

Personal Project Report

- Mobile Application Lab 2022 -

Student ID : 2020314994

Student Name : 황수경

어플리케이션 이름
KINGO ACADEMY
어플리케이션 설명
<p>'KINGO ACADMEY'는 학원에서 학생과 선생님들이 필요한 기능을 담은 어플리케이션이다.</p> <p>KINGO ACADEMY 학원에 등록하면 학생과 선생님은 각각 고유번호를 받는데 이는 자신만 아는 비밀번호와 같은 기능을 한다. 단, 선생님은 학생의 고유번호를 알고 있다. 모든 사용자들은 이 고유번호에 따라 구분된다. 계정이 기존에 존재하지 않는다면 회원가입을 한 후에 이름과 고유번호를 입력하여 로그인을 하면 학생과 선생님은 각각의 구분된 페이지로 이동하게 된다.</p> <p>학생은 금주 시험 성적을 조회하고, 학원의 연락처를 조회하여 연락을 걸 수 있다.</p> <p>선생님은 학생의 고유번호를 입력해 학생에게 금주 시험 성적을 입력, 수정하거나 본인의 연락 정보를 수정할 수 있다.</p>
액티비티 설명
<p>① Main Activity : 사용자의 이름, 고유번호를 받아 로그인, 회원가입을 실행한다. 로그인은 DB에 저장된 (이름, 고유번호)의 쌍이 일치할 경우 성공이며 회원가입은 DB에 해당 고유번호가 존재하지 않을 때 (이름, 고유번호)를 DB에 입력하여 계정을 생성한다. 선생님의 고유번호는 2로 시작하는데, 선생님의 경우 회원가입을 허용하지 않는다. aws 람다 플러스 서버에 POST로 /login, /adduser request를 보내 response json이 {"success" : true}일 때 환영 toast와 함께 다음 액티비티로 넘어가고 그렇지 않을 경우 틀렸다는 toast를 띄운다.</p> <p>② Student Page Activity : 사용자가 학생일 경우 사용할 "마이페이지"를 보여준다. 학생은 이 페이지에서 "성적조회하기", "연락처확인" 버튼을 통해 다음 액티비티로 넘어갈 수 있다. Main Activity에서 intent로 학생의 이름과 고유번호를 전달받아 이름은 상단에 표기하며 고유번호는 성적을 조회하는 View Grade Activity로 넘겨준다.</p> <p>③ View Grade Activity : 선생님이 입력한 금주의 성적을 조회할 수 있다. POST로 /gradeview request를 보내 자신의 성적을 response로 받아 listview에 업로드한다. 성적이 업로드 되지 않은 경우 조회할 성적이 없다는 문구를 띄운다.</p>

- ④ Teacher Number Activity : 전화번호, 이메일을 각각 listview로 보여준다. 선생님이 자신의 연락처를 수정할 수 있어 POST /contactview로 값을 받아 listview를 업데이트하여 보여주며 해당 연락처 옆 "전화연결", "메일작성" 버튼으로 통화, 메일 어플로 넘어갈 수 있다.
- ⑤ Teacher Page Activity : 사용자가 선생님일 경우 "마이클래스"를 보여준다. 학생은 이 페이지에서 "성적조회하기", "연락처확인" 버튼을 통해 다음 액티비티로 넘어갈 수 있다. Main Activity에서 intent로 학생의 이름과 고유번호를 전달받아 이름은 상단에 표기하며 고유번호는 성적을 조회하는 View Grade Activity로 넘겨준다.
- ⑥ Grade Upload Activity : 학생의 고유번호와 성적을 입력하여 DB에 POST /gradeupdate request로 입력하고 response로 success=true, false를 받는다. 성적 입력을 위해서는 고유번호가 존재해야 한다.
- ⑦ Update List Activity : Teacher Page Activity로부터 고유번호를 intent로 받아 이 번호로 POST /contactview request를 보내 DB에 입력 되어있는 연락처를 response로 받는다. 자신의 기존 이메일, 휴대폰 번호와 변경된 신규 이메일, 휴대폰 번호를 입력하여 기존 연락처가 response로 받은 연락처와 동일할 경우 신규 연락처로 DB상에 업데이트를 하며 기존 연락처가 동일하지 않을 경우 업데이트하지 않는다.

Network API 디자인 (AWS lambda – 안드로이드 간 통신 프로토콜 설명)

사용자는 기기에 이름, 고유번호, 성적, 연락처(이메일, 휴대폰 번호)를 입력하고 이를 aws flask lambda 서버의 두 가지 class에서 다음과 같은 함수를 이용해 통신할 수 있다.

- ① @app.route("/adduser", methods=['POST'])
고유번호(passwd), 이름(name)을 입력 받아 고유번호가 기존 DB에 존재하지 않을 때 회원가입을 진행하여 success 여부를 response로 보낸다. 이름은 중복이 존재할 수 있다.
- ② @app.route("/login", methods=['POST'])
고유번호(passwd), 이름(name)의 쌍을 입력 받아 해당 데이터 존재 유무에 따라 로그인 성공여부를 response로 보낸다.
- ③ @app.route("/gradeupdate", methods=['POST'])
고유번호(passwd)와 성적(grade) 쌍을 입력받아 해당 고유번호가 존재할 경우 그 인스턴스의 성적을 입력받은 값으로 업데이트하여 성공여부를 response로 보낸다.
- ④ @app.route("/gradeview", methods=['POST'])
고유번호(passwd)를 입력받아 해당 고유번호의 성적을 response로 보낸다.
- ⑤ @app.route("/updatecontact", methods=['POST'])
선생님 고유번호(tid), 이메일(email), 휴대폰번호(phonenum)을 받아 해당 고유번호의 데이터가 존재한다면 그 데이터의 이메일과 휴대폰번호를 업데이트하고 성공여부를 response로 보낸다.
- ⑥ @app.route("/updateemail", methods=['POST'])
선생님 고유번호(tid), 이메일(email)을 받아 해당 고유번호의 이메일을 수정한다.

- ⑦ @app.route("/updatenum", methods=['POST'])
선생님 고유번호(tid), 휴대폰번호(phonenum)를 받아 해당 고유번호의 번호를 수정한다.
- ⑧ @app.route("/contactview", methods=['POST'])
선생님 고유번호(tid)를 받아 해당 고유번호의 연락처를 response로 보낸다.

도전 요소 (Challenge)

- ① Aws 플라스크 람다 서버에 User, Contact 두 가지의 class 테이블을 생성하였다.
이 User, Contact에서 passwd와 tid만을 unique로 설정하여 나머지 값은 중복이 가능하게 처리하였고 이를 이용하여 데이터를 수정하는 기능을 추가하여 총 8개의 다양한 라우트 함수를 만들었다.
- ② POST를 사용하여 서버에서 response로 데이터를 받아 업데이트한 값을 연락처 listview를 조회할 때마다 자동으로 업데이트하는 기능을 만들었다.
- ③ 연락처를 조회하는 Listview에 각각 버튼을 추가하여 휴대폰번호 옆 버튼의 경우 누르면 해당 번호가 눌린 다이얼 (통화 어플)로 넘어가게 만들었다. 이메일 옆 버튼의 경우 이메일을 전송하는 모든 기능이 가능한 어플에 implicit intent를 보내 메일을 보낼 수 있는 어플 중 하나를 선택하는 팝업창이 뜨며 그 중 하나를 선택할 경우 해당 메일주소를 수신인에, "내용을 작성하세요"라는 메일 내용, 메일 제목이 자동으로 어플에 전송되어 입력되게 서버, 리스트뷰, 버튼과 intent의 기능을 하나로 묶어 만들었다.
- ④ 다양한 액티비티에 intent를 사용하여 서버 없이도 계정 정보를 기억할 수 있게 만들었다.