

# < 보드 in >

모바일 프로그래밍 응용 - 기말 프로젝트 최종보고서



과목명	모바일 프로그래밍 응용
담당교수	김지심 교수님
학과	컴퓨터공학과
학번/이름	2020081014 이경혜 2020081098 임예진
제출일	2022.06.19.

# 목차

## 1. 프로젝트 개요

- 개발목적 및 필요성
- 개발환경
- 변경 사항

## 2. 주요 기능

- 주요 기능 요약

## 3. 프로젝트 개발 결과 상세

- DB 구성
- 화면 구성 및 화면전환 계획서
- 앱 화면 캡쳐
- 구현 클래스 · XML · PHP 목록
- 주요 코드 설명

## 4. 최종 정리

- 개발 후기

# 1. 프로젝트 개요

- 개발목적 및 필요성
  - 보드게임은 오프라인, 온라인 그리고 다양한 플랫폼에서 쉽게 접함
  - 그러나 보드게임 정보 제공 모바일 앱은 없음
  - 목표 : 보드게임의 정보와 룰을 빠르게 찾아 시간의 낭비 없이 보드 게임을 선택할 수 있도록 도움을 주는 것
- 개발환경
  - Window System : Window 10
  - Tool Platform : Android Studio
  - Programming Language : Java
  - Data Base : XAMPP(Apache, MySQL)
- 변경 사항
  - 개발 범위 축소 : 리뷰, 등급, 보관함 등
  - 범위 조정으로 DB 재설계 : boardgame 테이블, users 테이블

## 2. 주요 기능

- 주요 기능 요약

- 회원가입

- 로그인

- 로그인 / 로그아웃

- 마이페이지

- 개인정보 변경(닉네임, 비밀번호 변경 지원)

- 보드게임 정보 검색

- 보드게임 이름 입력 시 관련된 보드게임 목록 나열

- 플레이 인원, 소요시간, 장르, 난이도 필터로 검색 지원

- 보드게임 추천

- 랜덤 보드게임 추천

### 3. 프로젝트 개발결과 상세

- DB 구성

The screenshot shows a database interface with two tables:

보드게임 정보	유저 정보
game_name : varchar(255)	userName : varchar(10)
youtube_link : varchar(255)	userEmail : varchar(255)
# genre_couple : tinyint(1)	# userPassword : varchar(20)
# genre_detective : tinyint(1)	
# genre_puzzle : tinyint(1)	
# genre_strategy : tinyint(1)	
# genre_coop : tinyint(1)	
# genre_card : tinyint(1)	
# genre_mafia : tinyint(1)	
# genre_economy : tinyint(1)	
# genre_lie : tinyint(1)	
# genre_handskill : tinyint(1)	
# genre_quickness : tinyint(1)	
# genre_trpg : tinyint(1)	
# min_player : int(11)	
# max_player : int(11)	
# difficulty : int(11)	
# play_time : int(11)	

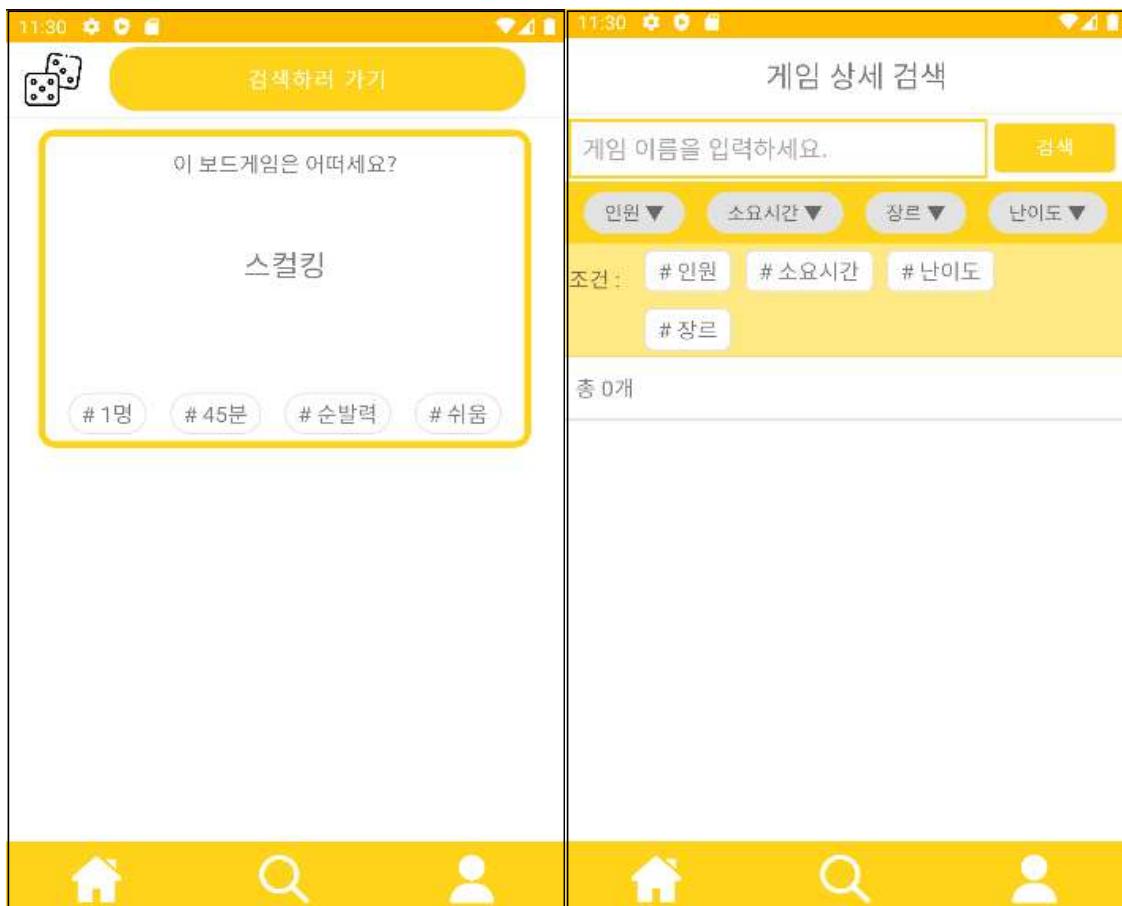
- 사용자 정보 : users 테이블
- 보드게임 정보 : boardgame 테이블

- 화면 구성 및 화면전환 계획서

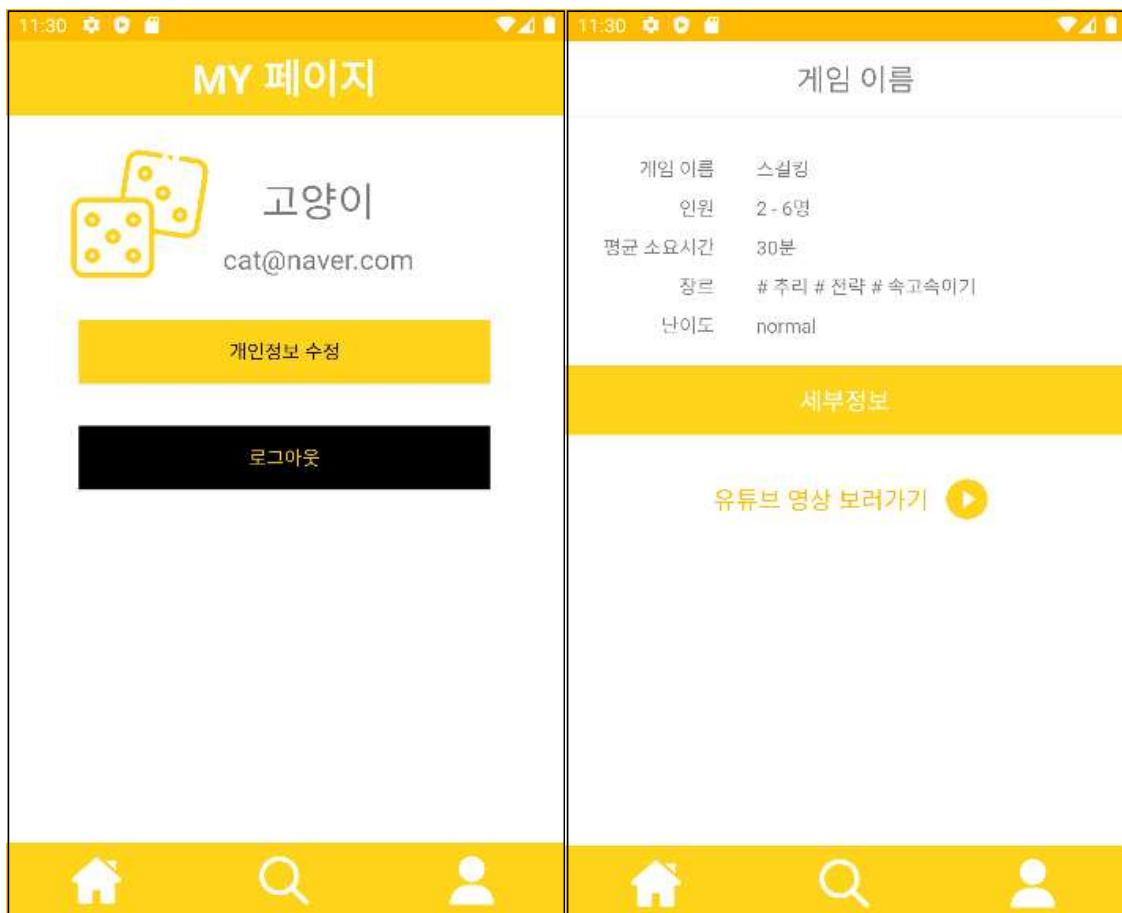
- 화면 구성

로그인	회원가입
<input type="text" value="이메일"/>	<input type="text" value="서명(닉네임)"/>
<input type="text" value="비밀번호"/>	<input type="text" value="이메일"/> <input type="button" value="중복확인"/>
<input type="button" value="로그인"/>	<input type="text" value="비밀번호"/>
<input type="button" value="회원가입"/>	<input type="checkbox"/> 이용약관 및 개인정보 수집에 동의합니다.
	<input type="button" value="회원가입"/>

( 로그인 & 회원가입 )



( 메인 & 게임 상세 검색 )



( 마이 페이지 & 게임 상세 )

### - 화면전환 계획서

NO.	화면	화면전환
1	로그인 화면	2, 5
2	회원가입 화면	1, 3, 4
3	개인 정보 취급 방침 화면	2
4	이용 약관 화면	2
5	메인화면	6, 8
6	검색화면	5, 7
7	보드게임 상세화면	5, 6
8	마이페이지	1, 9
9	개인정보 수정	8

- 구현 클래스 · XML · PHP 목록

Class	주요설명
LoginActivity	로그인 화면 기능 구현
LoginRequest	아이디, 비밀번호 데이터를 서버에 전달하는 요청자 클래스
RegisterActivity	회원가입 화면기능 구현
RegisterRequest	회원가입에 필요한 사용자 데이터(닉네임, 이메일, 비밀번호)를 서버에 전달하는 요청자 클래스
ValidateRequest	이메일 중복체크 기능 구현 이메일을 서버에 전달하는 요청자 클래스
MyinfActivity	개인정보 수정 기능 구현
PersonalInformationCollectionActivity	개인정보 취급방침 스크롤 및 뒤로가기 버튼 구현
TermsAndConditionsActivity	이용약관 스크롤 및 뒤로가기 버튼 구현
UserdataRequest	이메일(유저의 키값)을 서버에 전달하는 요청자 클래스
UserNameChangeRequest	이메일(유저의 키값)과 새로운 닉네임을 서버에 전달하는 요청자 클래스
UserNamePwChangeRequest	이메일(유저의 키값)과 새로운 닉네임/비밀번호를 서버에 전달하는 요청자 클래스
MypageFragment	マイ페이지 기능 구현
CustomAdapter	리스트뷰에 넣을 커스텀 어댑터 클래스
FragViewModel	ViewModel을 상속받는 FragViewModel 클래스
GameData	데이터 전달용 클래스(DTO) - 커스텀 어댑터 및 게임 정보 필요 영역에서 사용
GameInfoFragment	게임 세부정보 프래그먼트 처리
HomeFragment	홈(기본) 프래그먼트 처리
MainActivity	Fragment를 담을 inLayout과 하단 네비게이션 바의 탭을 처리
SearchFragment	게임 검색 프래그먼트 처리
SearchRequest	게임 검색용 리퀘스트 정의

Layout	주요설명
activity_login.xml	로그인 화면
activity_register.xml	회원가입 화면
myinf.xml	개인 정보 수정 화면
fragment_mypage.xml	마이페이지
personal_information_collection.xml	개인 정보 취급 방침 화면
terms_and_conditions.xml	이용 약관 화면
activity_main.xml	메인화면
adpater_listview.xml	리스트뷰에 들어갈 어댑터 요소
dialog_difficulty.xml	난이도 다이얼로그 화면
dialog_time.xml	소요시간 다이얼로그 화면
fragment_game_info.xml	게임정보화면
fragment_home.xml	메인화면
fragment_search.xml	검색 화면

PHP	주요설명
boin_UserLogin.php	로그인을 위해 아이디/비밀번호를 받아와 조회문의 결과와 비교하여 처리 결과를 리턴
boin_UserRegister.php	회원 정보(닉네임, 이메일, 비밀번호)를 받아와서 DB에 저장 후 처리 결과를 리턴
boin_UserValidate.php	중복체크를 위해 이메일을 받아와 조회문의 결과와 비교한 후 처리 결과를 리턴
boin_UserData.php	이메일을 받아와 조회문을 통해 나온 닉네임과 비밀번호를 리턴
boin_UserNameChange.php	이메일과 새로운 닉네임을 받아와 이메일을 통해 나온 데이터의 닉네임에 새로운 닉네임으로 수정 후 결과를 리턴
boin_UserNamePwChange.php	이메일과 새로운 닉네임/비밀번호를 받아와 이메일을 통해 나온 데이터의 닉네임/비밀번호에 새로운 닉네임/비밀번호로 수정 후 결과를 리턴
boin_ForRecommend.php	보드게임 테이블에서 랜덤한 레코드 하나를 리턴
boin_ForSearch.php	검색을 위해 설정한 조건을 받아 그에 맞는 SELECT문 실행 결과를 JSONArray로 리턴

## • 주요 코드 설명

- FragViewModel : extends ViewModel

SearchFragment에서 GameInfoFragment로 GameData형의 값을 전달하여 게임 세부 정보 화면에 값이 적용될 수 있도록 ViewModel을 상속받은 FragViewModel을 사용

( SearchFragment.java )

```
// 리스트뷰의 아이템 클릭 시 그 아이템에 해당하는 게임의 상세 정보 프래그먼트로 전환
lv.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        fragViewModel.setLiveData(gameDataList.get(position));

        ((MainActivity) getActivity()).replaceFragment(GameInfoFragment.newInstance());
    }
});
```

```
//FragViewModel 객체 생성
@Override
public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {
    super.onViewCreated(view, savedInstanceState);

    fragViewModel = new ViewModelProvider(requireActivity()).get(FragViewModel.class);
}
```

( GameInfoFragment.java )

```
@Override
public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {
    super.onViewCreated(view, savedInstanceState);
    //Android 앱에서 구성요소 : 뷰-모델로 생명 주기를 고려하여 UI 관련 데이터를 저장하고 관리
    //프래그먼트 간의 데이터 전달을 위해 생성 후 변화 시 값을 받아옴
    fragViewModel = new ViewModelProvider(requireActivity()).get(FragViewModel.class);
    fragViewModel.getLiveData().observe(getViewLifecycleOwner(), new Observer<GameData>() {
        @Override
        public void onChanged(GameData gameData) {
            txtGameName.setText(gameData.getName());
            txtGamePlayer.setText(gameData.getPlayer());
            txtGameTime.setText(gameData.getTime());

            //해당하는 장르를 모두 꺼내서 # a # b ... 형식으로 교체
            String str = "";
            for(String s:gameData.getGenre()) {
                str += "# " + s + " ";
            }
            txtGameGenre.setText(str);
            txtGameDifficulty.setText(gameData.getDifficulty());

            uri = Uri.parse(gameData.getYoutube());
        }
    });
}
```

# 4. 최종 정리

## • 개발 후기

- 2020081014 이경혜

프래그먼트와 탭을 이용한 화면 전환을 처음 구현해봐서 context나 데이터 전달하는 부분 등에 대해서 많은 어려움을 겪었다. 처음에 프래그먼트를 넣을 레이아웃으로 constraint layout을 설정하였더니 프래그먼트 화면의 내용들이 왼쪽으로 정렬된 것처럼 붙어서 나오는 문제가 발생하였다. Backgorund 값을 통해 constraint layout의 layout\_width와 layout\_height가 화면에 들어맞지 않는 것이 문제라는 것을 발견했다. 이에 관하여 찾아보다가 프래그먼트는 특정 레이아웃에 넣어야 제대로 적용된다는 사실을 알게 되었고 기존의 프래그먼트를 담는 데 사용되었던 FrameLayout을 상속받은 FragmentContainerView로 레이아웃을 변경하였더니 제대로 적용되었다. 인텐트, 번들, 뷰모델 방식, 레인지 슬라이더 등 새로운 요소를 접하고 익히는 좋은 기회가 되었던 것 같았다. 하지만 php로 json 오브젝트를 받아오는 과정에서는 아직 공부가 더 필요한 것 같다.

- 2020081098 임예진

보드in 개발 과정에서 가장 어려웠던 점은 php의 사용이다.

php 사용과정에서 mysqli\_stmt\_bind\_param과 mysqli\_stmt\_execute를 통해 매개변수 값을 전달하는 부분에서 오류가 많이 발생했다.

그래서 다음과 같이 "UPDATE users SET userName = ".\$NewName.", userPassword= ".\$userNewPassword." WHERE userEmail= ".\$userEmail.""

php에서 문자열을 더해주는()을 사용해서 sql문에 변수를 직접 넣어 해결했다.

또한 테이블을 검색하여 원하는 값을 받아오기 위해서는 mysqli\_prepare로 해결이 불가능했다.

이는 mysqli\_query를 통해 sql문을 실행하고 mysqli\_fetch\_array로 값을 받아와 해결 할 수 있었다.

보드in 개발과정을 통해 반성한점은 기획단계에서 실현 가능한 범위를 정해야했다는 것이다. 기획자체는 좋았으나 개발 가능한 실력 수준과 기간을 고려하지 못해 기획대로 모든 부분을 개발하지 못했다.

현재 개발된 보드in은 로그인기능이 존재하나 로그인을 통해 사용자가 추가로 얻을 수 있는 이점이 없다.

첫 기획 대로라면 로그인한 사용자는 게임 보관함을 통해 좋아하는 게임을 따로 볼 수 있으며, 선호도 검사를 통해 메인화면에서

선호도에 따른 추천정보를 받아볼 수 있다. 또한 게임에 리뷰를 달아 게임을 평가할 수 있었다.

이런 기능은 핵심 기능이라 볼 수 있는 게임 검색기능을 만을 남기고 사라졌다.

향후에는 미처 개발하지 못한 범위 뿐만 아니라 보드게임의 도움이 될 수 있는 팀나누기, 주사위 등을 추가로 구현하고자 한다.