

## PERSONAL

서민구

안드로이드 신입 개발자

홍익대학교(서울) 영어영문학과

서울시 송파구 문정동

010. 5673. 0703

[tjalsrnek@naver.com](mailto:tjalsrnek@naver.com)

<https://github.com/Gooreum>

## SKILLS

언어

Java, PHP, JavaScript

운영체제

Android, Linux(Ubuntu)

웹서버

Apache, Nginx

데이터베이스

MySQL, MariaDB, SQLite

프로토콜

HTTP, TCP/IP

## PROJECTS

(포트폴리오에 작성된 프로젝트 순서는 최근에 만든 프로젝트 순서입니다.)

당근마켓 클론 (개인프로젝트)

Y&I (개인프로젝트)

ShareProject (개인프로젝트)

---

## 소개

당근마켓은 자신의 동네를 중심으로 한 중고거래 플랫폼입니다. 월 사용자 수가 280만명에 이르지만 오프라인 거래를 지향하도록 하여 중고거래 사기를 최소화 시키고 있습니다. 또한 동네 상인들의 광고를 유도하여 '지역 경제'를 활성화 시킬 수 있는 구체적인 대안을 마련하였다는 점에서 당근마켓은 매력적인 서비스입니다. 이러한 당근마켓을 클론 해보면서 실제 성공적인 서비스의 개발 로직을 공부해볼 수 있었습니다.

## 사용기술

OS Android, Linux(Ubuntu)

사용언어 Java, PHP, JavaScript

프로토콜 HTTP

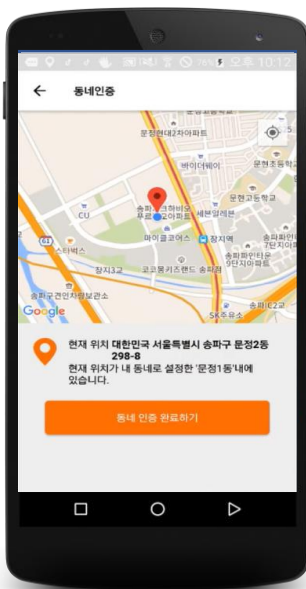
데이터베이스 MariaDB

서버 호스팅 AWS EC2

웹서버 Nginx

채팅서버 Node.js

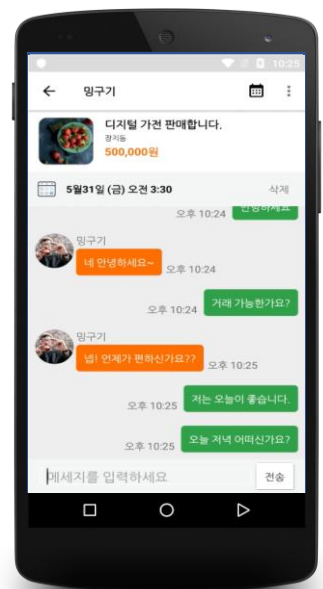
기타 Retrofit2, Facebook Account Kit, Butterknife, Glide, Google Map, Socket.io



[동네인증 화면]



[전체 게시물 화면]



[채팅 화면]

## 문제해결

### MVP 디자인패턴 적용

프로젝트를 진행하면서 코드의 양이 많아져 **유지보수**의 어려움을 느꼈습니다.

코드를 규칙없이 작성하였기 때문이라 판단하였고, 따라서 MVP 디자인패턴을 적용하여 시간이 지나도 코드의 구조를 쉽게 파악할 수 있도록 수정하였습니다.

### [홈화면 View]

```
public interface HomeView {  
  
    void showProgress();  
  
    void hideProgress();  
  
    void onErrorLoading(String message);  
  
    void onGetResult(List<Product> products);  
  
    void onGetRefreshResult(List<Product> products);  
  
    void moveActivity(Class activity);  
  
}
```

화면에 보일 View를 Interface로 구현함으로써 View의 역할을 명시적으로 정의하였습니다. 따라서 시간이 지나 홈화면의 기능을 수정하거나 개선 할 경우 **홈화면의 역할**이 무엇인지를 **빠르게 파악**할 수 있었습니다.

## [홈화면 Presenter]

```
public class HomePresenter {  
    private HomeView view;  
    SessionManager sessionManager;  
  
    public HomePresenter(HomeView view, Context context) {  
        this.view = view;  
        this.sessionManager = new SessionManager(context);  
    }  
  
    void getProducts(String nick, String city, String gu, String dong) {  
        view.showProgress();  
  
        ApiInterface apiInterface = ApiClient.getApiLocation().create(ApiInterface.class);  
        Call<List<Product>> call = apiInterface.getProduct(nick, city, gu, dong);  
  
        call.enqueue(new Callback<List<Product>>() {  
            @Override  
            public void onResponse(@NonNull Call<List<Product>> call, @NonNull Response<List<Product>> response) {  
                view.hideProgress();  
                if (response.isSuccessful() && response.body() != null) {  
                    view.onGetResult(response.body());  
                }  
            }  
  
            @Override  
            public void onFailure(@NonNull Call<List<Product>> call, @NonNull Throwable t) {  
                view.hideProgress();  
                view.onErrorLoading(t.getLocalizedMessage());  
            }  
        });  
  
        void getProductsFromDetail(String nick, String city, String gu, String dong) {...}  
        void getProductsFromDetailNotLogin() {...}  
        void getProducts() {...}  
  
        void nextActivityIsLogin(Context context, Class activity) {...}  
        void nextActivityWithoutLogin(Class activity) {...}  
        void showDialog(Context context) {...}  
    }  
}
```

Presenter는 View에 값을 전달해주거나, View를 통해 전달 받은 값을 Model에 전달하여 값을 갱신하게 됩니다.

View에 필요한 역할을 명시한 후 Presenter 클래스에서 필요한 메서드를 구현하였습니다.  
따라서 Activity/Fragment 안에서 단순히 주먹구구식의 Method를 작성하는 이전의  
개발 프로세스를 개선하였습니다.

이러한 프로세스의 개선은 홈화면 코드를 수정할 경우 기존에 작성된 코드를 이해하는 시간을  
확실히 단축시켰고 보다 빠른 유지보수를 가능하게 만들었습니다.

## 링크

GitHub : <https://github.com/Gooreum/CarrotMarket2>

개발일지 : [https://drive.google.com/open?id=1vG0H1qaqWm\\_amafujbQSn8Cr6SI1v\\_-4](https://drive.google.com/open?id=1vG0H1qaqWm_amafujbQSn8Cr6SI1v_-4)

## 소개

SNS에서 함께 프로젝트를 만들어 나갈 사람들을 모집할 수 있습니다.

Y&I는 다음 장에 나올 'ShareProject' 기획의 연장선이며, 처음으로 안드로이드 Client-Server간 HTTP 통신과 TCP/IP 소켓 통신을 통한 채팅을 구현했습니다.

## 사용기술

OS Android, Linux(Ubuntu)

사용언어 Java, PHP, JavaScript

프로토콜 HTTP, TCP/IP

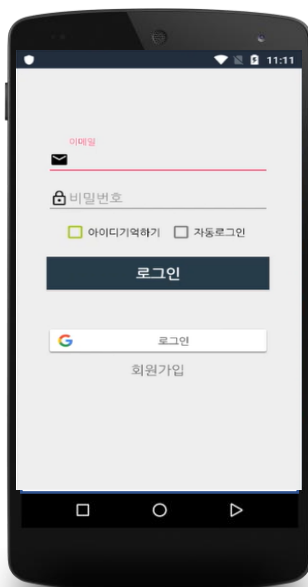
데이터베이스 MySQL, SQLite

서버 호스팅 AWS EC2

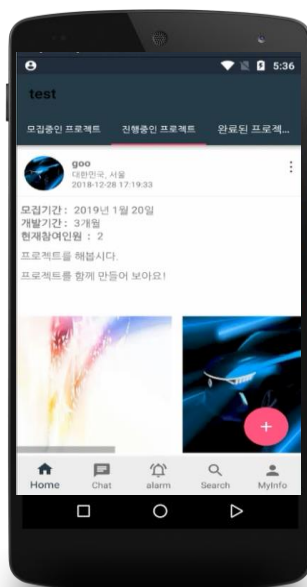
웹서버 Apache

채팅서버 Java

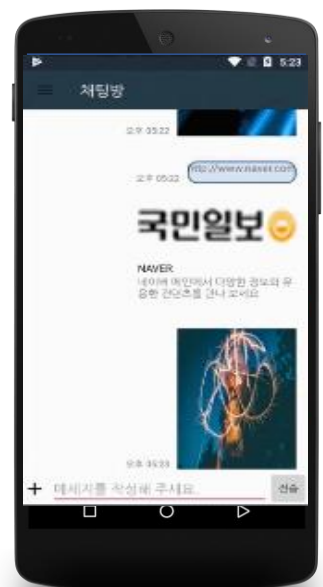
기타 Volley, Picasso, Google Login,



[인트로 화면]



[전체 게시글 화면]



[채팅 화면]

## 문제해결

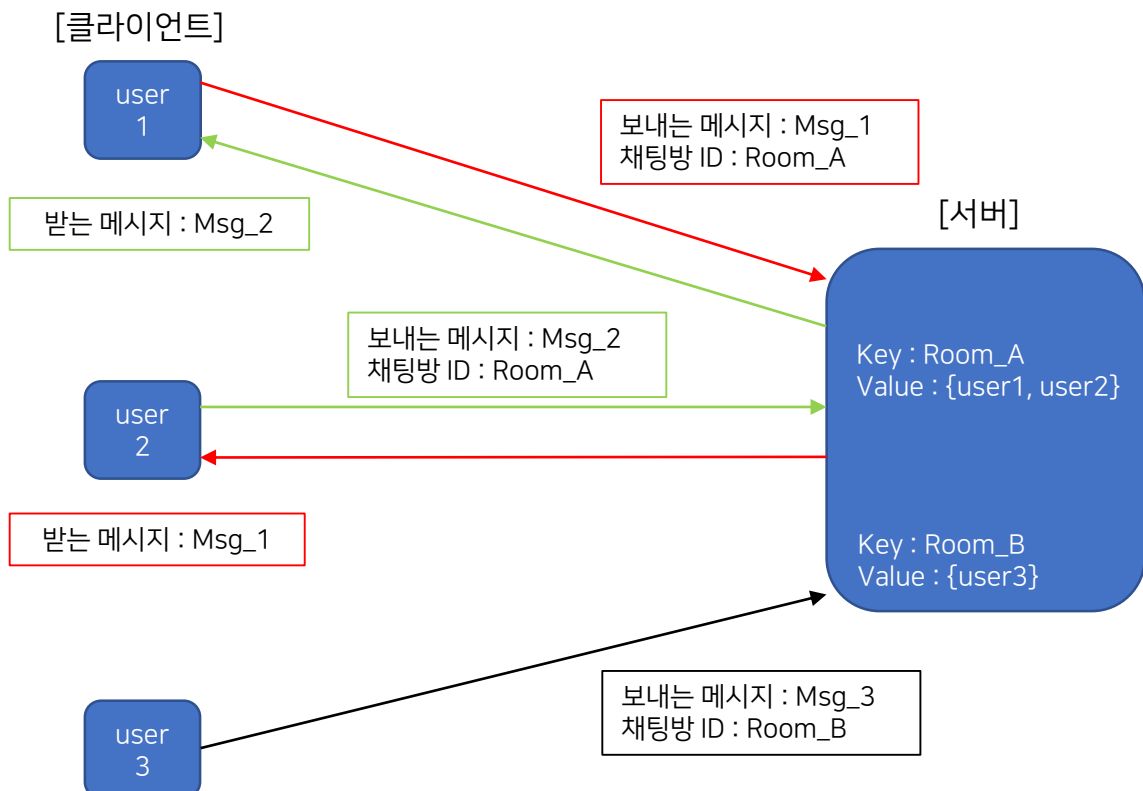
### 채팅방 나누기

TCP/IP로 소켓통신을 할 때, 채팅방을 나누어 각 방에 속한 인원에게만 메시지를 보내기 위해 다음과 같이 해결했습니다.

1. 클라이언트에서 메시지를 서버에 보낼 때, 채팅방 ID를 메시지 정보에 추가해서 전송
2. 서버는 HashMap(Key : 채팅방 ID, Value : list[접속한 유저])을 이용하여 클라이언트에서 받은 메시지 값을 받아 각 채팅방에 접속해 있는 유저에게 메시지를 전송

### 채팅 메시지 저장하기

1. 많은 사용자가 채팅 메시지를 서버에 요청하게 되면 서버에 부담이 생길 것이라 생각했습니다. 따라서 채팅 메시지를 안드로이드 로컬 저장소인 SQLite에 저장하였습니다.
2. '내가 보낸 메시지'는 서버에서 다시 보내지 않고 바로 SQLite에 저장하여 Adapter에 값을 전달하도록 하였습니다.



## 링크

GitHub : [https://github.com/Gooreum/Y\\_I](https://github.com/Gooreum/Y_I)



## 소개

첫 안드로이드 작품입니다.

SNS에서 함께 프로젝트를 만들어 나갈 사람들을 모집할 수 있습니다.

Github, StackOverflow와 같은 오픈소스 문화에 매료되어 개발자가 되고 싶은 마음을 가지게 되었습니다. '내 생각과 코드를 나누면 보다 더 좋은 사회를 만들 수 있다' 라는 오픈소스 철학을 제 나름 실현하기 위해 기획하고 개발하였습니다.

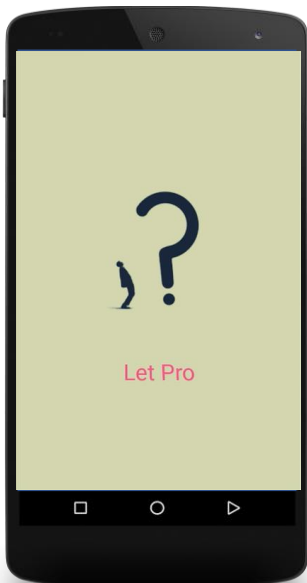
## 사용기술

OS Android

사용언어 Java

저장소 SharedPreferences

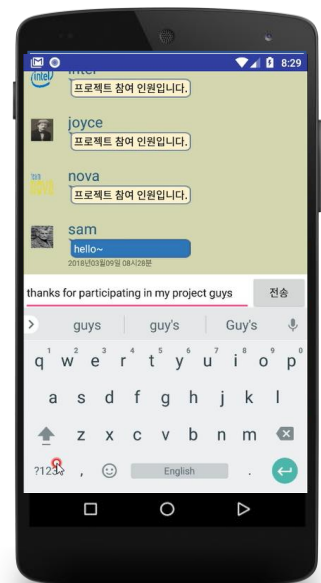
기타 Gson



[인트로 화면]



[전체 게시글 화면]



[채팅 화면]

## 문제해결

### SharedPreferences에 사용자 게시글 저장하기

각 유저별 게시글 정보를 ArrayList에 담았고, ArrayList를 SharedPreferences에 Json형식으로 데이터를 저장할 수 있도록 해주는 Gson 라이브러리를 사용하였습니다.

## 링크

GitHub : <https://github.com/Gooreum/ShareProject>