>사용자가 저장한 자신의 정보를 수정하기 위해서 우선 DB의 사용자 테이블을 열어준다. 그 후 우선 자신이 저장한 정보들을 확인 할 수 있게 자신의 아이디 값을 이용하여 사용자 정보 테이블 내의 자신의 정보를 가진 행을 찾아 받아와 수정하지 않는 항목의 데이터들을 화면에 띄워준다. 이후 회원 정보 수정에서 수정하려는 핸드폰 번호 정보를 우선 똑같이 다른 정보들처럼 볼 수 있게 띄워 준 후, DB 수정 방법인 update를 사용하여 사용자 정보 테이블내의 사용자 아이디 행을 찾은 다음, 그 행에 전화 번호 필드를 다시 설정한다. 테이블에 값을 삽입할 때의 방법과 같이 수정한 데이터를 배열로 저장하여 DB를 실행시켜 내가 현재 로그인한 아이디와 같은 행에 속한 항목 데이터를 수정 해준다.</p>
비밀번호 수정 적용 방법	<p>자기가 새로 수정하고 싶은 비밀번호를 입력 받는다. 그 다음, 위의 회원 정보 수정 방법과 같이 사용자가 저장한 자신의 정보를 수정하기 위해서 우선 DB의 사용자 테이블을 열어준다. 사용자 테이블의 정보를 수정하기 싶은 경우이므로 DB 수정 방법인 update를 사용하여 사용자 정보 테이블내의 사용자 아이디 행을 찾은 다음에 그 행에 속한 비밀번호 필드의 데이터를 새로 설정한다는 것을 알려준 후, 새로운 비밀번호를 배열에 저장시킨 후, DB를 실행시켜 수정을 완료한다.</p>
비밀번호 수정 에서의 if문	<p>이 if문은 사용자가 새로 설정하고 싶은 비밀번호를 입력하고 나서, 다시 새로 입력한 비밀번호를 확인할 수 있는 부분에도 똑같이 입력을 하고 만약 새로 설정하고 싶은 비밀번호 입력과 확인 입력이 같다면 DB를 열 수 있게 하여 다시 한 번 사용자에게 새로 바뀐 비밀번호를 기억하게끔 만들었다.</p>

2. DrawerLayout 적용

DrawerLayout	메인페이지에서 메뉴를 구현을 위한 역할
DrawerLayout 이란 & 사용한 이유	<p>메인페이지에서 화면의 특정 영역에서 동적으로 열리고, 닫히는 사용자 인터페이스를 구현하기 위해서 DrawerLayout라는 안드로이드 레이아웃 관련 위젯을 사용했다. DrawerLayout란 서랍이 열리고 닫히는 것처럼, 평소에는 화면의 한쪽에 숨겨져 있다가 사용자가 액션을 취하면 화면에 나타나는 기능을 만들 수 있게 해주는 레이아웃이다. DrawerLayout에 추가된 자식레이아웃이 DrawerLayout의 영역 안에서 Drawer와 같은 동작을 수행한다. 자식레이아웃들 중 layout_gravity 속성 값을 가지지 않은 자식레이아웃이 기본적으로 표시되는 주 화면으로 취급되고, layout_gravity 속성 값을 가진 자식레이아웃만이 Drawer로써 동작한다.</p> <p>이 앱에서는 메뉴라는 부분을 추가하기 위해서 사용했다. 메뉴란 사용자의 정보를 보여 주고 앱의 전체 모든 곳(설정, 톡톡, 레시피, 마켓, 마이페이지)을 다 이동할 수 있게 해주는 기능이다.</p>
DrawerLayout 적용 방법	<p>메인 페이지의 전체 레이아웃을 DrawerLayout으로 설정한 후, 메인 화면과 메뉴 화면으로 나누어 만든 다음 액션 바 대신 안드로이드 위젯인 툴바를 이용하여 서랍 역할을 하는 레이아웃을 열고 닫을 수 있게 한다. 이 때 툴바를 사용하기 위해서는 스타일에서 액션바를 안 쓰도록 바꿔준 후 사용한다. 그 다음 메인페이지의 코틀린 파일에서 드로어 기능을 사용하기 위해 ActionBarDrawerToggle이라는 구문을 추가하여 DrawerLayout의 드로어 기능을 ToolBar 에서 제어할 수 있도록 연결 해준 후, 드로어 리스너를 이용하여 스트링 값으로 받아온 open과 close로 열고 닫힘을 행할 수 있게 한다. 최종적으로 현재 Drawerlayout 상태와 ActionBarDrawerToggle 의 상태를 맞춰줌으로써 기능 구현 완료한다.</p>

3. LayoutInflater 적용

LayoutInflater	톡톡, 마켓, 레시피에서 각 서브 항목들을 눌렀을 때, 전체 페이지가 바뀌는 것이 아닌 서브 항목에 관련된 부분 레이아웃만 변경 되도록 하는 역할
LayoutInflater 사용한 이유	예를 들어 마켓의 메인 페이지 내에서 여러 서브 항목을 넘어갈 때, 각각의 액티비티를 계속 호출하여 실행 하는 방식이 아닌 동적으로 화면을 필요할 때만 구성하여 앱의 메모리 측면에서 좀 더 빠르게 작동 될 수 있도록 하기 위해 사용했다.
LayoutInflater 적용 방법	View inflate 라는 함수를 통해 현재 내가 추가하고 싶은 레이아웃 파일과 attachToRoot를 false로 설정하여 root가 생성 되는 View의 Layout의 매개변수로 사용하여 적용시킴으로써 자신이 루트가 되어 다른 부모 뷰에 속하지 않고 독립적으로 실행 되게 한다.

4. ListView 및 GridView적용

ListView 사용한 이유	톡톡에서 메인 페이지 및 각 서브 항목들에 추가된 글들이 리스트 형식으로 보이게끔 하기 위해 사용했다. 스크롤 가능한 글들을 보이기 위해서 사용했다.
GridView 사용한 이유	마켓, 레시피에서 메인 페이지 및 각 서브 항목들에 추가된 글들이 2차원 그리드 형식으로 보이게끔 하기 위해 사용했다. 스크롤 가능한 글들을 보이기 위해서 사용했다.

<p>GridView & ListView 적용 방법 (Adapter를 이용)</p>	<p>먼저 데이터 클래스를 정의한다. 리스트 뷰에 띄울 데이터를 선언해준다. 그리고 실제 변수를 초기화한 후 띄우고 싶은 항목을 가지고 있을 ArrayList를 Adapter에 추가해준다.</p> <p>ListView의 각 item 및 거기에 넣을 데이터까지 설정했으면 Adapter를 만들어야 한다. 어느 항목을 어느 View에 넣을 것인지 연결해주는 것이 Adapter의 역할이다. Adapter의 클래스 파일을 만들어준 후, 임의로 Custom한 Adapter Class에 BaseAdapter()를 상속(extend)받는다.</p> <p>Adapter에서 각 메소드가 담당하는 기능을 추가해준다.</p> <ul style="list-style-type: none"> • getView: xml 파일의 View와 데이터를 연결하는 핵심 역할을 하는 메서드이다. 이 메서드에서 항목 뷰는 convertView에 해당이 되는데, 이 객체가 아직 생성이 되지 않은 경우에는 LayoutInflater를 이용하여 ()_item.xml파일을 호출하여 항목 뷰의 객체를 생성하도록 하고 있습니다. 그런 다음, ImageView, TextView등의 객체를 찾아서 각각 그림과 텍스트를 대입한다. 어느 위치에 리스트를 넣을지 정해진 후, 리스트의 각 항목을 매핑하여 데이터를 반영해줌으로써 Adapter를 완성한다. <p>- Adapter의 다른 방법들</p> <ul style="list-style-type: none"> • getItem: 해당 위치의 item을 가져오는 메서드이다. Int 형식으로 된 position을 파라미터로 갖는다. • getItemId: 해당 위치의 항목 id를 반환하는 메서드이다. • getCount() : ListView에 속한 항목의 전체 수를 반환한다. • AddVO() : ListView에 속한 항목에 데이터 추가하는 함수이다. • clearVO() : ListView에 속한 항목에 데이터를 변경하기 위해 clear 하는 함수이다. <p>Adapter를 만들었다면, Adapter를 사용할 코틀린 파일에서 Adapter를 초기화해야 한다. 그 후 ListView 또는 GridView와 Adapter를 연결한다.</p> <p>만약 글 등록이 되었다면 홈 화면에 만들어진 어떤 카테고리에 넣어서 뷰를 보일지 결정해야 하므로, 우선 select 메서드를 통해 특유의 테이블의 내용을 찾게 다고 한 뒤, 커서를 이용해서 현재 내가 등록하고 싶은 카테고리의 텍스트 값과 DB에 저장된 카테고리 이름을 비교한다. 그런 다음 Adapter를 이용하여 리스트에 DB에 저장된 내용인 데이터를 추가한다. 그러면 이제 ListView 또는 GridView로 원하는 뷰가 보인다.</p>
--	--

그 밖의 기능 구현들

- AlertDialog

Dialog 사용한 이유	다이얼 로그를 이용하여 로그인을 안 했을 시, 앱의 모든 곳의 기능을 막아 사용자가 필수로 로그인을 하도록 알림을 뜨게 만들기 위해 사용했다.
Dialog 적용 방법	Dialog의 기능 중 AlertDialog을 이용하여 구현을 했다. Dialog 화면의 제목과 메시지 내용 및 사진을 설정 해준 뒤, Dialog 인터페이스를 통해 Positive button눌렀을 시, 로그인 페이지로 이동이라는 메시지를 toast를 통해 띄우고 로그인 페이지로 이동한다, Negative button이 눌렀을 시, 로그인 페이지 이동 취소라는 메시지를 toast를 통해 띄우고 Dialog 화면이 꺼지게끔 적용했다

- Spinner

Spinner 사용한 이유	Spinner을 이용해서 마켓의 어느 카테고리를 선택할지 고르게 할 수 있게 하기 위해서 사용했다
Spinner 적용 방법	기본적으로 ArrayAdapter는 현재 Context와 사용할 LayoutResource, ArrayList를 받아 스피너를 만들어준 후, Spinner ID를 참조시킨 Spinner 객체에 Adapter를 세팅하고 적용해주면 Spinner를 사용 할 수 있게 된다. 이 때, 사용한 리스너 객체는 OnItemSelectedListener으로 OnItemSelectedListener를 이용해 선택된 Item의 position값(0부터 시작)을 가져와 Spinner객체의 getItemAtPosition() 메서드를 사용해 해당 포지션에 세팅된 내용을 얻어낸다. 이로써, 스피너의 아이템 클릭 시 원하는 카테고리로 DB에 설정 되고 화면에 카테고리 이름이 출력된다.

- 사진을 가져오기 위한 디바이스 사진첩 접근 허용

<p>사진첨 접근 사용한 이유</p>	<p>마켓이나 레시피 부분에서 글 등록을 할 때, 자신이 올리고픈 내용에 대한 사진이 첨부되도록 구현하고 싶어 사용했다.</p> <p>이로써 자신의 기기에서 촬영한 사진까지 첨부할 수 있게 된다.</p>
<p>사진 불러오기 접근 적용 방법</p>	<p>이미지 버튼을 클릭한다면 우선, 이미지를 가져오게 하는 메서드의 버전을 체크한다. 만약 버전이 일정 버전 이상이면 허용 코드를 넘겨주어 이미지를 가져올 수 있게 세팅해준다. 만약 버전이 일정 버전 이하로 낮게 된다면 버전이 낮은 채로 이미지를 가져올 수 있게 세팅해준다.</p> <p>만약 이제 이미지 불러오기가 허가가 된다면 원하는 이미지를 갤러리에서 찾아 pic이라는 변수에 저장한다. 이 때도 이미지를 가져올 수 있는 코드를 넘겨주어 가져오게 된다. 이제 만약 요청한 이미지 불러오기 허가와 이미지를 가져올 수 있는 코드를 둘 다 받아 만약 둘 다 가능한 코드라면 이미지의 URL을 받아와 스트링에 저장한 후, 그 URL에 맞는 이미지를 사진 파일로 띄워 준다.</p>

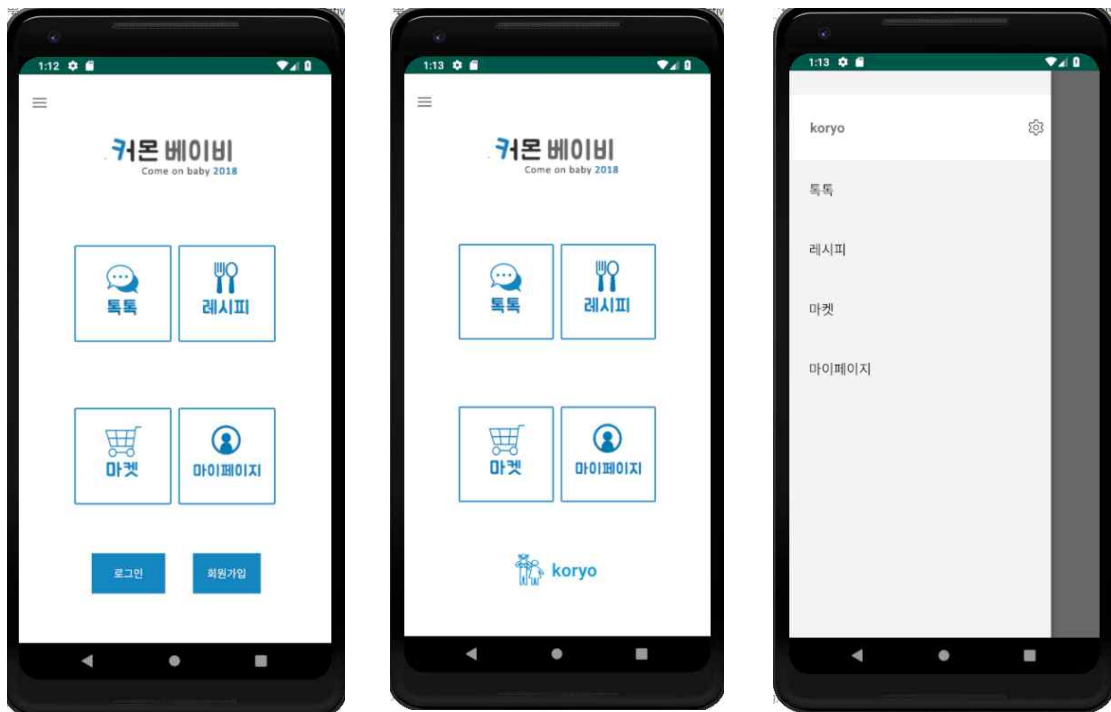
5 디자인 전체 필수 구조

- 모든 화면 공통 디자인(메인 화면 및 비밀번호 수정 화면 부분 제외)

모든 화면에는 위 상단에 앱 로고와 함께 메인페이지로 이동하는 홈 버튼이 위치하도록 디자인 했다.

- 메인 화면

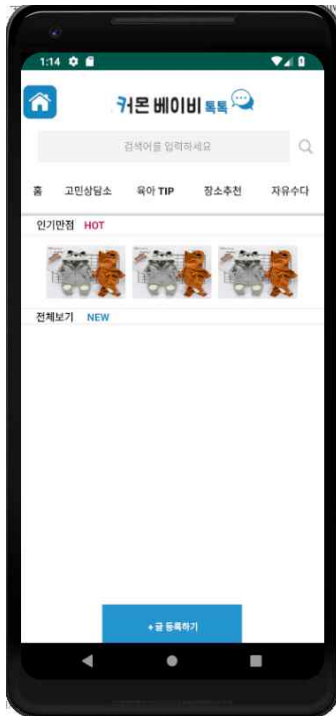
메인 화면은 크게 로그인 했을 때의 화면과 안 했을 시의 화면으로 크게 두 가지 버전이 있다. 각 메인 화면은 DrawerLayout이 큰 화면을 감싸고 있으며, drawer 기능이 있는 자식 Linear Layout과 주 화면을 이룰 자식 Linear Layout으로 구성 되어 있다. 주 화면 레이아웃 상단에는 앱의 로고와 톨바가 자리하고 있으면 밑으로 톡톡, 마켓, 레시피, 마이 페이지의 이미지를 담은 버튼이 2행 2열 형식으로 자리 잡고 있다. 드로어 화면은 유저의 아이디 및 설정으로 넘어가는 부분 및 각 하위 카테고리로 넘어 갈 수 있는 버튼이 나열 되어 있는 구조로 디자인 했다.



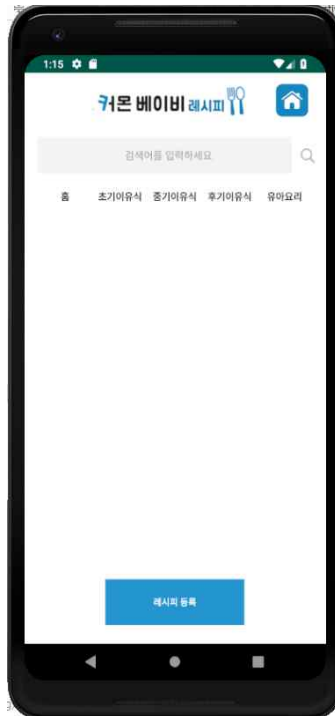
- <로그인 안 했을 시, 화면> <로그인 했을 시, 화면> <메뉴 화면-드로어 기능 화면>
- 톡톡, 마켓, 레시피 메인 화면 및 하위 카테고리 화면

톡톡, 마켓, 레시피 메인 화면은 각자 비슷한 디자인 구조를 띄고 있다. 우선 상단에 앱의 로고와 함께 밑에 각 톡톡, 마켓 레시피의 하위 카테고리로 넘어갈 수 있는 버튼들이 존재 한다. 각 하위 카테고리들은 인플레이터를 통해 각 하위 카테고리에 맞는 부분만 디자인이

변경 될 것 이므로 톡톡은 리스트 뷰 형식으로 디자인이 되고, 마켓과 레시피는 그리드 뷰 형식으로 디자인 된다. 리스트 뷰와 그리드 뷰는 각각 스크롤 뷰 또한 포함하고 있으며, 그리드 뷰에서는 한 행에 몇 개의 열 항목을 추가해서 보여줄지 또한 설정해서 디자인 했다.



<톡톡 홈 화면>



<레시피 홈 화면>

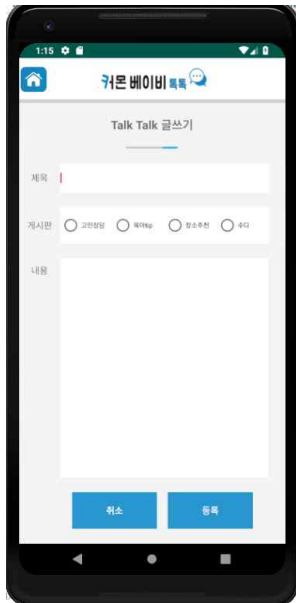


<마켓 홈 화면>

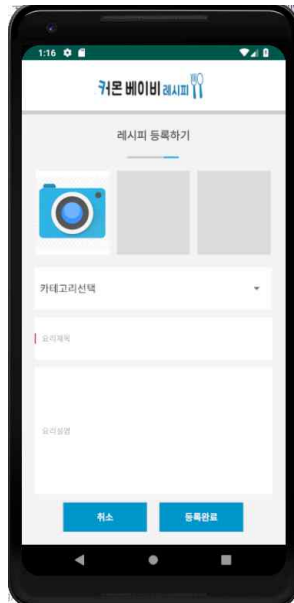
- 톡톡, 레시피, 마켓 글 등록 화면

글 등록 화면은 크게 글의 내용만 다를 뿐 전체적인 디자인은 비슷하다. 예를 들어 톡톡과 같이 글의 제목, 카테고리 선택, 글의 내용과 같이 사진이 필요 없는 부분에서는 3가지 레이아웃으로만 되지만 레시피 및 마켓과 같이 글의 전반적인 내용 및 제목, 사진, 카테고리 이렇게 사진이 추가된 레이아웃 경우 총 4가지의 레이아웃으로 디자인 했다.

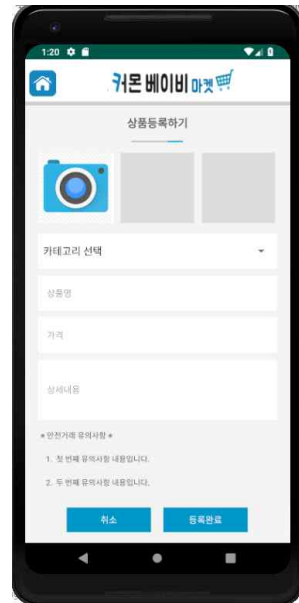
<톡톡 글 등록 화면>



<레시피 글 등록 화면>



<마켓 글 등록 화면>

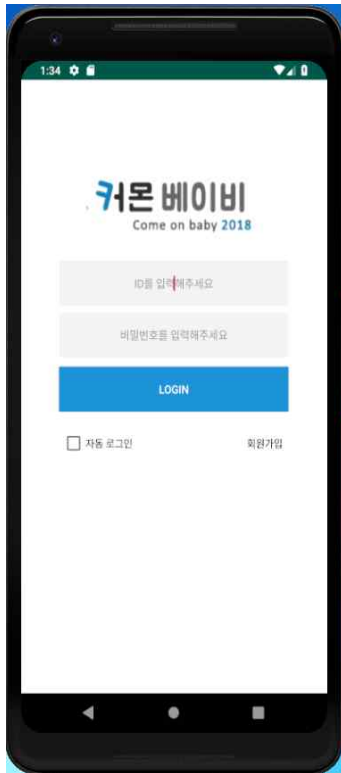


- 로그인 화면

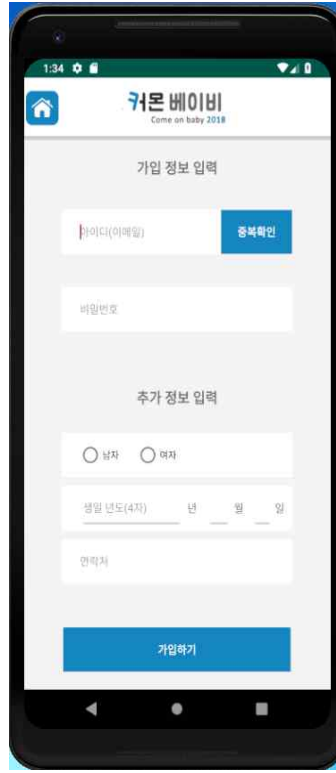
로그인 화면은 크게 아이디와 비밀번호를 입력하는 부분과 밑에 로그인 정보를 저장하는 체크 버튼 및 회원가입을 할 수 있는 화면으로 넘어가는 버튼 이렇게 구성되어 디자인 했다.

- 회원 가입 화면

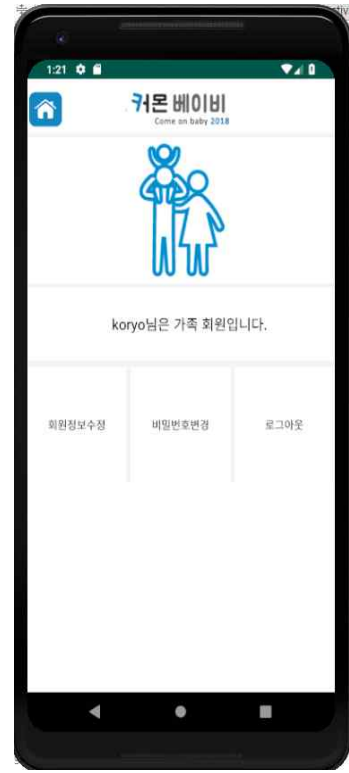
회원 가입 시에는 아이디, 비밀번호, 생년월일, 전화번호를 입력할 수 있는 EditText와 성별을 저장하는 RadioButton, 아이디를 중복 확인 할 수 있는 버튼으로 크게 구성되어 있으며, 만약 아이디 중복 확인을 할 시, 아이디 입력하는 칸 아래에 아이디 중복입니다 또는 사용 가능한 아이디입니다. 라는 것을 띄워 주는 구문이 보이도록 디자인 했다



<로그인 화면>



<회원가입 화면>



<마이페이지 화면>

- 회원 정보 수정

(1) 전화 번호 수정

전화 번호를 수정하는 페이지에서는 아이디, 성별, 생년월일, 전화번호 정보를 띄워주는 TextView와 전화 번호를 수정할 수 있는 EditText, 수정하기 Button이 디자인 했다.

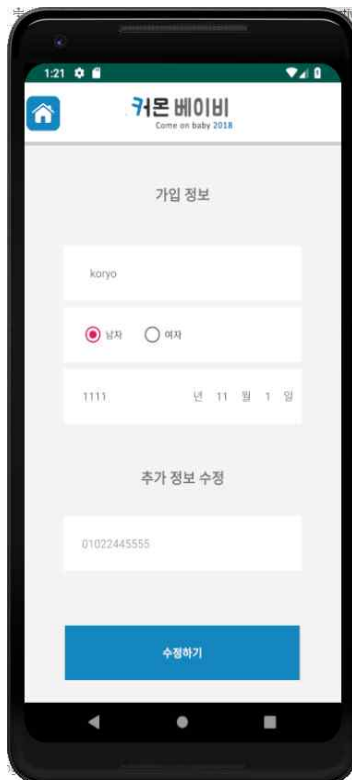
(2) 비밀 번호 수정

이 부분은 새로운 비밀번호와 새로 입력한 비밀번호를 확인하는 EditText가 있으면 밑에 Button을 두어 수정과 수정 취소를 할 수 있도록 디자인 했다.

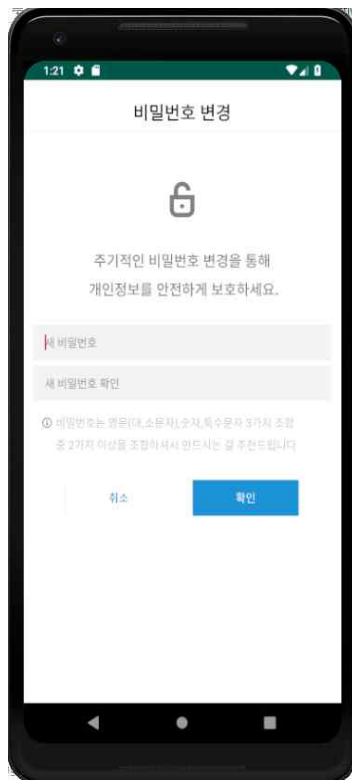
- 설정 화면

설정 화면은 사용자 아이디 정보를 나타내는 TextView, 그 옆에는 로그아웃을 할 수 있게 하는 Button, 밑으로는 문의 사항 메일, 버전 정보, 개발자 정보를 나타내는 TextView로 구성되어 디자인 했다.

<회원 정보 수정 화면>



<비밀 번호 수정 화면>



<설정 화면>

