CS492 A 조 최종 보고서

– 바뀌어라 머리머리 –

20160633 조재민
20170057 김 건
20170719 한승희

사전 설문 조사 결과

어플리케이션을 개발하기에 앞서 주변 지인들을 대상으로 간단한 사전 설문조사를 진행하였다. 우선 첫번째 질문으로 지금의 헤어스타일을 유지한 기간을 물어봤을 때, 52 퍼센트의 사람들이 지금의 헤어스타일을 유지한 지 3 년이 지났다고 응답을 하였다. 이러한 응답 결과에 대해, "왜 헤어스타일을 그렇게 오래 유지 하셨나요?" 라는 항목으로 다시 설문을 진행해본 결과, 과반수의 사람들이 "패션을 몰라 어떻게 바꿔야할지 모르겠다"는 응답을 하였다. 이러한 설문 조사결과를 통해, 상당히 많은 사람들이 헤어스타일 변화를 시도해보고는 싶지만 자신에게 어떤 헤어스타일이 어울릴지 알지 못해 헤어스타일 변화를 망설이고 있다는 것을 알 수 있었다.

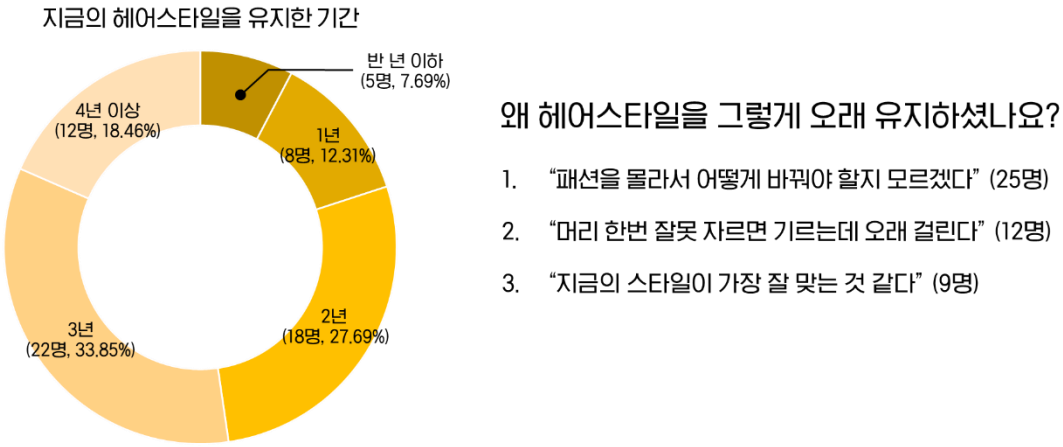

그림 1 사전 설문 조사 결과

어플리케이션 개요

사전 설문 조사 결과를 통해 자신에게 어떠한 헤어스타일이 어울리는지 모르는 사람들에게 가상으로 헤어스타일링을 해줄 수 있는 서비스에 대한 수요가 있을 것이라 판단을 하였다. 따라서 딥 페이크 프로젝트를 통해 "바뀌어라 머리머리"라는 이름의 어플리케이션을 개발하기로 결정하였다. "바뀌어라 머리머리" 어플리케이션을 통해 딥 페이크를 활용하여 사용자의 사진에 헤어스타일을 합성하여 변화한 자신의 머리를 미리 확인해 볼 수 있는 서비스를 제공하는 것을 목표로 하였다.

그림 2 어플리케이션 로고

시스템 설계

1. 프론트 엔드

바뀌어라 머리머리 어플리케이션의 프론트 엔드는 다음과 같이 설계되어 있습니다. 첫째, 어플리케이션 개발 환경으로는 안드로이드 스튜디오를 사용하였으며 개발 언어로는 자바를 사용하였다. 그리고 OS 는 안드로이드 10 (API 29)을 타겟으로 하여 개발을 진행하였다. 또한 프론트 엔드에서 백 엔드로 HTTP Request 를 보내기 위해 Retrofit2 라이브러리를 사용하였으며, 추가적인 기능 (구글 지도, 광고 서비스 등)을 구현하기 위해 Google API 를 활용하였다.

2. 백 엔드

1) 백 엔드 구조 개요

바뀌어라 머리머리의 백 엔드 서버는 총 4 개의 구성요소로 설계되어있다. 우선 웹 서버 프레임워크로는 Django 를 사용하였고, 데이터베이스로는 SQLite3 을 사용하였다. 그리고 백 엔드 서버를 외부 IP 에 위치해 있는 어플리케이션에서 접근할 수 있게 하기 위해 Ngrok 을 사용하였으며, 딥 페이크 모델로는 FaceSwap 을 이용하였다.

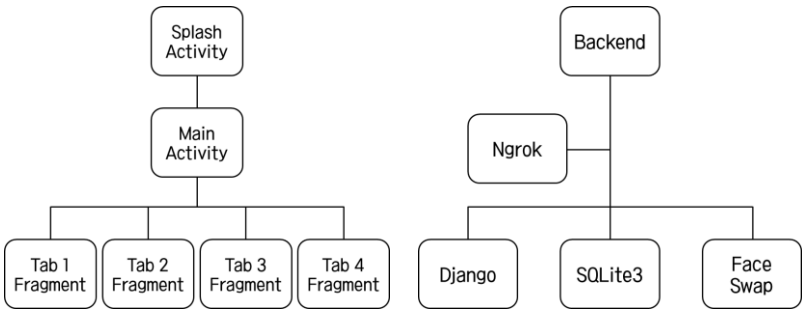

그림 3 프론트 엔드 및 백 엔드 시스템 구조

2) Django

Django 는 파이썬 기반 웹서버 프레임워크로 SQLite3 와의 연동성이 좋다는 장점이 있다. 또한 Python Django-restframework 라이브러리를 통해 프론트 엔드와 통신을 진행하였다.

3) SQLite3

SQLite3 는 데이터베이스를 관리하기 위한 추가적인 프로세스가 필요가 없기 때문에 다른 데이터베이스 관리 프로그램에 비해 훨씬 가볍다는 장점을 가지고 있으며, Python sqlite3 라이브러리를 통해 Django 웹 서버에서 데이터베이스를 쉽게 조작을 할 수 있다. 백 엔드에서는 어플리케이션의 다양한 종류의 데이터를 저장하기위해 5 종류의 테이블을 정의하였으며, 각 테이블의 저장되는 데이터는 다음과 같다.

(1) Login Info: 로그인 정보를 담기 위한 테이블로 ID 와 Password 정보가 담겨있다.

(2) User Info: 사용자 정보를 담기 위한 테이블로, ID, 이름, 사용자 타입 등의 정보가 담겨있다.

(3) Request Info: 리퀘스트 정보를 담기 위한 테이블로 사용자와 리퀘스트 ID 등이 담겨 있다.

(4) Hairshop Info: 미용실의 정보를 담기 위한 테이블로 미용실 이름, 위치, 평점 등이 담겨있다.

(5) Hairshop Time Info: 예약관련 정보를 담기 위한 테이블로, 미용실 ID 와 예약가능시간이 담겨 있다.

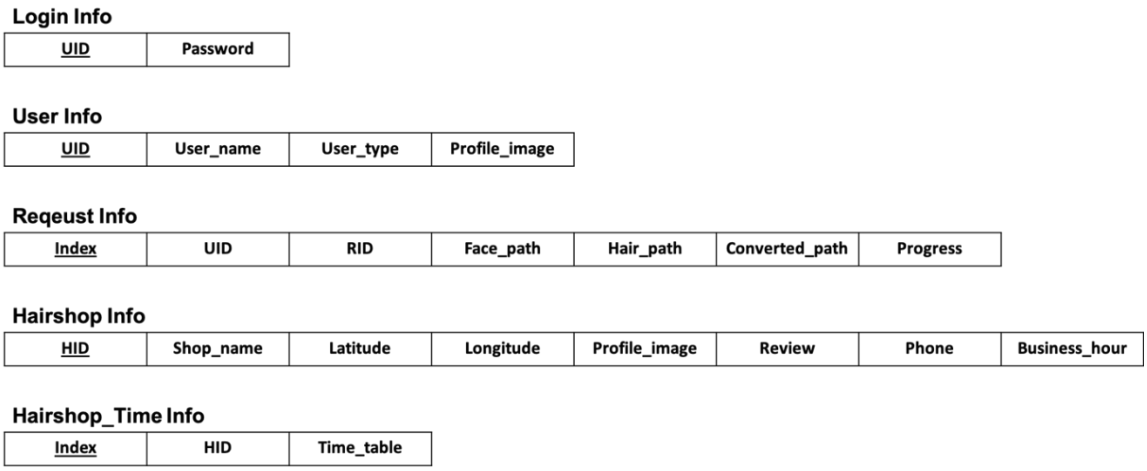

그림 4 SQLite3 데이터베이스의 Relational Data Model Schema

4) FaceSwap

FaceSwap 은 오픈소스 GitHub 라이브러리로 소스 이미지와 타겟 이미지에 특이적인 딥러닝 모델을 구축하고 학습을 진행하기 때문에 다른 모델에 비해 자연스러운 딥 페이크 이미지를 생성 가능하다는 장점이 있어 딥 페이크 모델로 채택하였다. 우리는 FaceSwap 을 사용하기에 앞서 모델에 대한 두가지 테스트를 진행해보았다. 첫 번째 테스트는 FaceSwap 와 다른 조에서 많이 채택한 딥 페이크 모델인 Few-shot face translation 을 비교해보았다. 결과는 다음과 같다. FaceSwap 을 사용하여 딥 페이크 이미지를 생성하였을 때 사용자의 얼굴의 모습이 Few-shot 에 비해 많이 남아 있고 얼굴에 대한 왜곡도 덜해 훨씬 자연스러운 이미지가 생성된다는 것을 확인할 수 있었다. 두 번째 테스트는 효율적인 학습을

하기 위한 최적 iteration 을 찾는 것이었다. 따라서 10,000 회, 50,000 회, 100,000 회, 그리고 200,000 회의 iteration 의 학습을 통해 만들어진 딥 페이크 이미지를 비교해본 결과 100,000 회까지 iteration 을 증가시켰을 때에는, 생성되는 이미지가 점점 자연스러워졌으며, 100,000 회가 넘어간 이후에는 만들어지는 이미지의 품질에는 영향을 주지 않았다. 따라서, 이 테스트를 통해 최적 iteration 은 100,000 회라고 알 수 있었다.

그림 5 FaceSwap 과 Few-Shot Face Translation 의 비교

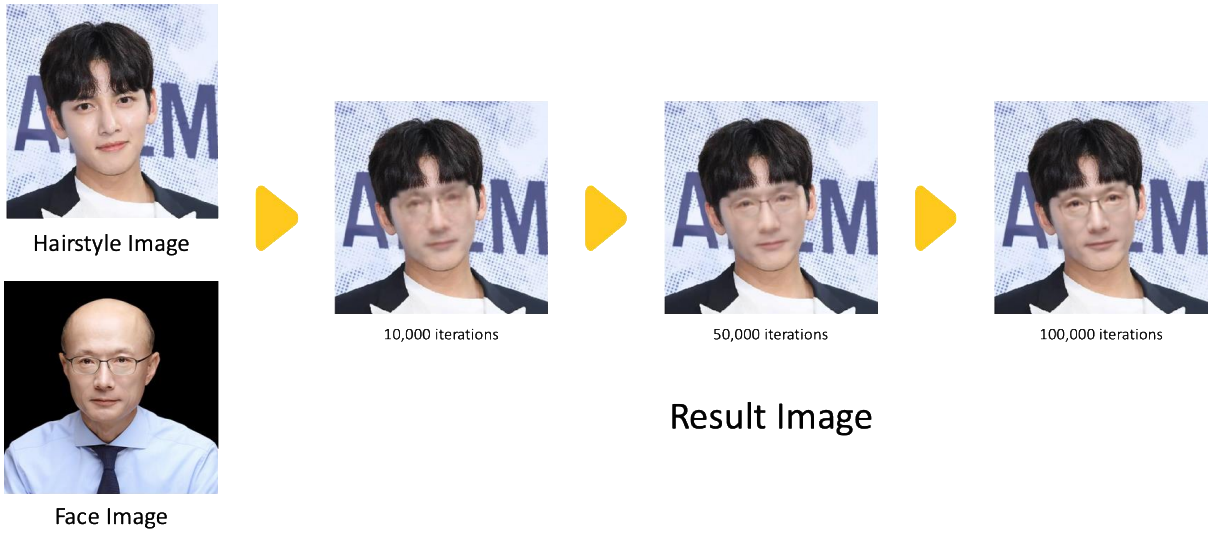

그림 6 Iteration 테스트

3. 핵심기능 동작 과정

바뀌어라 머리머리의 핵심 기능은 사용자가 원하는 헤어스타일에 사용자의 얼굴을 합성한 딥 페이크 이미지를 생성하는 것이다. 이 기능이 프론트 엔드와 백 엔드 단계에서 동작하는 과정은 다음과 같다. 우선 프론트 엔드에서 사용자가 원하는 헤어스타일을 선택하고 합성을 하기 위한 본인 사진을 촬영한다. 이렇게 결정한 헤어스타일 사진 정보와 사용자 얼굴 사진 파일을 백 엔드로 전송하면 백 엔드에서는 딥 페이크 프로세스에게 사진 데이터를 전달하여 딥 페이크 이미지 생성을 시작하도록 한다. 이후 프론트 엔드에서 완성된 이미지를 요청하면 백 엔드에서 해당 딥 페이크 이미지 데이터베이스에서 찾아 전송하는 방식으로 어플리케이션에서 완성된 이미지를 받아볼 수 있다.

어플리케이션 기능 구현

1. 로그인 기능

어플리케이션을 최초로 실행했을 때 가장 먼저 나오는 로그인 화면이다. 여기서 사용자는 로그인과 회원가입을 할 수 있다. 각 기능을 처리되는 과정은 다음과 같다. 우선 로그인 기능은 프론트 엔드에서 로그인 화면에 있는 아이디와 비밀번호 칸에 각각 사용자의 아이디와 비밀번호를 입력한 이후 로그인 버튼을 누르게 되면 입력된 아이디와 비밀번호를 백 엔드 서버에 전송한다. 서버에서는 데이터베이스에 해당 아이디와 비밀번호가 모두 일치하는 사용자가 있다면 로그인이 성공했다는 신호와 함께 로그인을 위한 유저 정보를 보내게 된다. 그리고 회원가입은 프론트 엔드에서 새 계정 만들기 버튼을 누르면 나오는 다이얼로그 창에 사용자의 아이디, 비밀번호, 이름을 입력하여 회원 가입 버튼을 누르면 서버로 해당 데이터가 전달된다. 이후 서버에서는 해당 아이디와 중복된 아이디가 없다면 그 사용자 정보를 저장하고, 어플리케이션으로 회원가입이 성공했다는 신호를 보내게 된다.

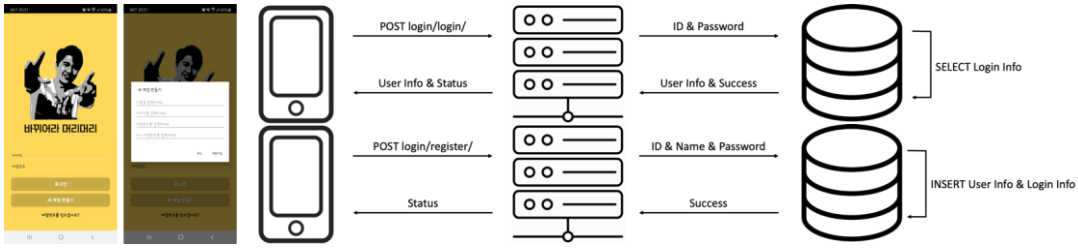

그림 7 로그인 화면 (좌), 로그인 기능 및 회원 가입 기능 처리 과정 (우)

2. 헤어스타일 추천 및 딥 페이크

로그인을 성공하면 탭 형태의 화면이 나오는데, 각각의 탭마다 어플리케이션의 서브 기능들이 구현되어 있다. 첫번째 탭에는 랭킹 기능이 구현 되어있다. 이 곳에서는 다양한 헤어스타일들을 남성과 여성, 긴 머리와 짧은 머리로 나누어 볼 수 있는데, 인기가 많은 순으로 정렬이 되어 있어서 어떤 헤어스타일이 유행하는지 한눈에 파악할 수 있다. 이때, 여기서 말하는 인기 순은 특정 헤어스타일에 좋아요 버튼을 누른 사용자의 숫자에 의해 결정된다. 좋아요 버튼은 각 헤어스타일 사진의 우측 하단에 위치해 있다. 그리고 헤어스타일 사진을 클릭하게 되면, 다이얼로그 창을 통해 큰 화면으로 사진을 볼 수 있다.

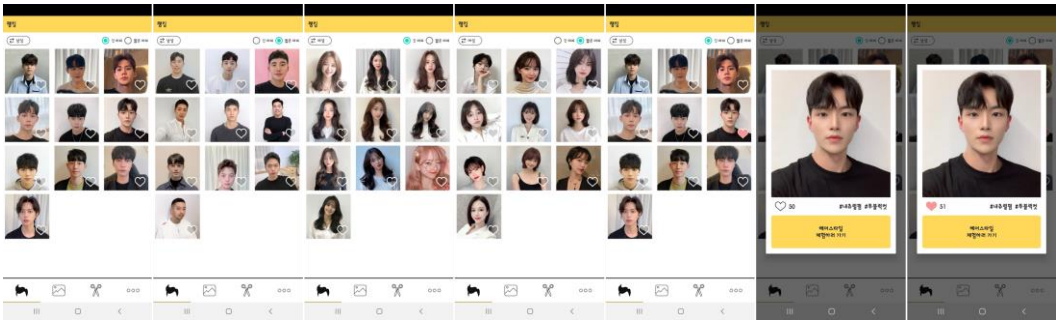

그림 8 랭킹 화면 구성

이때, 다이얼로그 창 맨 밑에 있는 헤어스타일 체험하러 가기 버튼을 누르게 되면 카메라를 통해 얼굴 사진을 촬영할 수 있는 인터페이스로 전환이 된다. 이 창에서 카메라 화면에 떠있는 가이드라인에 맞춰 얼굴의 정면 사진, 양쪽 측면 사진을 찍을 수 있다. 그리고 세 장을 다 찍고 나면, 촬영 버튼 오른쪽에 OK 라는 버튼이 활성화된다. 이 버튼을 누르게 되면 유저 ID, 헤어 스타일 정보, 3 장의 사용자 얼굴 사진이 백 엔드 서버로 전송이 된다. 그러면 서버에서는 해당 정보들을 서버의 데이터베이스에 저장을 하게 된다. 이후에 딥 페이크 프로세스가 아직 딥 페이크 이미지 생성이 시작되지 않은 리퀘스트를 발견하게 되면 해당 리퀘스트의 이미지 생성을 시작하게 된다. 문제 없이 해당 데이터가 데이터베이스로 저장이 된다면 서버는 새로 발급한 리퀘스트 아이디와 함께 어플리케이션으로 성공 신호를 보낸다.

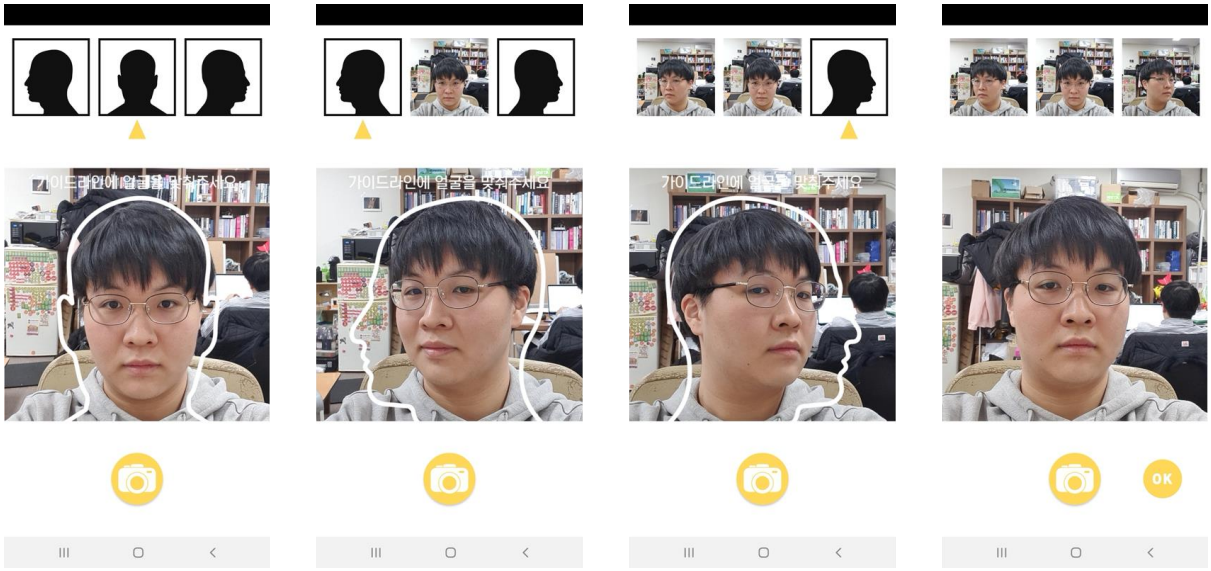

그림 9 사용자 얼굴 사진 촬영 화면

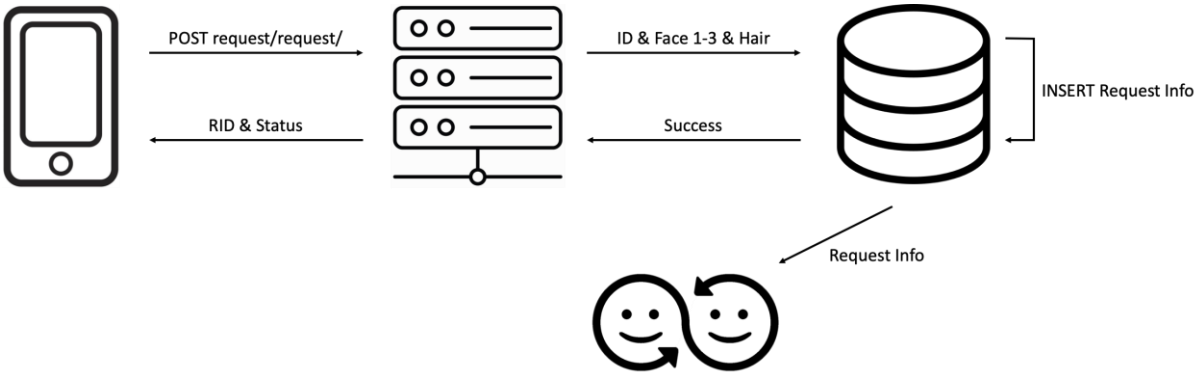

그림 10 딥 페이크 이미지 생성 요청 처리 과정

3. 갤러리

두 번째 탭에서는 첫 번째 탭에서 딥 페이크 이미지 생성 요청해 놓은 리퀘스트 항목을 확인할 수 있다. 딥 페이크 리퀘스트는 얼굴 추출 중, 얼굴 변환, 이미지 합성, 완료의 4 가지 단계로 진행상황을 확인할 수 있다. 그리고 두 번째 탭에서 화면을 밑으로 당김으로써 갤러리를 업데이트 할 수 있다. 완료 상태의 사진을 클릭하게 되면, 다이얼로그 창이 뜨면서 더 큰 사이즈로 완료된 이미지를 볼 수 있는데, 다이얼로그 창의 우측 하단에 위치한 세가지 버튼을 통해서, 저장, 공유 그리고 삭제를 기능을 수행할 수 있습니다. 두번째 탭의 처리과정은 다음과 같습니다. 우선 업데이트 기능은 사용자가 프론트 엔드에서 사용자의 아이디와 함께 업데이트 요청을 보내게 되면, 서버에서는 해당 아이디를 가지는 사용자의 딥 페이크 리퀘스트 항목 리스트를 데이터베이스에서 찾아 유저에게 보내는 식으로 처리가 된다. 다운로드 기능은 프론트 엔드에서 사용자 아이디와 리퀘스트 아이디를 서버로 보냈을 때, 서버에서 해당 리퀘스트 항목을 데이터베이스에서 찾아 완료된 이미지를 어플리케이션으로 보내는 방식으로 처리가 된다. 그리고 삭제 기능은 다운로드와 같이 사용자와 리퀘스트 아이디를 서버로 보냈을 때, 서버에서 해당 리퀘스트 항목을 데이터베이스에서 삭제하는 방식이다.

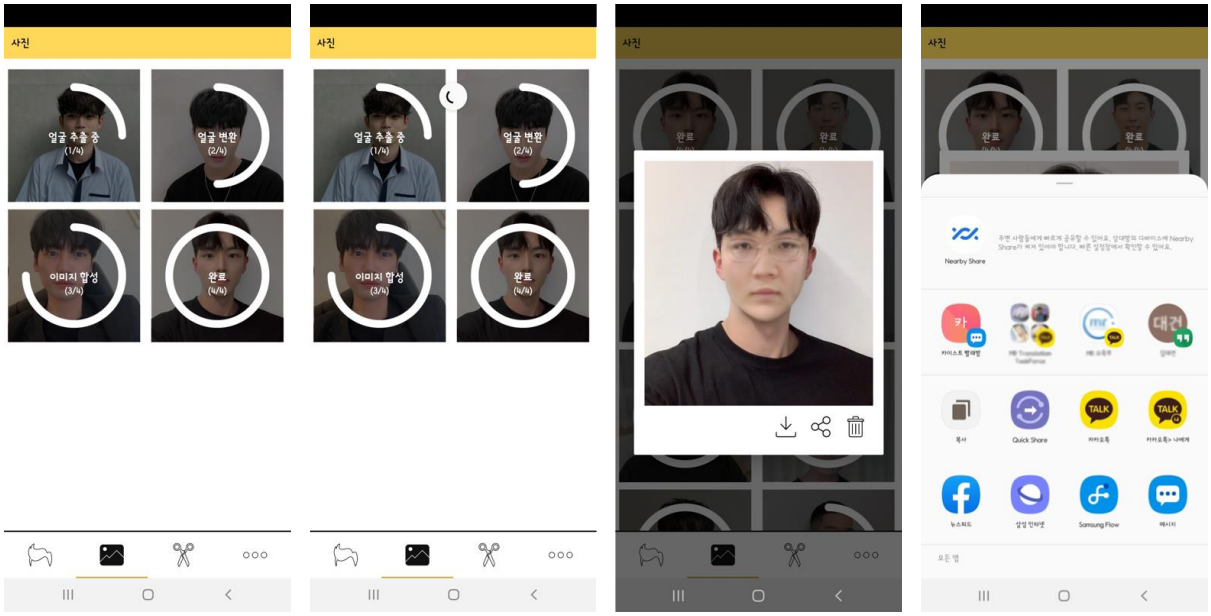

그림 11 갤러리 화면

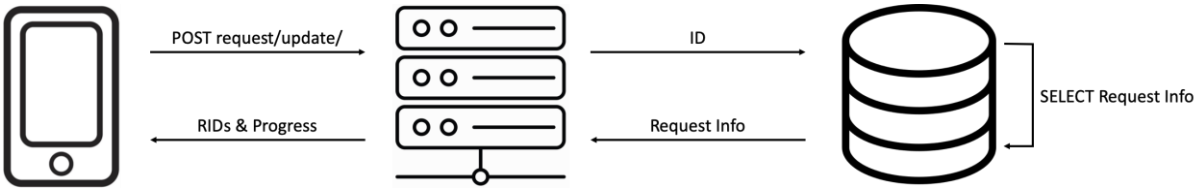

그림 12 업데이트 기능 처리 과정

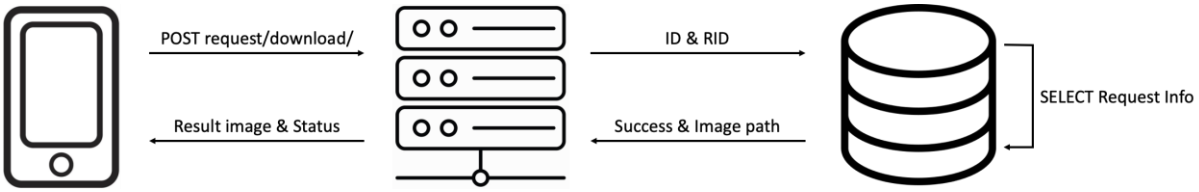

그림 13 다운로드 기능 처리 과정

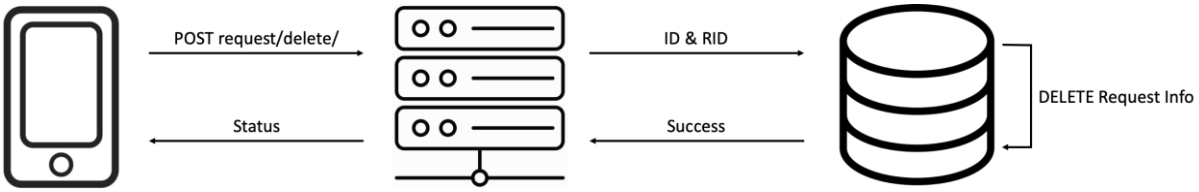

그림 14 삭제 기능 처리 과정

4. 미용실 예약

세 번째 탭에는 지도와 함께 미용실의 목록들이 나온다. 최초로 현재 탭을 실행하였을 때, 지도는 현재 사용자의 위치를 중심으로 보여 주변 미용실의 정보를 보여준다. 이때, 특정 미용실의 사진을 누르면 해당 미용실의 위치를 지도에 표시해 줍니다. 그리고 해당 사진을 한번 더 누르면, 다이얼 로그 창이 열리면서 그 미용실에 대한 자세한 정보를 볼 수 있다. 이 창에서 사용자는 미용실 예약을 할 수 있는데, 사용자가 원하는 시간을 클릭 하고, 본인 사진 추가 버튼을 통해 어플리케이션에서 생성된 딥 페이크 사진을 첨부하여 예약을 할 수 있다. 그리고 해당 탭의 미용실 정보 업데이트 하는 과정은 다음과 같다. 어플리케이션에서 사용자의 현재 위치 정보를 서버로 보내게 되면, 서버에서는 데이터베이스 상의 그 위치를 중심으로 일정 거리 안에 위치한 미용실들의 정보를 어플리케이션으로 반환하는 식으로 처리된다. 또한 예약 기능은 사용자의 아이디와 함께, 선택한 미용실의 미용실 ID, 예약 시간, 그리고 사용자의 딥 페이크 이미지를 서버로 전송하고, 서버는 해당 미용실로 그 정보를 전달해주는 방식으로 처리된다.

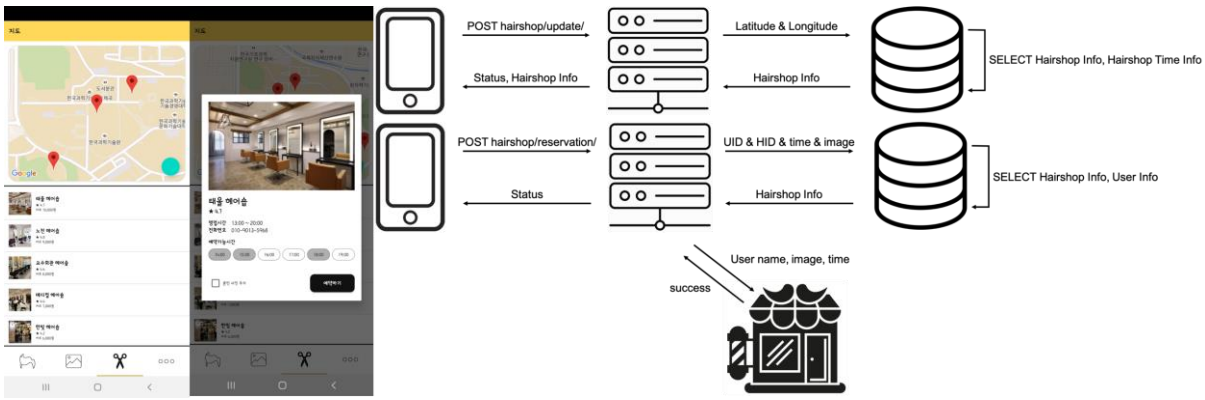

그림 15 주변 미용실 위치정보가 담겨 있는 지도 화면 (좌), 업데이트 및 예약 기능 처리 과정 (우)

5. 기타 설정

네 번째 탭은 마이페이지로 사용자의 프로필 정보를 보여줌과 동시에 사용자의 정보를 수정할 수 있는 기능을 가지고 있다. 첫 번째 기능은 멤버쉽 기능으로, 멤버십 버튼을 클릭하게 되면, 프리미엄 회원으로 전환할 수 있는 다이얼로그 창이 나오며, 프리미엄 회원의 각종 혜택을 확인하고 프리미엄 전환을 수행할 수 있다. 그리고 프로필 사진을 클릭하게 되면, 갤러리에서 사진을 선택하여 프로필 사진을 변경할 수 있다. 회원 전환 기능은 프론트 엔드에서 서버로 사용자 아이디 정보를 전송하게 되면 데이터베이스 상의 회원 정보를 수정하는 방식으로 처리하며, 프로필 사진 추가 기능은 서버로 사용자 아이디와 프로필 사진 파일을 전송하면 데이터베이스에 해당 사진 파일을 저장하고 데이터베이스에 해당 파일의 경로를 저장하는 식으로 구현하였다.

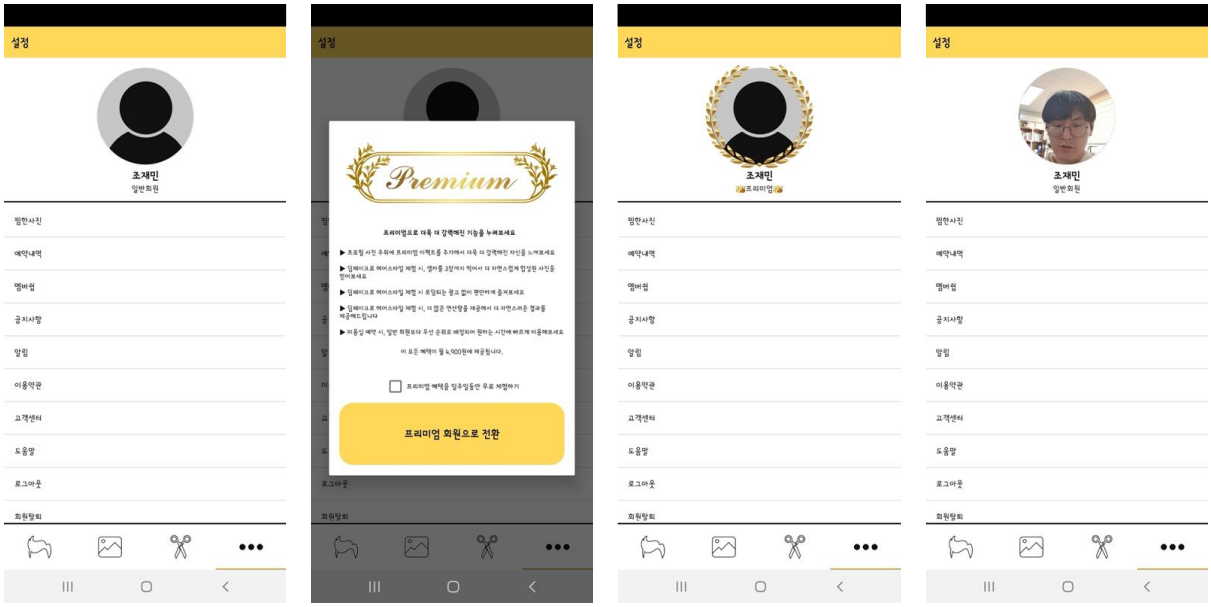

그림 16 마이페이지 화면

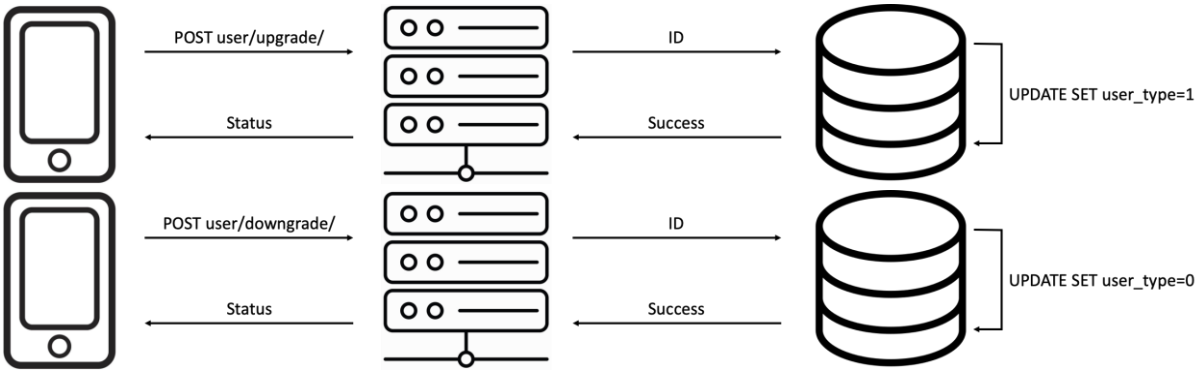

그림 17 프리미엄 회원 전환 처리 과정 (상), 일반 회원 전환 처리 과정 (하)

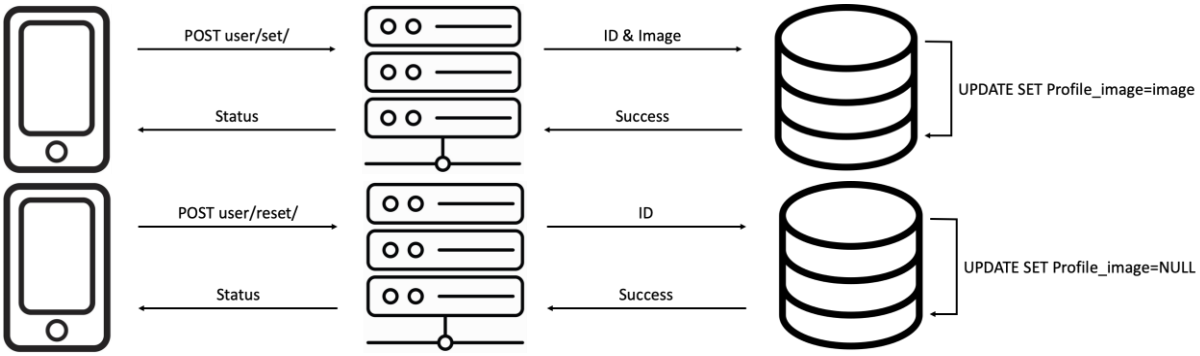

그림 18 프로필 사진 설정 처리 과정 (상), 프로필 사진 제거 처리 과정 (하)

비즈니스 모델

비즈니스 모델은 다음과 같이 세가지 타입으로 구성되어 있다.

1. Google – AdMob

구글에서는 개발자들이 어플리케이션에 쉽게 광고를 넣고, 수익을 창출 할 수 있도록 AdMob 라이브러리를 제공하고 있다. 이 라이브러리를 통해 리워드 형 광고를 어플리케이션에 탑재하여 딥 페이크 기능에서 "헤어스타일 체험" 버튼을 클릭했을 때, 해당 광고를 사용자에게 노출하는 식으로 수익을 창출할 수 있도록 하였다.

2. 유저 멤버쉽

두 번째 비즈니스 모델은 유저 멤버쉽이다. 사용자를 일반 회원과 프리미엄 회원으로 구분하여, 일반 회원이 월 4,900 원의 결제를 통해 구독형으로 프리미엄 회원 전환을 할 수 있게 만들었다. 프리미엄 회원에게는 프로필 사진 주위에 프리미엄 이펙트 제공하며 딥 페이크로 헤어스타일 체험 시, 프리미엄 회원은 얼굴 사진을 3 장 그리고 일반 회원은 한 장만 찍을 수 있게 하여 프리미엄 회원에게 더 자연스러운 합성 결과를 제공한다. 또한 프리미엄 회원에게는 리워드형 광고를 생략할 수 있게 만들어 주는 동시에, 미용실 예약 시 프리미엄 회원에게는 우선 순위로 배정해주는 등의 혜택을 제공해주는 방식으로 일반 회원에게 프리미엄 회원 전환을 유도하여 수익을 창출한다.

3. 미용실과의 연계

마지막으로 미용실과의 서비스 연계이다. 미용실과 정기 계약을 맺어, 세 번째 탭에서 사용자들에게 계약이 체결된 미용실들의 정보를 보여준다. 그렇게 되면 사용자들은 미용실에 쉽게 접근하여 예약을 할 수 있고, 미용실에서는 홍보효과와 함께, 유저와 헤어스타일이 합성된 사진을 미리 보고, 유저들에게 더 원활한 미용 서비스를 제공할 수 있게 된다. 현재 미용실과의 계약금은 월 100,000 원으로 설정하였으며 이러한 홍보 효과는 어플리케이션의 사용자가 늘어날수록 증가되기 때문에 어플리케이션의 이용자 수가 증가함에 따라 이를 고려하여 계약금의 상승을 고려하고 있다.

역할분담

김 건: 백 엔드 서버 개발

- Django 웹서버 구축 및 프론트 엔드와의 통신 API 설계
- SQLite3 데이터베이스의 Data Model 정의 및 데이터베이스 구축
- 딥 페이크 이미지 생성프로세스 설계
- FaceSwap 모델 테스트 진행 및 최적 Iteration 측정

조재민: 프론트 엔드 어플리케이션 개발

- 어플리케이션 로고, 페이지, 컨셉 컬러 등 어플리케이션의 전반적인 디자인 진행
- 헤어 스타일 선택 화면, 미용실 예약 화면 등 유저 편의성 인터페이스 구현
- 프로필 사진 및 멤버쉽 등 개인 정보 관련 유저 인터페이스 구현
- Google AdMob 활용 광고 수익 창출 기능 구현

한승희: 프론트 엔드 어플리케이션 개발

- 프론트 엔드 - 백 엔드간 어플리케이션의 전반적인 네트워크 통신 관련 파트 구현
- 카메라를 활용한 사진 촬영, 저장, 삭제 및 갤러리 관련 파트 등 딥 페이크 기능 관련 유저 인터페이스 구현
- 로그인, 회원가입 관련 기능 구현
- Google map API 활용 지도 및 미용실 마킹 기능 구현

소스코드

프론트 엔드 - https://github.com/jjm7307/HairChanger

백 엔드 - https://github.com/skygun88/HairChangerBackend