**캡스톤 디자인**

**프로젝트 최종 보고서**



Subject : 캡스톤 디자인

Professor : 박상오 교수님

Team : 9조

Member : 20134695 김명규

20130895 성호준

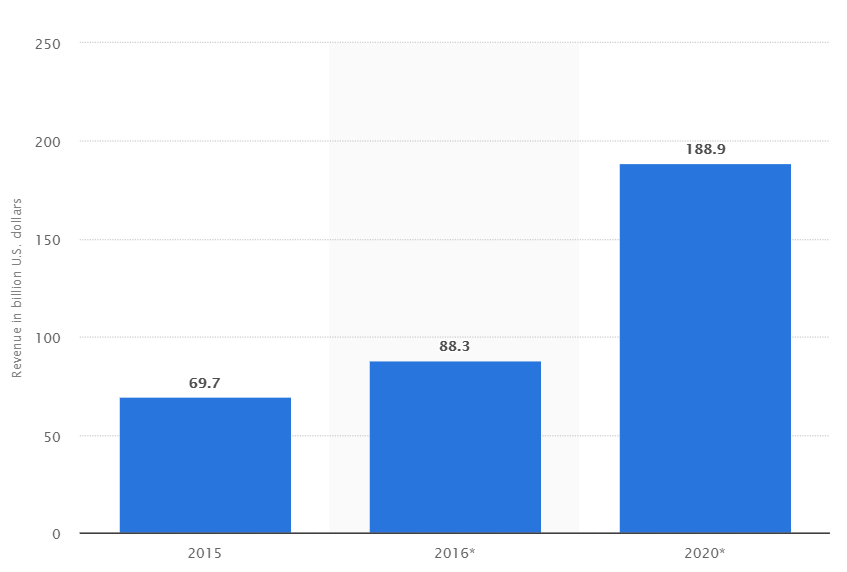
20161943 임민정

**프로젝트 소개**

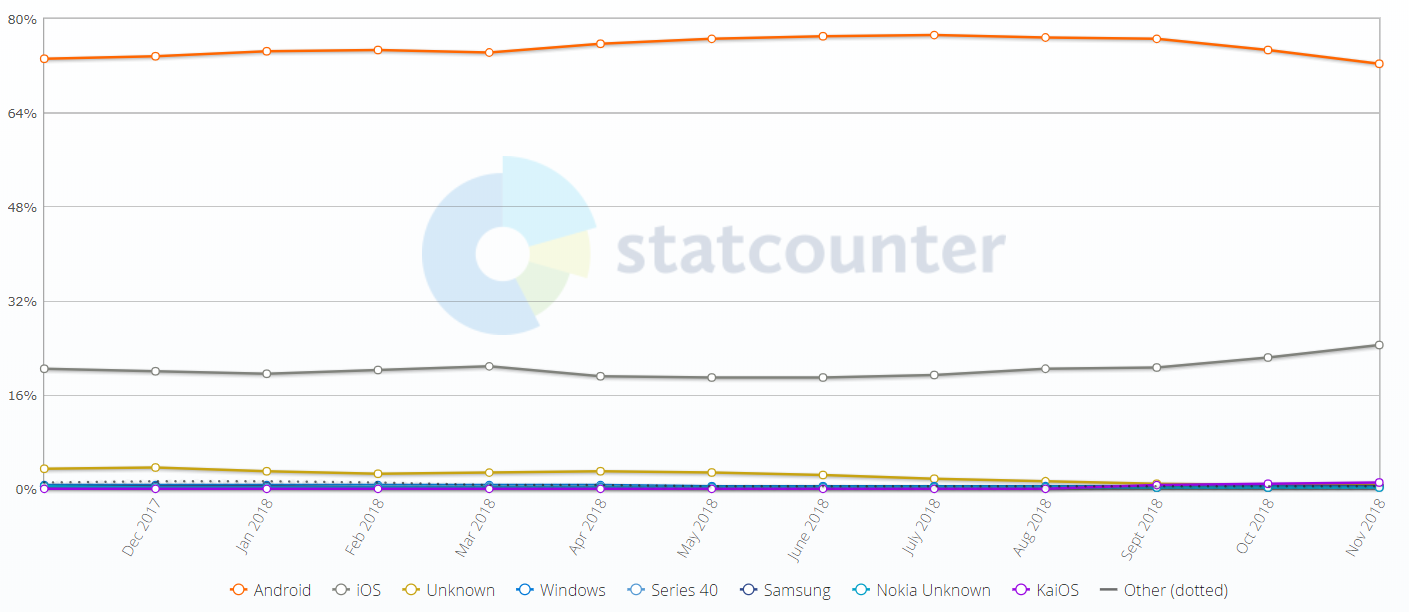
안드로이드 기반 사용자 참여형 학교 주변 방 구하기 모바일 어플 “HomeSeek” 을 개발한다. 현재 시중에 나와 있는 부동산 어플리케이션을 조사한 결과, 공인중개사와 연계하여 월세 이외에도 전세, 일반 거래 등 다양한 종류의 거래 매물을 취급한다. 이러한 경우, 자취하는 학생들의 입장에서 사용하기에 불편한 점이 있으며, 또한 중간에 법정중개수수료가 발생하는 경우도 있다. 따라서 학생의 입장에서 사용하기 편하고, 중간에 공인중개사를 거치지 않아 수수료가 들어가지 않는 참여형 모바일 어플을 제작한다.

**모바일 어플리케이션 선정 배경**

현재 모바일 어플 시장은 성장 가능성이 높은 시장 중 하나이다. 다양한 모바일 기기의 보급률이 높아짐에 따라 모바일 어플 제작의 수요가 늘어나고, 기존 PC 기반 프로그램의 모바일 어플리케이션 버전 제작도 진행됨에 따라 시장의 성장성이 높다. 이러한 시장의 성장성을 바탕으로, 모바일 어플 시장에서 얻어지는 순이익의 크기도 늘어나는 추세이며 2020년대에 2000억 달러에 약간 모자라는 수준까지 늘어날 것으로 예상된다.



다양한 모바일 기기들 중, 실생활에서 가장 많이 쓰이는 기기는 스마트폰이다. 스마트폰은 현재 단순 통신과 더불어, 동영상 시청, 게임, 카드결제 등 다양한 용도로 사용된다. 스마트폰에서 사용하는 OS는 다양하지만, 안드로이드, iOS 2가지가 시장점유율의 대부분을 차지한다.



이 중, 안드로이드의 경우 시장의 70 퍼센트 이상을 점유하고 있는 것으로 나타난다.

이러한 점들을 바탕으로, 모바일 어플리케이션에 대하여 알아보고, 그 제작을 경험해 보기 위해 안드로이드 기반 모바일 어플리케이션 제작을 주제로 선정하였다.

**핵심 기능 설명**

핵심 기능은 크게 4가지로 분류할 수 있다.

1. 회원정보 관리 기능

회원정보 관리를 위해 아래와 같은 기능을 구현하였다.

* 회원가입 및 ID 중복 체크 기능
* 로그인 기능
* 데이터베이스 상 비밀번호 암호화 기능
* 회원탈퇴 기능

회원가입 기능은 사용자 데이터베이스에 새로운 정보를 입력하기 위한 과정이다. ID, PW, 닉네임, 전화번호, 사용자 타입 등의 정보가 들어가며, ID 중복체크 기능의 경우, 회원가입 완료 전 데이터베이스에 같은 ID가 이미 사용되고 있는지 여부를 확인하기 위해 사용한다. 사용자 타입의 경우 Buyer와 Seller로 나누어 구현하였으며, 사용자 타입에 따라 사용가능한 기능에 차이가 있다.

로그인 기능은 입력 받은 ID와 PW가 데이터베이스에 입력된 정보와 일치하는지 확인하는 기능이다. 이 때, ID가 입력되어 있고, PW까지 완벽히 일치해야 로그인이 성공한다.

데이터베이스 상 비밀번호 암호화 기능은 데이터베이스에 PW 등의 개인정보를 암호화하는 기능이다. 이를 통하여 데이터베이스 해킹 등의 문제를 방지하였다.

회원탈퇴 기능은, 사용자가 원할 경우 데이터베이스 상에서 해당 사용자와 관련된 내용을 모두 삭제하는 기능이다. 여기에는 단순 사용자 정보 뿐만 아니라 사용자가 작성한 매물이나 게시판 글까지 모두 삭제한다.

2. 매물관리 기능

매물 관리를 위해 아래와 같은 기능을 구현하였다.

* 매물 추가 기능
* 매물 수정 기능
* 즐겨찾기 기능
* 매물 감추기 기능
* 매물 삭제 기능

매물 추가 기능은 데이터베이스에 새로운 매물을 추가하는 기능이다. 이 때, 기본적인 텍스트 정보는 Mysql에, 사진 등의 이미지 정보는 Google Firebase를 이용하여 저장한다.

매물 수정 기능은 데이터베이스에 이미 추가된 매물의 정보를 수정하는 기능이다. 해당 기능을 이용하여 내용을 수정하면, 데이터베이스 내부의 기존 정보가 수정되어 저장된다.

즐겨찾기 기능은 사용자 ID를 기준으로 관심있는 매물을 그룹화하는 기능이다. 아래의 검색 기능과 연계하여 상세 정보 표시 페이지에서 설정할 수 있으며, 그룹화 된 매물들은 My page 에서 모아서 확인할 수 있다.

매물 감추기 기능은 잠깐동안 사용하지 않는 매물을 검색 결과에 표시하지 않도록 변경하는 기능이다. 집이라는 매물 특성 상 매물의 대표적인 특징들은 변하지 않는데, 이미 거래가 이루어져 사용중인 경우 해당 매물을 감추어 표시되지 않도록 변경한다. 이 후, 계약이 종료되어 다시 임차인을 구할 경우, 다시 검색결과에 표시되도록 변경할 수 있다.

매물 삭제 기능은 사용자가 원할 경우, 데이터베이스에 저장된 매물 정보를 삭제하는 기능이다. 해당 기능을 사용하면 매물 정보와 이미지 파일이 전부 삭제되게 된다.

검색 기능은 특정 조건을 만족하는 매물을 데이터베이스에서 가져와 표시하는 기능이다. 이와 더불어 방의 넓이, 월세 등의 정보를 기준으로 정렬하는 기능을 추가로 구현하였다.

3. 검색 및 지도 기능

검색 및 지도 기능을 위해 아래와 같은 기능을 구현하였다.

* 매물 검색 기능
* 지도 검색 및 표시 기능
* 길찾기 기능

매물 검색 기능은 특정 조건을 만족하는 매물을 데이터베이스에서 가져와 표시하는 기능이다. 이 때, 방의 넓이, 월세 등의 기준으로 정렬하는 기능을 추가로 구현하였다. 이 후 표시된 매물을 클릭하여 상세 정보를 확인할 수 있다.

지도 검색 및 표시 기능은 위의 매물 검색 기능과 연계하여 매물들을 지도에 띄워주는 기능이다. 검색 기능과 마찬가지로 표시된 마커를 클릭하여 상세 정보를 확인할 수 있다. 또한 상세 정보에서 해당 주소에 마커를 표시하여 해당 매물이 어느 장소에 위치하는지 시각적으로 나타낸다.

길찾기 기능은 해당 매물에서 학교까지의 길을 표시하는 기능이다. 출발지와 목적지는 변경할 수 없으며, 길찾기 기능을 실행 시, 결과까지 지도에 표시되어 출력된다.

4. 게시판 기능

게시판 기능을 위해 아래와 같은 기능을 구현하였다.

* 게시판 목록 기능
* 게시글 작성기능
* 게시글 수정 기능
* 댓글 기능
* 게시글 삭제 기능

게시판 목록 기능은 데이터베이스에 저장된 게시판 글 데이터 전체를 불러와 표시하는 기능이다.

게시판 글 작성 기능은 새로운 게시글 데이터를 데이터베이스에 추가하는 기능이다.

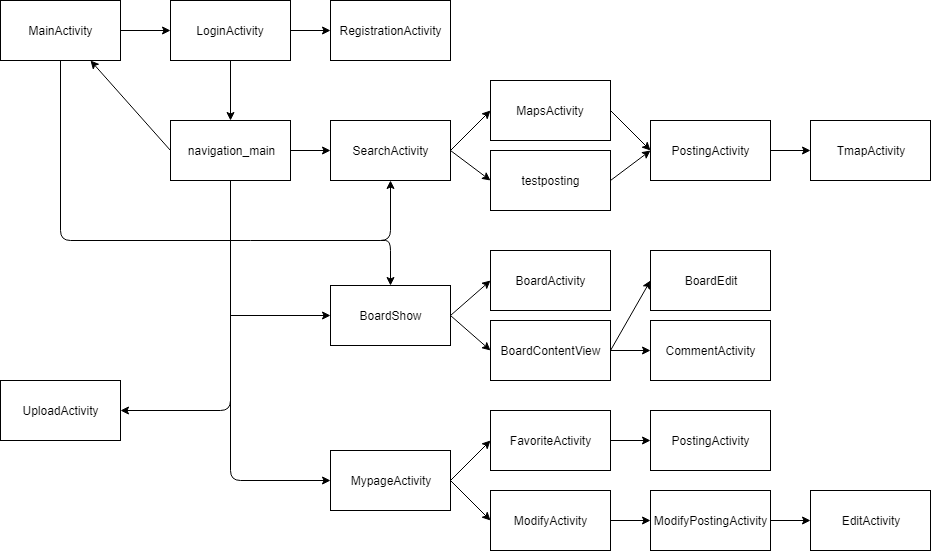
게시판 글 수정 기능은, 이미 작성된 게시글의 내용을 수정하는 기능이다.

댓글 기능은 작성된 게시글에 댓글을 추가하는 기능이다. 이 때 화면을 아래로 스와이프하여 새로고침하는 기능을 추가로 구현하였다.

게시판 글 삭제 기능은 사용자가 원할 경우 데이터베이스에 저장된 게시글 정보를 삭제하는 기능이다.

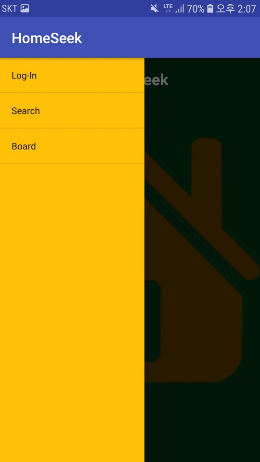
**액티비티 / 클래스**

간단한 액티비티 간 흐름은 아래와 같다.



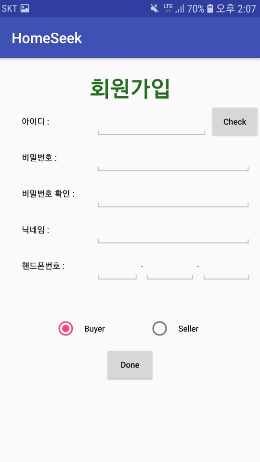


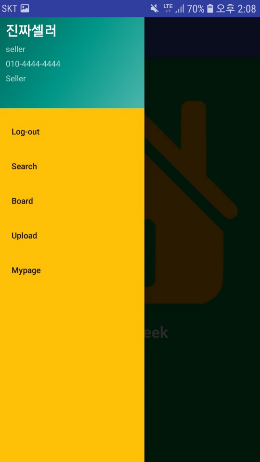
어플리케이션을 시작하게 되면 왼쪽과 같은 화면이 먼저 출력된다. 위의 표에 표시되지 않은 SplashActivity에서 구현된 화면으로, 일정 시간이 지나면 아래의 MainActivity 페이지로 이동한다.



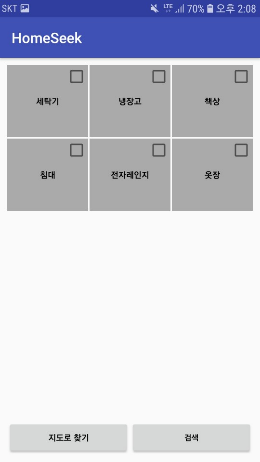
왼쪽의 화면은 MainActivity에서 구현된 메인 페이지이다. 기본적인 구조는 왼쪽 화면과 같으며, 왼쪽에 Drawer를 삽입하여 다른 기능들을 사용할 수 있도록 구현하였다. 해당 화면에서는 Drawer의 버튼을 이용하여 로그인페이지, 검색페이지, 게시판 페이지로 이동할 수 있다.

해당 화면은 LoginActivity에서 구현한 로그인 페이지이다. 사용자의 ID와 비밀번호를 입력받고, 이를 데이터베이스에 입력된 정보와 비교하여, ID와 비밀번호가 모두 일치할 경우 후술할 nav\_main 페이지로 이동한다. ID와 비밀번호가 일치하지 않을 경우, 오른쪽의 그림처럼 로그인 실패 메시지를 출력한다. 해당 화면에서는 Log-in버튼을 이용해 nav\_main으로, Sign up 버튼을 이용해 회원가입 화면으로 이동할 수 있다.

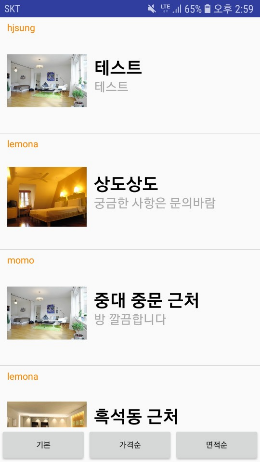
해당화면은 RegistrationActivity에서 구현한 회원가입 페이지이다. 사용자의 정보를 입력받고, 이를 데이터베이스에 추가하는 화면이다. ID 옆의 Check 버튼을 통하여, 입력한 ID 가 데이터베이스에 이미 추가되어 있는지 확인할 수 있다. ID Check를 반드시 완료해야 아래의 Done버튼을 통해 완료할 수 있으며, ID Check를 통해 ID가 중복되지 않은 것을 확인한 경우, ID를 더 이상 변경할 수 없도록 고정한다. 아래의 Done 버튼을 누르면 입력받은 정보를 데이터베이스에 추가하고 로그인 페이지로 돌아간다.



로그인 페이지에서 정확한 회원정보를 입력하고 로그인 버튼을 누르면 왼쪽의 nav\_main 페이지로 이동한다. 기본 구조는 MainActivity와 동일하고 Drawer 위쪽에 회원정보를 표시하는 부분이 추가되고, 로그인 상태에서만 이용할 수 있는 Upload, Mypage 버튼이 추가된다. 또한 기존의 Log in 버튼이 Log out을 위한 버튼으로 대체된다. 이 때, Upload의 경우 사용자 타입이 Seller인 경우에만 사용이 가능하며, Buyer가 사용하려 할 경우, Seller만 사용가능하다는 다이얼로그 메시지를 출력한다. 해당 화면에서는 Drawer의 버튼을 통해 MainActivity와 동일하게 이동이 가능하며, 추가로 Upload, Mypage 버튼을 이용하여, 매물 추가 기능을 사용하고, 마이페이지로 진입할 수 있다.

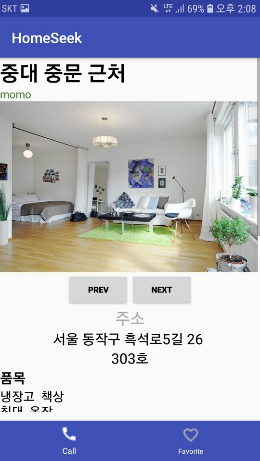
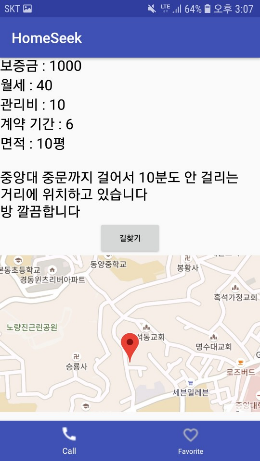
이 때, 회원정보는 User\_information 클래스를 활용하여 저장하고 추후에 활용한다.

해당 화면은 SearchActivity에서 구현한 검색 페이지이다. MainActivity 와 nav\_main 화면의 Drawer에서 왼쪽의 Search 버튼을 누르면 왼쪽의 페이지가 출력된다. Gridview, GridAdapter를 활용하여 체크박스들을 구현하였고, 이 체크박스들을 통해 본인이 포함하고 싶은 옵션들을 선택할 수 있다. PageAdapter 클래스를 이용하여 하단의 버튼들을 구현하였고, 왼쪽의 지도로 보기 버튼을 통해 MapsActivity로, 검색을 통해 testposting 화면으로 이동할 수 있다. 검색을 실행하면 데이터베이스로부터 모든 매물 정보를 불러와 파싱하여 List 클래스에 저장한다. 이 때, 아무 체크박스도 선택하지 않고 검색을 실행하면, 데이터베이스에 있는 모든 매물이 검색 결과로 표시된다.

해당 화면은 testposting에서 구현한 화면이다. 검색을 실행하여 List에 저장된 모든 정보들 중, 조건을 만족하는 매물들을 posting list 클래스를 활용하여 구분하고, 이를 postholder 클래스와 PostingTIlte 클래스를 이용하여 화면에 글자로 표시한다. 또한 사진들의 경우 MyappGlideModue, Glide Client 클래스를 활용하여 데이터베이스에서 가져온다. 아래쪽의 버튼들을 이용하여 원하는 기준으로 정렬할 수 있으며, 매물 하나하나를 클릭하면 해당 매물의 정보를 가진 PostingActivity 화면으로 이동할 수 있다.



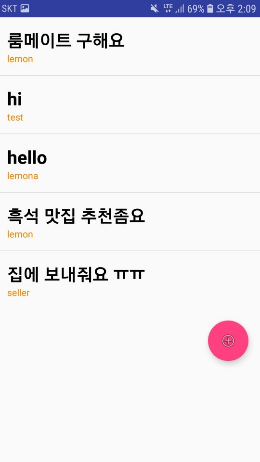
해당 화면은 MapsActivity에서 구현한 지도 화면이다. SearchActivity 화면에서 이동할 수 있으며, Testposting의 posting list와 동일한 방식으로 조건에 맞는 매물들을 구분하고, 이를 MapsEstatelist 클래스를 활용하여 저장한다. 저장된 정보는 구글맵 api를 통해 구현한 지도에 marker로 표시한다. 이 때, 특정 marker를 선택하면 해당 매물의 정보를 가진 PostingActivity로 이동할 수 있다.

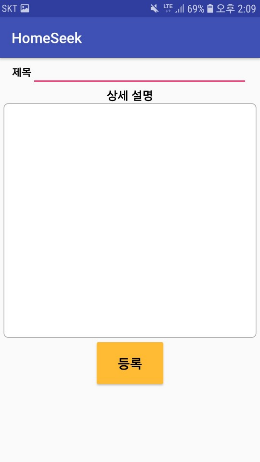
해당 화면은 PostingActivity에서 구현한 매물 상세 정보 화면이다. 위에 설명한 testposting, MapsActuvity에서 이동할 수 있으며, 인텐트로 매물 정보가 같이 전달된다. 해당 정보를 이용하여 내용을 표시한다. 왼쪽 그림의 중단의 PREV, NEXT 버튼을 이용하여 사진을 전환하여 볼 수 있으며, 매물을 올릴 때 입력한 정보들을 볼 수 있다. 아래쪽에는 구글맵 api 를 활용하여, 해당 매물의 위치를 표시해서 보여주며, 길찾기 버튼을 통하여 TmapActivity로 이동할 수 있다. Posting Page 클래스를 이용하여 페이지 하단의 Call, Favorite 버튼의 기능을 구현하였으며, Call 버튼을 클릭하면 스마트폰 자체의 통화 기능을 활용하여 판매자의 번호로 전화를 걸 수 있으며, Favorite 버튼을 클릭해서 즐겨찾기 기능을 사용할 수 있다. 이 때, 데이터베이스에 즐겨찾기로 해당 매물이 이미 들어가 있으면, 해당 즐겨찾기가 삭제된다.

해당 화면에 진입할 때, 저장된 주소를 구글맵 api를 이용하여 지오코딩하고, 결과로 나온 위도, 경도 값을 저장한다. 저장된 위도, 경도값은 위에 설명한 길찾기 기능을 사용할 때 TmapActivity로 같이 전송된다.

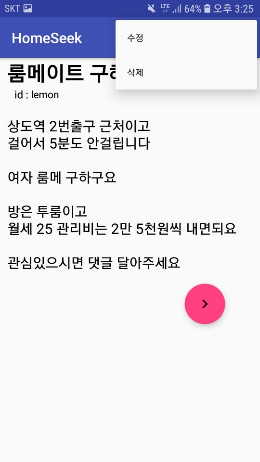


해당화면은 TmapActivity에서 구현한 길찾기 화면이다. 구글 지도의 경우, 한국에서 대중교통을 이용한 길찾기 기능만 지원하고, 보행자를 위한 길찾기 기능을 이용할 수 없기 때문에, 해당 기능은 Tmap api를 이용하여 구현하였다. 출발지는 학교로, 도착지는 해당 매물의 주소로 고정된다. PostingActivity에서 넘겨받은 위도 경도 값을 사용한다.

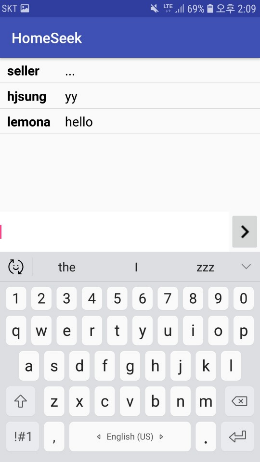
해당화면은 BoardShow에서 구현한 게시판 메인 화면이다. MainActivity나 nav\_main 화면에서 진입할 수 있다. BoardShow 화면에 진입하면, 데이터베이스로부터 모든 게시판 정보를 불러와 Board\_list 클래스를 이용하여 저장한다. 저장된 정보들은 BoardTitle 클래스를 이용하여 게시글로 하나하나 표시된다. 각각의 게시글을 클릭하면 해당 게시글의 정보를 가진 BoardContentView 화면으로 이동하며, 아래쪽의 분홍 버튼을 클릭하여 새로운 게시글을 작성하는 BoardActivity 화면으로 이동할 수 있다.

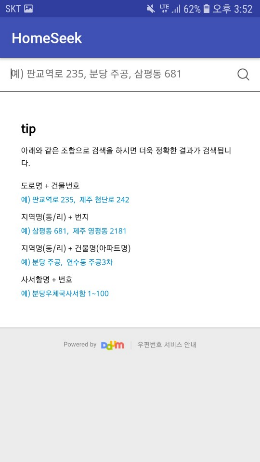


해당화면은 BoardActivity에서 구현한 게시글 등록 화면이다. 제목과 글 내용을 입력할 수 있으며, 내용을 모두 입력하고 아래쪽의 등록 버튼을 누르면, 입력한 내용을 게시판 데이터베이스에 추가한다. 또한 아래 설명할 게시글 수정 기능 사용시 BoardEdit 클래스와 함께 게시글 수정 역시 해당 화면에서 이루어진다.



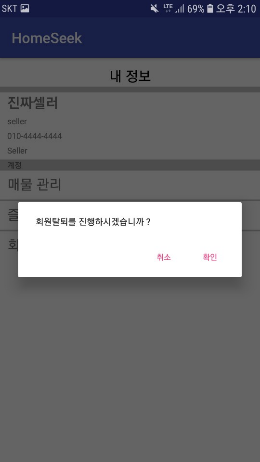
해당화면은 Board에서 구현한 게시글 상세내용 화면이다. 위의 BoardShow에서 각각의 게시글을 눌러 진입할 수 있으며, 제목과 작성자, 상세내용을 표시하여 준다. 우측 상단의 수정 버튼을 통해 게시글 내용 수정을 할 수 있는 BoardActivity 화면으로 이동하거나, 삭제를 통해 게시글 삭제를할 수 있으며, 하단의 분홍버튼을 이용하여 댓글을 작성하거나 볼 수 있는 CommentActivity로 이동할 수 있다.

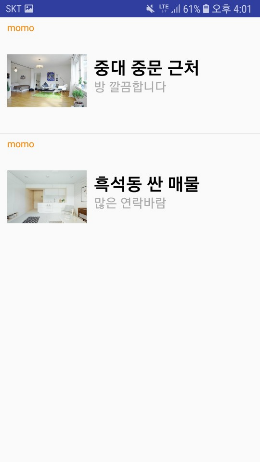
해당화면은 CommentActivity에서 구현한 댓글기능 화면이다. 입력된 댓글들을 볼 수 있으며, 간단한 댓글을 등록할 수 있다. 해당 화면 진입 시, 데이터베이스에 저장된 모든 댓글들을 가져와 Comment\_list 클래스를 활용하여 저장한다. 이 후 Comment 클래스를 이용하여 댓글 하나하나를 구성하여 표시한다. 댓글 작성 시, 입력한 내용을 Comment Holder 클래스에 임시로 저장한 후, 새로고침 기능을 실행시 데이터베이스에 추가하고 Comment\_list에 반영한다. 새로고침 기능은 화면을 아래로 스와이프해서 실행한다.

해당화면은 UploadActivity에서 구현한 매물 추가 화면이다. 사용자 타입이 Seller인 경우에만 사용이 가능하며, 제목, 사진, 주소, 포함 옵션 등 계약에 필요한 내용들을 등록할 수 있다. 사진의 경우, 상단의 + 버튼들을 이용하여 등록하며, 스마트폰 사진 갤러리에 있는 사진을 등록할 수 있다. 사진은 3장까지 등록이 가능하며, 3장이 모두 등록되어야 매물 추가가 가능하다.

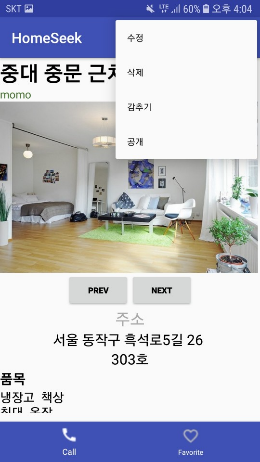
주소 검색 버튼을 눌러 주소를 검색할 수 있으며, 해당 기능은 다음지도 api 를 활용하였다. Daumaddress 클래스에 해당 내용이 구현되어 있다.

모든 과정을 마치고 하단의 등록버튼을 누르면 입력한 정보들이 데이터베이스에 추가된다. 이 때, 글, 옵션등의 정보는 Mysql 데이터베이스에, 사진은 Google Firebase에 저장된다.

해당화면은 MypageActivity에서 구현한 My page화면이다. 위쪽에는 로그인한 사용자 정보가 표시되며, 아래의 매물관리, 즐겨찾기 관리, 회원탈퇴 버튼을 클릭하여 해당 기능을 사용할 수 있다. 매물관리 버튼 클릭 시, ModifyActivity로, 즐겨찾기 관리시 FavoriteActivity로 이동하며, 회원탈퇴 버튼 클릭 시 오른쪽과 같은 다이얼로그 메시지를 출력한다. 회원탈퇴 실행 시, 유저 정보 및 작성한 매물, 게시글 등이 일괄적으로 모두 삭제된다.



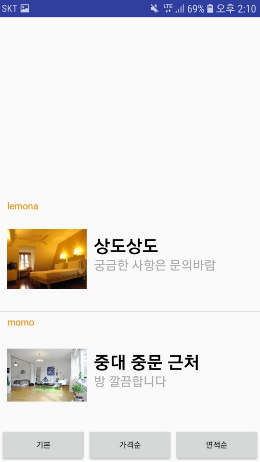
해당 화면은 ModifyActivity 에서 구현한 매물 관리 화면이다. MypageActivity의 매물관리 버튼을 통하여 진입할 수 있으며, 본인이 등록한 모든 매물의 목록을 표시한다. Testposting 과 유사하게 동작하며, 매물 하나하나를 클릭 시, 해당 매물의 정보를 가진 ModifyPostingActivity로 이동한다.



해당 화면은 ModifyPostingActivity에서 구현한 화면이다. PostingActivity와 유사하며, 우측 상단에 버튼들을 추가하여 매물 수정, 삭제 및 공개여부 선택 등의 기능을 지원한다. 수정 버튼을 클릭하여 매물 정보를 수정하는 EditActivity로 이동할 수 있다.



해당 화면은 EditActivity 에서 구현한 매물 수정 화면이다. UploadActivity와 유사하며, 화면 진입 시, 해당 매물의 정보가 들어간 상태로 시작된다. 이 때, 사진의 경우, 수정 시 Firebase 에 새로운 사진 정보로 등록되며, 그 외의 정보들은 Mysql에서 수정된다. 수정을 하지 않고 나갈 경우, 오른쪽과 같은 경고 메시지가 출력되며, 종료를 누르면 수정되지 않고, 이전 페이지로 돌아간다.



해당 페이지는 FavoriteActivity에서 구현한 즐겨찾기 관리 화면이다. 사용자가 등록한 즐겨찾기 정보가 모두 출력된다. 매물 하나하나를 클릭하면 해당 매물의 정보를 가지고 있는 PostingActivity로 이동한다.

**액티비티 스택 처리**

처음 어플리케이션을 제작할 때, 기능 구현 자체에 중점을 두고 프로젝트를 진행하였다. 중간 데모 이후, 각자 맡은 부분을 합치며 어플을 실행했을 때, 액티비티간 스택 관리를 하지 않아, 로그인에 성공한 후, 뒤로 가기 버튼을 누르면 다시 로그인 화면으로 돌아가는 등의 문제가 발생하는 것을 확인하였다. 이러한 상황을 처리하기 위해, 필요한 경우 다음 액티비티로 넘어갈 때 이전 실행중이었던 액티비티를 삭제하는 방식 등으로 액티비티 스택을 관리하여 어플리케이션이 보다 자연스럽게 진행되도록 구현하였다

**데이터베이스**

해당 프로젝트에서는 2가지 종류의 데이터베이스를 사용하였다.

첫 번째로, 단순한 텍스트의 경우 Mysql 데이터베이스를 사용하였다. 단순 텍스트만 저장하면 되기 때문에 사용하기 편한 Mysql 데이터베이스를 활용하였다. 총 8개의 테이블로 구성되어 있으며, 각각의 간략한 설명은 아래와 같다

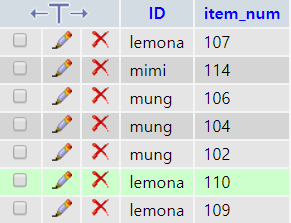
* homeseek\_user



homeseek\_user 테이블은 회원정보를 저장한 테이블로, 사용자 ID, 비밀번호, 닉네임, 사용자 타입, 전화번호 등을 저장한다. 이 때 , 비밀번호의 경우, 데이터베이스 해킹 등의 위험으로부터 개인정보를 보호하기 위해 PBKDF2 방식으로 암호화하여 저장하였다. 이후 비밀번호가 필요한 기능은 서버에서 입력한 비밀번호화 해쉬화 된 비밀번호를 대조하여 같은지 비교하는 방식으로 진행한다.

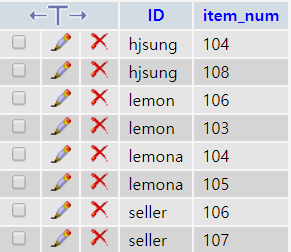
* homeseek\_user\_houselist

homeseek\_user\_houselist 테이블은 매물관리에 사용되는 테이블로, 특정 사용자가 올린 매물이 어떤 매물인지를 저장하고 있는 테이블이다. Upload 기능을 통해 매물을 등록하면, 매물정보를 저장하는 테이블과 더불어, 이 테이블에도 매물 정보가 저장된다.



* bookmark

bookmark 테이블은 즐겨찾기 기능에서 사용하는 테이블이다. 즐겨찾기를 추가하면 즐겨찾기를 등록한 ID와 매물의 번호를 매칭하여 저장한다. 즐겨찾기를 삭제할 경우, ID와 매물 번호를 검색하여 일치하는 행의 정보를 삭제한다.



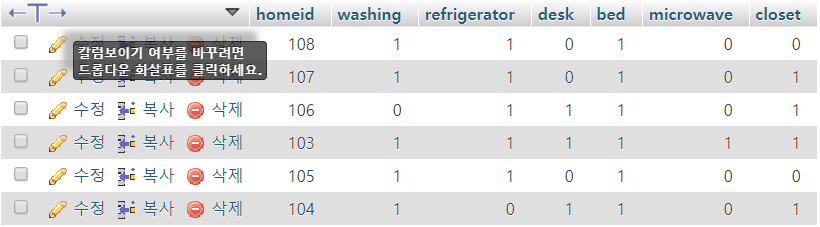
* home

home 테이블은 PostingActivity 등에서 사용되는 테이블로, PostingActivity 등에서 표시되는 상세한 정보들을 저장한 테이블이다. 주로 텍스트 관련 정보들이 저장되어 있다.



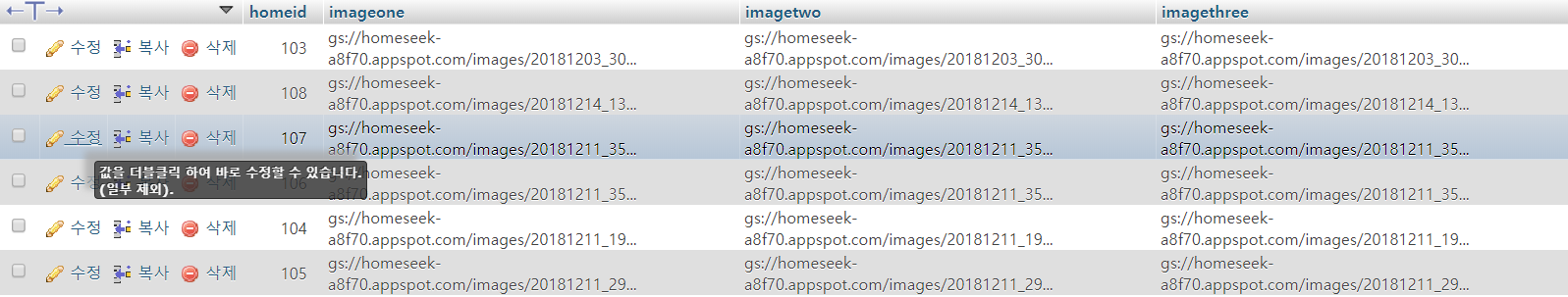
* roominform

roominform 테이블은 매물의 옵션과 관련된 정보를 저장하고 있는 테이블이다. 매물에 포함된 옵션을 기준으로 검색 시, 해당 테이블에 저장된 정보를 활용한다.



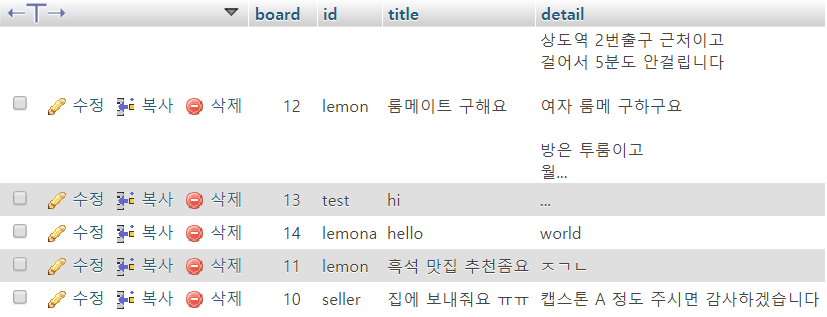
* image

image 테이블은 해당 매물의 사진이 저장되어 있는 Firebase 데이터베이스의 Url 을 저장하고 있는 테이블이다. 매물의 사진과 같은 이미지 파일은 Firebase에 따로 저장되는데, 해당 위치에 접근하기 위한 Url 을 저장한 테이블이다.



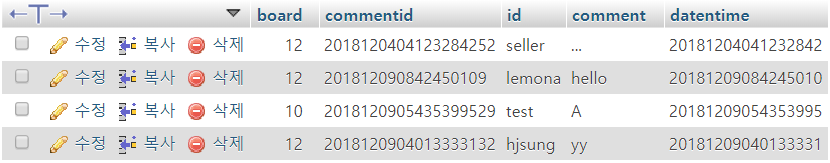
* board

board 테이블은 게시글을 저장하고 있는 테이블이다. 게시글의 작성자와 더불어 게시글의 내용을 저장하는 테이블이다.



* Comment

Comment 테이블은 게시글에 달린 댓글 정보를 저장하는 테이블이다. 어떤 게시글에 달린 댓글인지, 댓글을 단 사용자는 누구인지, 등록한 시간이 언제인지 등의 정보와 댓글의 내용을 저장한다.



두 번쨰로는 Google Firebase 데이터베이스를 이용하였다. 이미지 파일을 저장할 때, Mysql 의 경우, 속도도 느리고 이미지가 깨지는 현상이 발생하는 것을 확인하였다. 이를 보완하기 위해 이미지 파일 저장에 있어, 보다 안정적인 Firebase를 이용하여 이미지 파일을 저장하였다.

**데이터베이스 연동**

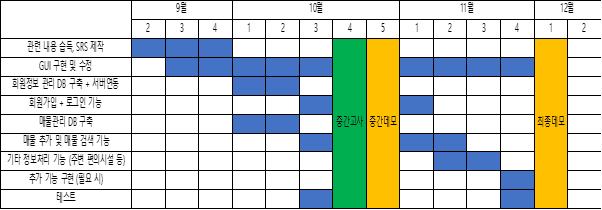
데이터베이스 관련 작업을 실행할 때는 Asynctask 클래스를 활용하였다. Asynctask 는 주로 백그라운드 작업을 위해 사용된다. 데이터베이스 접근은 어플의 동작과 별개로 보이지 않는 곳에서 이루어져도 어플 사용에 문제가 없으므로 Asynctask를 통해 병렬적으로 동작하도록 처리하였다. 어플 내부에 사용된 Asynctask의 경우 onPreExecute -> doInBackground -> onPostExecute 순서대로 동작하도록 구현하였으며, onPreExecute와 onPostExecute는 주로 다이얼로그 메시지 출력 등 어플쪽에서의 동작을 담당하며, doInBackground에서 서버 측과 통신하여 데이터베이스에 접근하는 방식으로 구현하였다.

서버 측에서 동작하는 코드는 php 코드로 작성하였다. 주로 어플로부터 넘겨받은 정보들을 이용하여 데이터베이스에 추가하거나 검색 및 수정, 삭제 기능을 지원하도록 쿼리를 구성하였다. 서버 측에서 작업을 끝마친 후, 작업 결과에 따라 특정 결과값을 반환하면 어플 측에서 반환한 결과값을 이용하여 특정 작입이 진행되도록 구현하였다.

**기타 프로젝트 정보**

|  |  |
| --- | --- |
| 사용 개발 도구 | Android Studio (3.1.2) – Android 코드 작성 / Atom (1.33.0) – php 코드 작성 |
| 웹 호스팅 | Cafe24, Dothome |
| 안드로이드 버전 | Kitkat (4.4) 이상 |
| 사용 Api | Google map api, T-map api, Daum map api, PBKDF2 open source |
| 깃 주소 | https://github.com/HJSUNG/capstone\_design |

**완성도**



프로젝트를 시작할 때, 기본적인 회원정보 및 매물정보를 데이터베이스와 연동하여 저장하고, 매물 검색 기능을 구현하고, 기타 정보처리 및 추가 기능을 구현하는 것을 목표로 하였다.

현재 회원정보 및 매물정보 데이터베이스를 구축하고, 이를 활용한 회원가입, ID중복체크 로그인 및 회원탈퇴 기능이 완벽하게 구현되었으며, 매물 정보의 추가, 수정, 삭제 기능 역시 구현되었다. 매물 검색 기능 역시 여러 옵션들을 포함하여 특정 조건(매물에 침대 옵션이 포함되는 경우 등)에 맞는 매물을 검색하는 기능까지 구현하였으며, 또한 검색 후 해당 매물을 지도에 띄우는 기능까지 구현하였다. 검색 후, 해당 매물의 정보에 접근하여, 상세 정보를 확인하고 상세 정보를 수정하고, 원할 경우 해당 매물을 삭제하는 기능을 구현하였다.

또한 프로젝트 진행 도중 받은 피드백을 바탕으로, Tmap api 를 활용한 길찾기 기능, 츶겨찾기 기능, 매물정보 감추기 기능, 게시판 및 댓글 기능 등을 추가로 구현하였다.

구현 내용을 미루어보아, 해당 프로젝트를 시작할 때 제시한 목표를 충분히 달성하였다고 생각된다.

**프로젝트를 마치며**

김명규 : 안드로이듣 어플 제작, 구글맵 api 사용, 데이터베이스 사용 등 수업에서 배우지 않았던 내용들을 인터넷으로 공부하면서 구현하여 어려웠지만 배운건 많았다. 나는 GUI와 구글맵을 중점적으로 다루면서 다른 팀원들이 작성한 코드들을 변경하거나 추가 할게 많았는데 다른 사람이 작성한 코드를 분석하면서 이용하는게 힘들었다. 추가로 이렇게 긴 프로젝트를 하는게 처음이라 매주 만나고, 정해진 목표를 맞추는 것이 힘들었다

성호준 : 안드로이드 어플 제작, 데이터베이스 사용, T-map api 사용 등 새로운 내용을 학습하며 구현하여 많은 것을 배울 수 있었다. 특히 mysql 데이터베이스를 처음 사용하면서, 사용하는 방법이나 php 코드를 짜는 방법을 몰라 많이 헤맸지만, 해당 내용들을 검색하고 알아가는 과정에서 정말 많이 배운 것 같다. 프로젝트 진행 중 가장 어려웠던 점은, 서버와 통신할 때 UTF-8 방식으로 진행하면 전송받는 String 앞에 Byte Order Mark가 추가되는데, 처음에는 이러한 내용을 몰라서 어려움이 있었다. 추후 해당 내용을 인지하고 이를 해결하기 위한 코드를 작성하였다.

임민정 : 처음 시작할 때는 어플리케이션을 만들어 본 적도 데이터베이스를 이용해 본 적도 없어서 막막했다. 어플 기능의 동작은 단순해 보이지만, 그것을 제대로 구현하기 위해서는 생각보다 시간과 머리를 많이 써야 했다. 특히 이미지를 firebase에서 불러오고 저장하는 것이 힘들었다. 이미지를 단순히 uri나 url을 안다고 해서 화면에 넣을 수 있는 게 아니어서 꽤 오래 헤맸다. Glide, picasso 등의 이미지로딩 라이브러리가 있다는 것을 알고, 이를 이용해 문제를 해결했다. 여러 기술을 알게 되고 발생하는 오류를 해결하면서 한 학기동안 그 어떤 수업보다 많은 것을 배웠던 수업이었다.