

지속 가능한 성장을 추구하는 안드로이드 개발자



공채운 Chaeun Gong

- ✉ knmy0101@gamil.com
- ☎ 010-9974-1449
- 🌐 <https://github.com/Gongcu>
- 📖 <https://bb-library.tistory.com/>

Introduction

안녕하세요 안드로이드 주니어 개발자 공채운입니다. 2018년 부터 안드로이드 개발을 시작하였고, 안드로이드를 활용한 다양한 개인, 팀 프로젝트 경험이 있습니다. 안드로이드 공식 사이트를 보며 필요한 기술을 사용하며 개발하는데 큰 문제가 없고, 개발 과정에서 발견한 해결하기 어렵고 잘 알려지지 않은 버그는 Stack Overflow에 직접 질문을 작성하여 해결하기도 합니다. 새로운 기술을 습득하는 것에 대해 관심이 많고, 이전에 진행한 프로젝트를 다시 보며 개선을 하기도 합니다. 또한, 학업에도 성실히 임하여 좋은 성적을 유지하고 있습니다.

Skills

업무에도 활용해요. 6

✓ Android

✓ Kotlin

✓ Java

✓ Express.js

✓ MySQL

⚙ Git

어느 정도 사용할 줄 알아요. 5

✓ Toast Cloud

✓ C++

✓ Socket.IO

✓ Javascript

✓ MongoDB

보고 이해할 수는 있어요. 1

✓ C

Prize

- 단국대학교 SW 경진대회 동상(2019.11)

Certificate

- TOEIC 830

HEALTH-ER

2019.06 ~ 2019.08

프로젝트 요약

- 안드로이드 기반의 웨이트 트레이닝 가이드 애플리케이션
- 구글 플레이 스토어에 등록되었으며, 다운로드 수 400에 20명 평균 평점 4.35를 기록
- 개인 프로젝트로 Java를 주로 사용하였고, SQLite를 활용
- 운동 정보, 알람 기능, 운동 일지 기록 등의 기능을 제공
- 최근 다시 프로젝트를 Kotlin, Jetpack을 활용하여 리팩토링

사용한 언어/프레임워크/라이브러리/구조

Java, Kotlin, Room, Data binding, LiveData

프로젝트를 통해 느낀 점

- 첫 애플리케이션 개발이었고 안드로이드 개발자의 길로 접어들게 된 계기
- 처음부터 끝까지 스스로 개발하고 배포하여 많은 경험을 할 수 있었다. 특히, 소프트웨어 개발 과정에는 실제 설계, 구현 과정뿐만 아니라 유지 보수에 더 많은 작업이 들어가는 것을 깨닫는 계기.
- 실제 개발에는 2개월이 소요됐지만 버그 픽스, 디자인 수정과 같은 유지보수 기간이 2개월이 넘게 소요되었다. 따라서, 유지보수하기 쉬운 코드를 작성해야 하는게 중요하다는 것을 깨닫는 계기
- 유저의 리뷰를 반영하여 더 좋은 기능을 제공하기 위해 노력
- 개인 개발자 입장에서 저작권에 자유로운 운동 관련 이미지를 찾는게 어려움을 느낌. 다른 분야의 이미지도 비슷할 거라 생각

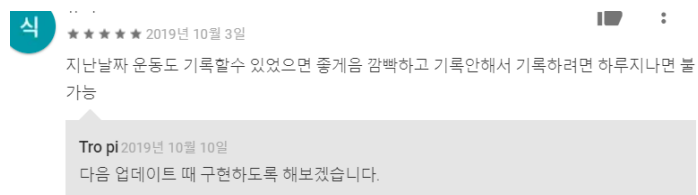


Figure 1. 유저의 리뷰를 참고하여 수정

Github: <https://github.com/Gongcu/HEALTH-ER-AD>

2차 사고 방지 애플리케이션 2020.03 ~ 2020.07

프로젝트 요약

- 고속도로에서 차량 2차사고 방지 기능 제공하는 안드로이드 애플리케이션
- 해당 애플리케이션을 실행 중인 고속도로를 주행하는 유저의 차량이 급감속, 충돌 등의 사건이 발생할 시 주변 사고 연관 유저들에게 PUSH 알림을 전송
- 가속도 센서, 위치 정보 활용
- 4인 구성 팀 프로젝트로 진행되었으며 개발의 90% 정도를 담당
- 팀 프로젝트가 종료된 뒤 개인적으로 코드 리팩토링을 진행하여 Java 코드를 Kotlin으로 변환하고, Retrofit, Data binding 등을 활용
- 서버의 경우 Express 프레임워크를 활용하여 Toast Cloud에서 구동

사용한 언어/프레임워크/라이브러리/구조

Kotlin, Room, Data binding, Retrofit, Express.js, Toast Cloud, 카카오 맵 API, Firebase Auth, FCM(Firebase Cloud Messaging)

프로젝트를 통해 느낀 점

- 과거 사이드 프로젝트에서 Firebase를 사용했으나 Firebase의 성능, 각종 제약에서 불편함을 많이 겪음.
- 해당 프로젝트에서 Express를 통해 직접 웹서버를 구축하면서 해당 제약이 없어지는 경험을 통해 개발에 더 흥미를 느낌
- 첫 팀 프로젝트로 진행 과정에서 많은 의견 충돌을 해결하는 과정, 협업하는 과정을 배우는 경험
- 처음으로 팀 단위로 github를 사용하여 Branch, Pull Request에 대한 기본적인 활용법을 습득
- 클라우드 서비스 사용 경험

Github: <https://github.com/Gongcu/2ndAccidentPrevention>

단국대 동아리 관리
애플리케이션
2020.09 ~ 진행 중

프로젝트 요약

- 안드로이드 기반의 단국대학교 동아리 관리 애플리케이션
- 코로나로 인한 언택트 시대로 줄어든 동아리 활동을 모바일에서 지원하여 시대적 위기를 기회로 전환하고자 함
- 기획자, 디자이너, 리액트 네이티브 개발자, 서버 개발자로 구성된 팀 프로젝트이며 Express를 활용하여 REST API 서버 구축 담당
- 리액트 네이티브에서 구현하기 까다로운 부분은 안드로이드 네이티브 모듈을 제작 중

사용한 언어/프레임워크/라이브러리/구조

Express, Socket.IO, MySQL, MongoDB, Toast Cloud, Postman, Swagger

프로젝트를 통해 느낀 점

- 기획자, 디자이너, 개발자로 역할 분담이 되어 진행된 프로젝트로 기획자, 디자이너의 관점이 개발자의 관점과 다르기에 서로를 이해하고 소통해야 함을 경험 중
- 백엔드에 대한 기본적인 역량이 생겼고, 주력인 안드로이드 클라이언트 측에서도 백엔드가 어떻게 돌아가는지 더 자세히 알 수 있는 기회
- 요구사항 정의 - 분석 - 설계 - 구현/테스트 - 유지보수의 SW 개발 프로세스를 채택 중이며 현재는 구현/테스트 단계

Github: <https://github.com/Gongcu/REST-API-SERVER>

MVVM 구조
학습 사이드 프로젝트
2020.07

프로젝트 요약

- 해당 프로젝트는 MVVM 구조와 Jetpack을 학습하기 위한 사이드 프로젝트
- 영화 API를 활용하여 영화 목록을 가져오고, 보고싶은 영화 목록을 추가/삭제/편집/정렬하는 기능을 제공한다.

사용한 언어/프레임워크/라이브러리/구조

Kotlin, Room, Data binding, ViewModel, Retrofit, Glide, ListAdapter

프로젝트를 통해 느낀 점

- HEALTH-ER 개발 경험에서 유지보수하기 쉬운 코드를 작성해야 함의 중요성을 깨닫고 시작
- 초기에는 코드가 더 복잡해져 유지보수하기가 더 어려워지는게 아닌가 싶었지만 이러한 구조를 통해 관심사를 분리하고, 클래스간 결합도가 느슨해지며 일명 '모듈화'가 된다는 것을 학습하게 됨
- 다른 개발 분야에도 활용되는 아키텍처 패턴의 기초를 학습하는 계기

Github: <https://github.com/Gongcu/MVVMExample>

Socket.IO를 활용한 실시간 채팅 사이드 프로젝트 2020.10

프로젝트 요약

- Socket.IO를 이용한 안드로이드 애플리케이션 사이드 프로젝트
- 채팅 내역은 MySQL DB에 기록
- 소켓 서버는 Express로 구현

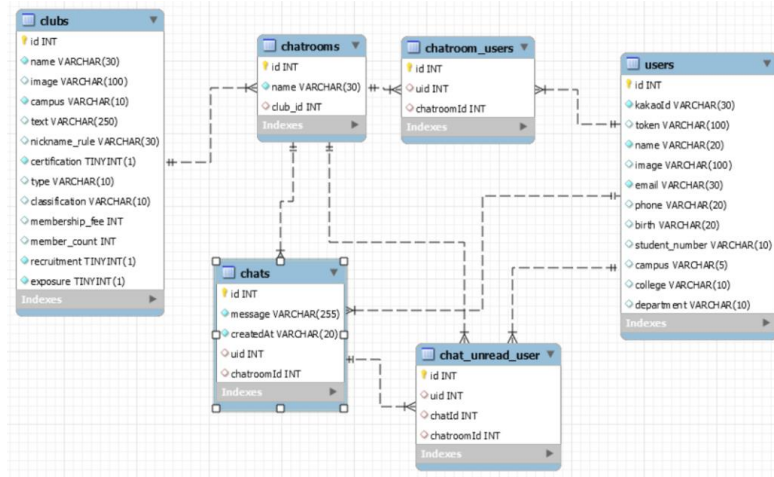


Figure 2. 채팅 관련 E-R 다이어그램

사용한 언어/프레임워크/라이브러리/구조

Kotlin, Socket.IO, Glide, 나인패치

프로젝트를 통해 느낀 점

- 기존의 클라이언트에서 단방향 요청-응답 방식이 아닌 양방향인 소켓 통신에 대한 기본적인 개념 학습
- 나인패치를 활용하여 UI 개선한 경험

Github: <https://github.com/Gongcu/SocketExample>

자기소개

지속 가능한 성장을 추구하는 주니어 개발자 공채운 입니다. 고등학교 때 삼성 소프트웨어 아카데미를 통해 아두이노로 처음 개발을 처음으로 접하게 되었습니다. 그때는 수업을 하나도 따라가지 못해서 개발에 대한 흥미가 별로 없었습니다. 이후, 단국대학교 소프트웨어학과에 입학하며 1학기에 처음으로 C언어를 배우게 되었는데 이때도 역시 코딩을 잘 하지 못했습니다. 1학년 전공인 '프로그래밍1' 과목에서 B 학점을 맞는 그냥 남들과 비슷한 수준의 학생이었습니다.

그런 제 주변에는 안드로이드 애플리케이션 개발을 잘 하는 동기가 있었습니다. 스스로 눈에 보이는 무언가를 만드는 이 친구를 보고 많은 동기 부여가 되어, 프로그래밍 공부에 전념하기 시작했습니다. 1학년 첫 여름방학 때 '열혈강의 C 포인터' 책을 구입하여 매일 적어도 1~2시간 씩 코딩 공부를 진행했고, 실제 다음 학기 '프로그래밍2' 과목에서 A를 받게 되었습니다. 이때부터 프로그래밍에 대한 자신감이 생기기 시작했고, 전공에 더 열중하게 되어 지금까지 이 두 과목을 제외한 모든 전공 과목은 A+라는 좋은 학점을 받았습니다.

2학년 1학기에는 Java를 학습하게 되었고, 이후 Java를 통해 안드로이드 개발을 할 수 있음을 알고 안드로이드 개발을 시작하였고, 이때 'HEALTH-ER'를 개발하고자 하는 목표가 생겼습니다. 취미가 운동이었고 플레이 스토어에 존재하는 기존 애플리케이션은 마음에 들지 않아 직접 운동하는 사람의 입장에서 필요한 기능을 정의하고 개발했습니다. 처음으로 무언가를 무작정 개발하고 배포하는 경험을 하게 되었습니다. 하지만, 실제 배포 후 개발 단계에서 발견하지 못했던 버그가 구글 콘솔에 굉장히 많이 등록되었고, 이런 버그 수정 과정과 유저 리뷰를 통해 앱을 수정하는데 개발에서 배포까지의 시간보다 훨씬 많은 시간이 들었습니다. 'HEALTH-ER'는 처음 개발한 앱이다 보니 그냥 기능이 '동작'하는 부분에 초점을 맞추었습니다. 따라서, 코드를 수정할 때 가독성도 매우 떨어지고 보일러 플레이트 코드도 굉장히 많으며 수정하기도 매우 어려웠다. 이때, 유지 보수가 쉬운 코드가 작성하는 게 중요하다는 것을 처음으로 깨달았습니다.

이후 좀 더 '좋은 코드'를 작성하기 위해 노력했습니다. 변수명, 클래스명을 짓는데 조금 더 생각을 하며 시간을 쓰고, 객체 지향, MVVM 구조를 학습하였습니다. 이러한 학습 과정을 통해 관심사의 분리와 의존성을 줄이는 코드를 작성하는게 유지 보수가 쉽다는 것을 알게 되었습니다. 한편으론, 이런 코드 작성 방법이 너무 난해하여 어려움이 많고 지금도 많은 학습이 필요하다는 것을 느껴 지금도 관련 학습에 열중하고 있습니다.

지금까지의 경험을 비추어 보면 실제 경험을 통해 무언가에 대한 필요성을 느끼는 경우에 더 적극적으로 해당 개념에 대해 학습하게 되었습니다. 그리고 특정 개념을 학습하면 이후에 또 다른 개념에 대한 학습의 필요성을 느끼는 과정이 반복되었습니다. 결국 계속해서 학습해야 하고 이러한 과정은 마라톤과 같다고 생각합니다. 따라서, 오버페이스 하지 않으며 짧은 시간이라도 꾸준히 개발 공부를 하려고 노력하는 편입니다. 새로 학습한 내용이나 디버깅, 리팩토링하는 과정에서 학습한 개념은 블로그에 간단히 기록하기도 하고, Medium에서 매일 전송하는 안드로이드 관련 영문 포스트를 못해도 하루에 한 개는 읽고 있습니다. 이 같은 과정을 통해 지속 가능한 성장을 추구하고 실천하여 좋은 개발자가 되는 게 인생의 목표입니다.

해당 페이지까지 제 문서를 읽어주셔서 감사합니다.

더 궁금한 점이 있으시면 아래 메일로 연락 주세요.

knmy0101@gmail.com