

# 캡스톤 디자인

## Final report



**Team**                      너랑나랑은

**Member**            20161851 조민수

20163828 김현석

20153819 감지훈

## **목차**

### **1. 개요**

#### **1.1 프로젝트 소개 및 목표**

### **2. 개발 내용**

#### **2.1 설계 및 아키텍처**

#### **2.2 시나리오**

##### **2.2.1 모바일**

##### **2.2.2 웹 및 관리자페이지**

#### **2.3 차별성**

### **3. 프로젝트 수행 결과**

#### **3.1 기대효과**

#### **3.2 결과물**

# 1. 개요

## 1.1 프로젝트 소개

대학생들은 사회로 나가기 전, 진로를 선택하는 과정에서 진로목표를 설정하고 구체적인 계획을 세운다. 진로목표 설정 과정에선 흔히 성공한 사례를 가진 사람들을 모티브로 삼고, 그들의 행보를 쫓기도 합니다. 그럼 단순히 그들의 행동을 따라한다고 해서 모두가 좋은 결과를 얻을 수 있을까요?

물론 성공한 사례를 가진 사람들은 그 분야에서 많은 노력을 하여 그들의 경험이나 조언을 듣는 것만으로도 충분히 도움이 될 수 있습니다. 하지만 멘티마다 기본적인 배경지식이나 상황이 각각 달라 멘토 입장의 멘토링은 좋은 이정표가 될 수도 오히려 길을 잃게 만들 수도 있습니다. 예를 들면 술 먹기전에 숙취해소제를 먹으면 숙취가 덜한다고 하지만 체질 상의 이유 든 과도하게 섭취를 했든 어떤 조건에 의해서 어떤 사람에겐 효과가 미미하게 나타날 수 있습니다. 즉 조언을 듣는 사람의 상황(입장)마다 받아들이는 조언이 효과적일지는 모릅니다.

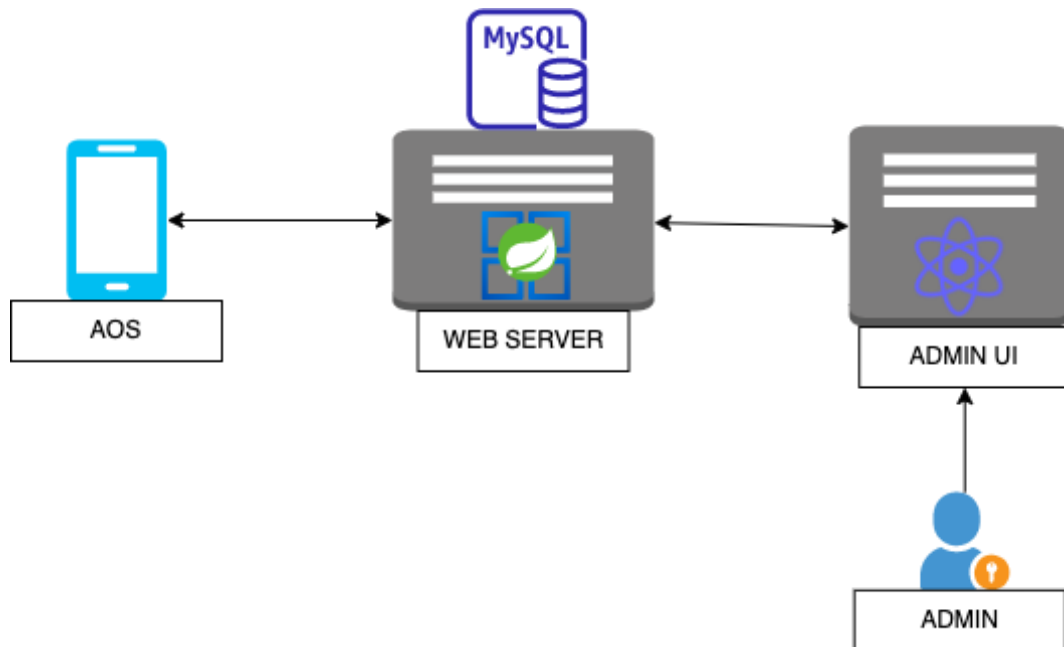


이렇게 멘토링은 멘토의 사례보다는 멘티에게 가장 도움이 될 법한 멘토들을 추천해 줌으로써 조언을 효과적으로 받아들일 수 있도록 앱을 고안했습니다. 멘티의 모든 능력을 측정하지 못하지만, 학점, 학교 같이 능력을 대변할 수 있는 지표들을 보고 이에 맞는 멘토들을 추천해주어 멘토링의 만족도를 올릴 수 있도록 진행하였습니다.대

학생의 경우만 해도 2명 중 1명이 진로고민을 할 정도로 많은 사람들이 진로에 대해 고민합니다. 그러니 우리 사이트가 진로를 헤매는 사람들의 이정표가 되었으면 합니다.

## 2. 개발 내용

### 2.1 설계 및 아키텍처



유저 정보 관리 서버는 상시 켜져 있어야 하기 때문에 GCP 서버를 사용했고 Java-spring 과 Mysql 을 이용하여 웹서버를 구축하였다. 이때 Restful 서비스를 이용하여 앱과의 연동을 진행했습니다.

앱에선 안드로이드 스튜디오를 이용해 작업하였고, kotlin 언어를 사용했다. 추가적으로 retrofit2 라이브러리를 사용하여 서버와의 연동을 진행했습니다.

## 2.2 시나리오

### 2.2.1 모바일

#### 1. 네트워크 구성

- Retrofit2를 사용해서 rest API의 GET,POST 그리고 이미지 파일을 보내기 위한 Multipart를 사용하였다.

AndroidManifest.xml에 인터넷 연결을 위한

`<uses permission android:name="android.permission.INTERNET" />`를 추가하였고,

Multipart에서 이미지파일을 받기위해 폼데이터를 만드는데 이때 필요한 경로를 읽기위해

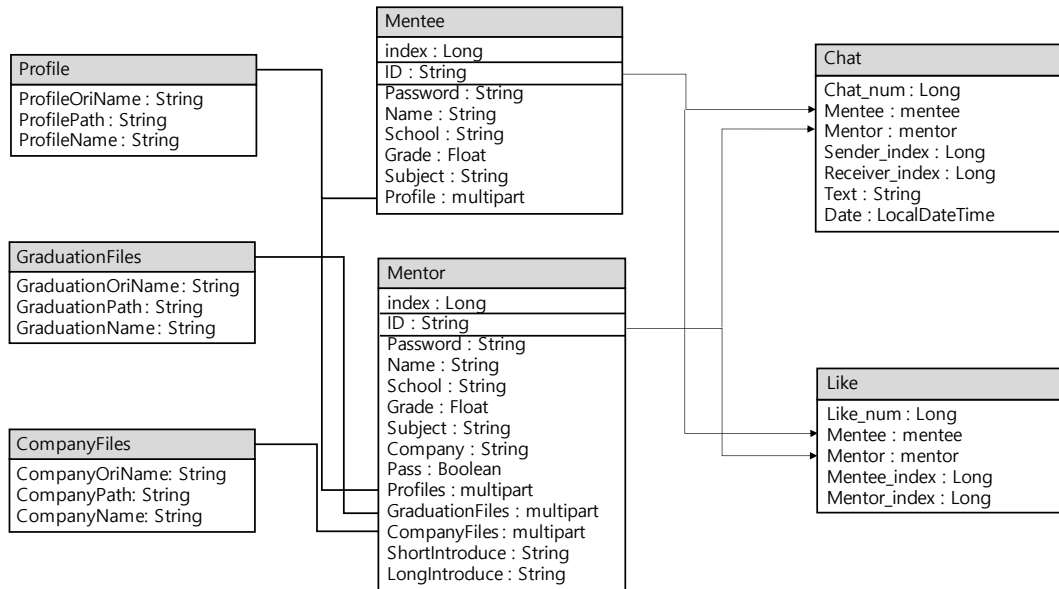
`<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />`를 추가하였다.

#### 2. UI/UX 구성

- 회원가입 페이지에서는 대학, 학과 검색을 위한 searchspinner 라이브러리를 사용하였다.
- 멘티페이지에서는 navigation에 4개의 fragment를 담아 구성했다. 멘토 추천 페이지는 cardstackview 라이브러리를 사용하여 카드스택뷰를 사용하였고, 전체 멘토 리스트와 채팅, 좋아요한 리스트는 각각의 정보를 제공하는 ListView를 사용하였다. 멘토 상세검색과 채팅에 맞는 형식의 다이어로그를 사용하였고, 전체적인 이미지 url 데이터를 출력하기위해 glide 라이브러리를 사용하였다.
- 멘토페이지에서는 navigation에 3개의 fragment를 담아 구성하였다. 좋아요한 멘티페이지, 채팅 모두 각각의 정보에 맞게 ListView를 사용하여 구성하였다.

## 2.2.2 웹 및 관리자페이지

웹서버는 GCP 서버를 통해 상시 구동하였고 java-spring 과 이용하여 설계하였다. Restful 서비스를 이용해 모바일과의 연동을 진행했고 Mysql 을 DB 로 사용했습니다



### < 데이터베이스 스키마 >

#### ● Mentee

Index	Mentee 의 고유 index 값
ID	Mentee 의 ID
Password	Mentee 의 password
Name	Mentee 의 이름
School	Mentee 의 학교
Grade	Mentee 의 성적
Subject	Mentee 의 학과
Profiles	Mentee 의 프로필 정보

멘티 정보를 가지고 있는 Table 입니다. Java-spring 에서 프로필 파일을 전송을 위해 multipart 를 사용하였고, 비밀번호는 해시함수를 이용하여 암호화를 시킨 뒤 저장했습니다. 추후 recommend 에션 해당 멘티의 학교, 성적, 학과를 이용하여 추천리스트를 보여주게 됩니다.

- Mentor

Index	Mentor의 고유 index 값
ID	Mentor의 ID
Password	Mentor의 password
Name	Mentor의 이름
School	Mentor의 학교
Grade	Mentor의 성적
Subject	Mentor의 학과
Company	Mentor의 회사
Pass	Mentor의 승인여부
Profiles	Mentor의 프로필 정보
GraduationFiles	Mentor의 졸업증명서 정보
CompanyFiles	Mentor의 재직증명서 정보
ShortIntroduce	Mentor의 한 줄 자기소개
LongIntroduce	Mentor의 자기소개

멘토 정보를 가지고 있는 Table입니다. 멘티와 마찬가지로 파일 전송을 위한 multipart를 이용 하 되, 멘토의 신뢰성을 위한 졸업증명서( graduationFiles)와 재직증명서(companyFiles)를 추가로 받습니다. 관리자는 이 파일들을 보고 수동으로 진위확인을 진행하여 멘토가 신뢰성이 있다고 판단되면 ‘승인’을 눌러 pass 값을 true로 바꿉니다. pass 값이 true 여야 서비스를 사용 가능하도록 하였습니다.

- Chat

Chat_num	채팅의 고유 index 값
Mentee.ID	Mentee 정보
Mentor.ID	Mentor 정보
Sender_index	채팅을 보내려는 사람의 고유 index
Receiver_index	채팅을 받으려는 사람의 고유 index
Text	채팅으로 보내려는 메시지
Date	채팅이 전송된 시간

채팅 Table 은 멘티가 멘토에게 혹은 멘토가 멘티에게 보낼 수 있으므로 Sender 와 Receiver 의 ID 값을 받은 후 고유 index 값을 받아 데이터 처리를 빨리 하도록 했습니다.

- Like

Like_num	좋아요의 고유 index 값
Mentee.ID	Mentee 정보
Mentor.ID	Mentor 정보
Mentee_index	좋아요한 Mentee 의 고유 index
Mentor_index	Mentee 에게 좋아요를 받은 Mentor 의 고유 index

좋아요 Table 은 채팅과 마찬가지로 멘티 입장에서든 좋아요 한 멘토의 정보를 받아야하고 멘토 입장에서든 자신을 좋아요 한 멘티들 정보를 받아야 하므로 멘티와 멘토 모두의 정보가 필요합니다. 마찬가지로 빠른 처리를 위해 index 값만 저장하도록 했습니다.

- Profiles

ProfileOriName	저장된 프로필의 원본이름
ProfilePath	프로필이 저장된 경로
ProfileName	중복 방지를 위한 임의값 + 프로필 이름

- GraduationFiles

GraduationOriName	저장된 졸업증명서의 원본이름
GraduationPath	졸업증명서가 저장된 경로
GraduationName	중복 방지를 위한 임의값 + 졸업증명서 원본이름



- CompanyFiles

CompanyOriName	저장된 재직증명서의 원본이름
CompanyPath	재직증명서가 저장된 경로
CompanyName	중복 방지를 위한 임의값 + 재직증명서 원본 이름

3가지 class 는 java-spring 에서 파일 전송을 위해 multipart 를 사용했고, 파일 이동을 위해 필요한 값들을 저장해 놓았습니다. 경로 + 이름으로 이미지를 불러올 수 있습니다.

웹과 앱의 연동을 위해 restful api 를 이용하여 get 과 post 방식으로 데이터를 주고 받도록 했습니다. 어디에서 어떤 api 를 사용했는지 살펴보도록 하겠습니다.

- 홈 컨트롤러

1. (GET) <http://34.125.157.72:8080/>

첫 로그인 화면입니다. 관리자 계정으로 로그인 할 수 있고 초기 설정값은 “ID = admin / password = admin “으로 설정했습니다.

2. (POST) <http://34.125.157.72:8080/>

로그인할 때 administrator 데이터베이스에 해당 id 와 password 가 있는지 확인합니다. 있다면 로그인이 성공하고 없다면 실패한다. 로그인에 성공하면 멘티/멘토 리스트(목록)를 확인할 수 있고, 멘토 목록을 통해 멘토 승인까지 진행할 수 있습니다.

- 멘티 컨트롤러

1. (POST) <http://34.125.157.72:8080/mentees/new>

멘티의 회원가입을 위한 api 입니다. 앱에서 관련 정보들을 받으면 mentee 데이터베이스에 mentee 정보들을 저장합니다. 이곳에서 이미지는 사전에 설정했던 “/images/” 경로에 저장되고, 비밀번호는 hash 함수를 통해 암호화되어 데이터베이스에 저장됩니다.

2. (GET) <http://34.125.157.72:8080/mentees/new>

관리자페이지에서 간단한 멘티 회원가입을 구현했습니다.

3. (GET) <http://34.125.157.72:8080/mentees>

관리자페이지에서 멘티들의 리스트를 보기 위해 구현했습니다.

4. (POST) <http://34.125.157.72:8080/mentees/join>

멘티들의 로그인을 위한 api 입니다. 입력된 ID 와 password 가 데이터베이스에 저장됐는지 확인합니다. 물론 password 는 hash 함수 거쳐 비교하게 됩니다. 그 이외에도 multipart 를 이용하여 파일도 저장 받았습니다.

5. (POST) <http://34.125.157.72:8080/mentees/join/like>

좋아요 기능을 위한 api 입니다. 좋아요를 하려는 멘티, 좋아요를 당하는 멘토들의 index 들을 Like 데이터베이스에 저장했습니다.

6. (GET) <http://34.125.157.72:8080/mentees/join/like>

좋아요 리스트입니다. 해당 멘티가 좋아요한 멘토들의 정보를 출력했습니다. 이때 멘토들을 클릭하면 멘토들의 상세정보까지 확인할 수 있게 했습니다.

7. (POST) <http://34.125.157.72:8080/mentees/join/unlike>

싫어요 기능을 위한 api 입니다. 좋아요와 반대기능을 하며, Like 데이터베이스에 해당 멘티/멘토들의 index 를 가지고 있는 column 을 삭제했습니다.

8. (POST) <http://34.125.157.72:8080/mentees/join/chat>

채팅 기능을 위한 api 입니다. 멘티가 멘토에게 채팅을 보내는 기능으로, 멘티/멘토 index, 내용, 그리고 현재시간이 chat 데이터베이스에 저장됩니다.

9. (GET) <http://34.125.157.72:8080/mentees/join/chat>

채팅 리스트입니다. 멘티가 보내거나, 받았던 채팅 내용들을 전부 받아와 일어난 시간 순서대로 정렬해 놓았습니다.

10. (POST) <http://34.125.157.72:8080/mentees/join/modifyFile>

멘티의 프로필사진을 바꾸기 위한 api 입니다. 프로필 사진 이외에는 이전 데이터를 유지하도록 하였고 보다 깔끔한 작업을 위해 이전 프로필 사진을 삭제하는 작업까지 진행했습니다.

11. (POST) <http://34.125.157.72:8080/mentees/join/modifyInfo>

멘티의 정보를 바꾸기 위한 api 입니다. 학생은 여러가지 이유로 학교, 학점, 학과가 바뀔 수 있으므로 정보를 수정할 수 있는 api 를 제공하도록 했습니다.

12. (GET) <http://34.125.157.72:8080/mentorsMatchingList>

멘티에게 멘토를 추천해주는 리스트입니다. 기본적인 추천은 학교, 학점, 학과를 기준으로 이루어지고 입력 받은 조건들에 부합하는 멘토들을 리스트로 추천해줍니다. 이때 Like 데이터베이스를 확인하여 이미 좋아요 한 멘토는 리스트에서 제외되도록 하여 가시성을 높였습니다.

13. (GET) <http://34.125.157.72:8080/mentees/likeList>

자신이 좋아요 누른 멘토들의 리스트입니다. 이는 채팅 기능에서 좋아요 누른 멘토와 채팅을 진행하기 위해 필요하고, 또한 좋아요를 취소(싫어요) 할 수도 있게 설계했습니다.

● 멘토 컨트롤러

1. (POST) <http://34.125.157.72:8080/mentors/new>

멘토의 회원가입을 위한 api 입니다. 멘티와 받는 정보는 거의 같으나 추가적으로 재직중인 회사, 재직증명서 파일, 졸업증명서 파일을 입력 받습니다. 재직증명서, 졸업증명서는 각각 “/graduation\_certification/” “/company\_certification/” 경로에 저장됩니다.

2. (GET) <http://34.125.157.72:8080/mentors/new>

관리자페이지에서 간단한 멘토 회원가입을 구현했습니다.

3. (GET) <http://34.125.157.72:8080/mentors>

관리자페이지에서 멘토들의 리스트를 보기 위해 구현했습니다. 이곳에서 멘토들의 서류를 수동적으로 심사하여 승인절차를 내립니다.

4. (GET) <http://34.125.157.72:8080/mentorList>

멘티가 이용하는 기능이지만 멘토에 대한 내용이므로 `mentorController` 에 집어넣었습니다. 전체 멘토 리스트로, 모든 멘토들을 보거나 직접 필터링하여 멘토들을 선별할 수 있습니다.

5. (GET) [http://34.125.157.72:8080/mentors/{mentor\\_id}/pass](http://34.125.157.72:8080/mentors/{mentor_id}/pass)

3 번의 내용과 이어집니다. 관리자페이지에서 승인절차를 내릴 때 사용한 api 입니다. 단순히 pass 값을 true 로 설정해주었습니다.

6. (POST) <http://34.125.157.72:8080/mentors/join/adminChat>

관리자가 멘토에게 채팅을 보내는 api 입니다. 단방향으로 이루어지고, 멘토들의 승인절차에 문제가 생겼을 경우 통보하기 위해 개발했습니다.

7. (GET) <http://34.125.157.72:8080/mentors/join/adminChat>

멘토가 관리자에게 받은 메시지를 확인할 수 있는 api 입니다. 말했듯 단방향 통신으로 멘토가 관리자에게는 보내지 못하도록 했습니다,

8. (POST) <http://34.125.157.72:8080/mentors/join>

멘토들의 로그인을 위한 api 입니다. 전체적인 기능은 위의 멘티로그인(4)와 같습니다.

9. (POST) <http://34.125.157.72:8080/mentors/join/chat>

채팅 기능을 위한 api 입니다. 멘토가 멘티에게 채팅을 보내는 기능으로, 멘티채팅(8)과 기능이 같습니다.

10. (GET) <http://34.125.157.72:8080/mentors/join/chat>

채팅 리스트입니다. 멘토가 보내거나, 받았던 채팅 내용들을 전부 받아와 일어난 시간 순서대로 정렬해 놓았습니다.

11. (POST) <http://34.125.157.72:8080/mentors/join/modifyProfile>

멘토의 프로필사진을 바꾸기 위한 api 입니다. 프로필만 수정하므로 나머지 값들은 이전과 동일하도록 설정했습니다.

12. (POST) <http://34.125.157.72:8080/mentors/join/modifyGraduationFile>

멘토의 프로필사진을 바꾸기 위한 api 입니다. 프로필만 수정하므로 나머지 값들은 이전과 동일하도록 설정했습니다.

13. (POST) <http://34.125.157.72:8080/mentors/join/modifyCompanyFile>

멘토의 프로필사진을 바꾸기 위한 api 입니다. 프로필만 수정하므로 나머지 값들은 이전과 동일하도록 설정했습니다.

14. (POST) <http://34.125.157.72:8080/mentors/join/modifyInfo>

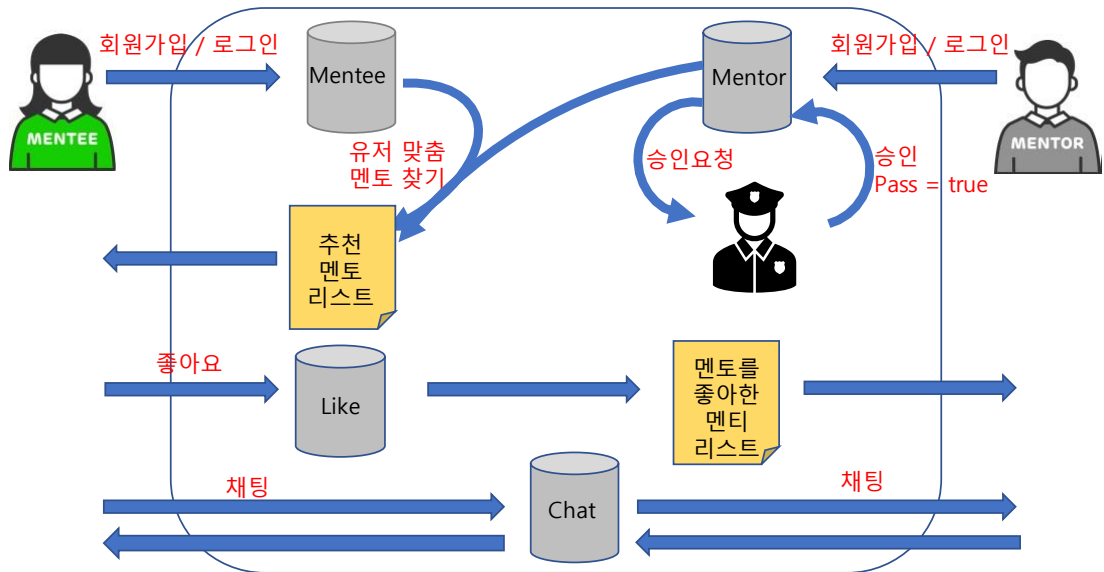
멘토의 프로필사진을 바꾸기 위한 api 입니다. 프로필만 수정하므로 나머지 값들은 이전과 동일하도록 설정했습니다.

15. (GET) <http://34.125.157.72:8080/mentors/likedList>

자신을 좋아요한 멘티들 리스트입니다. 멘토들은 이곳에서 멘티들의 정보를 확인하고 멘토링 계획을 세울 수 있습니다.

++ 모든 작업들은 데이터 송수신을 위해 여러가지 data form 을 사용하여 기능에 필요한 정보들만 담았습니다.

## + Flow



전체적인 flow는 다음과 같습니다. 멘티의 경우 회원가입을 하면 Mentee 데이터베이스에 정보가 저장되고, 그 정보를 바탕으로 추천 멘토 리스트를 보여준다. 이 때 리스트들을 카드뷰로 보면서 맘에 드는 멘토를 좋아요 할 수 있는데, 좋아요 한 멘토와 채팅 기능을 사용할 수 있는 서비스를 제공하게 됩니다. 멘토의 경우 회원가입을 하고 관리자의 승인을 기다리게 됩니다. 승인을 받고나서 서비스를 이용할 수 있게 되는데, 자신을 좋아요 한 멘티들의 리스트를 볼 수 있고, 해당 멘티들과 채팅도 할 수 있게 됩니다.

## 2.3 차별성



\* 잇다 ( 멘토들이 live class 를 열어 강의하는 서비스 )



\* 김과외 ( 학생들과 선생님들을 매칭해주는 서비스 )



\* 숨고 ( 여러 전문가들과 매칭시켜주는 서비스 )

너랑나랑은 과 유사 플랫폼들과의 차이점은 우선 앱이라는 강점도 있지만은 멘티의 조건에 맞는 멘토를 필터링해서 보여준다는 점이 주 차별성입니다. 대표적으로 *잇다*가 주관하는 형식은 강의형식과는 달리 개인간의 1 대 1 형식이다 보니 멘토링 조건들을 멘티와 멘토끼리 조정할 수 있습니다. *김과외*는 단순히 과외를 위한 학생-선생님을 매칭해줍니다. 하지만 우리는 공부에 국한되지 않고 취업 나아가 직장생활에 관련한 멘토링을 받을 수 있습니다. 마지막으로 *숨고*는 매번 번거로운 필터링 작업을 거쳐서 진행되지만 너랑나랑은 자동으로 필터링되어 보다 빠르게 서비스를 제공하는데 의의가 있습니다.

## 3. 프로젝트 수행결과

### 3.1 기대효과

우리 서비스를 이용했을 때 얻을 수 있는 기대효과는 크게 3 가지입니다.

#### 1. 서비스 이용의 자유성

다른 멘토링사이트들 같은 강의 형식이 아닌 개인끼리 매칭을 시켜주는 형식이기 때문에 공간적, 시간적인 제약에서 벗어나서 서비스를 이용할 수 있습니다. 개인간의 소통을 통해 자유롭게 멘토링 시간, 멘토링 금액 등을 정할 수 있어 시간을 잡아 여러명에게 강의하는 형식보다는 부담감이 적을 것이고, 비슷한 멘티들을 만나기 때문에 높은 수준의 멘토링을 기대할 수 있을 것입니다. 또한 맞춤 추천이기 때문에 조건에 맞는 다양한 멘토들이 추천되므로, 고스펙이 우선인 타 앱, 플랫폼보다는 멘토들의 진입 허들이 낮을 것입니다.

#### 2. 분야의 확장성

지금 우리는 취업 ,학업 등 공부와 관련된 멘토링을 주 기능으로 두었기 때문에 학교, 학점 등 공부와 관련된 정보를 처리합니다. 하지만 멘토링은 공부뿐만 아니라 다양한 분야에서도 필요성이 두드러집니다. 사람과의 화법이나 정원 관리하는 법 등 숙련된 사람들이 노하우를 알려주는 것도 멘토링의 일부이기 때문에, 추후 주거지역, 전문성 등 새로운 조건들을 제시해 더욱 멘티에게 알맞은 멘토를 추천해 줄 수 있을 것입니다.

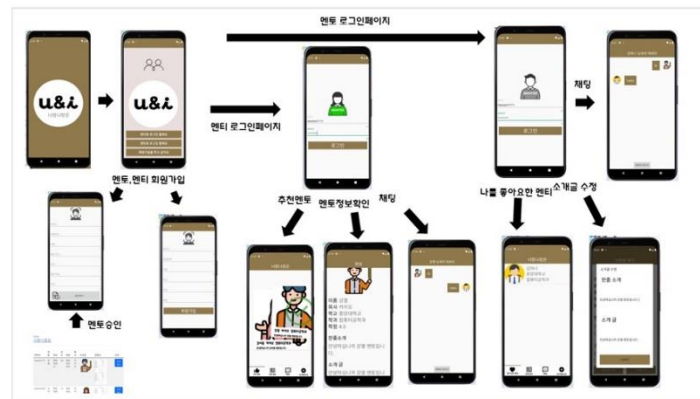
#### 3. 멘토링 효과 증가

현재 멘토링 시스템은 멘토들의 스펙을 보고 진행되는 것이 대부분입니다. 하지만 우리 앱은 멘토뿐만 아니라 멘티의 스펙과 함께 비교하여 조건에 맞는 멘토링을 진행하여 더 효율적으로 가르치고 이해하며 멘토와 멘티 모두 만족도가 증가함에 따라 좋은 결과를 기대할 수 있을 것입니다.



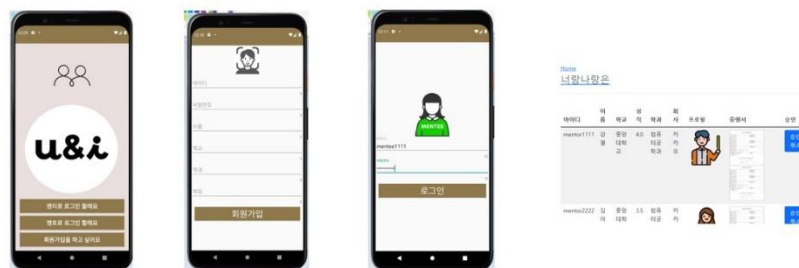
### 3.3 결과물

## 전체 UI 진행 방식



## 회원 가입 및 로그인

회원이입의 경우 알맞은 회원정보를 입력하여 진행합니다.  
멘토의 경우 증명서확인 후 승인요청 후 서비스 이용이 가능합니다.



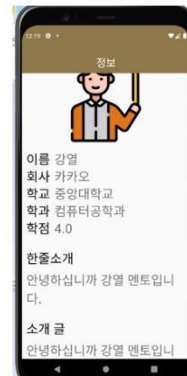
## 멘티 메인 페이지

가이드라인과 함께 앱이 시작됩니다.

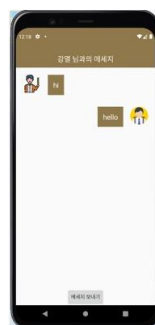
네개의 탭이 있으며 추천멘토에서 좌우로 슬라이드하여 좋아요를 선택할 수 있습니다.

또는 전체 멘토에서 자신이 직접 확인할 수 도 있습니다.

클릭 시 더욱 더 자세한 정보 확인이 가능합니다.



좋아요한 멘토에 한해서 채팅이 가능합니다. 또한 마이페이지에서 자기 정보, 좋아요한 멘토 리스트 확인할 수 있습니다.



## 멘토 메인 페이지

가이드라인과 함께 앱이 시작됩니다.

네개의 탭이 있으며 자신을 좋아하는 멘티 리스트를 확인하고 채팅을 할 수 있습니다.

또한 마이페이지에서 자기 정보 확인하고 수정할 수 있습니다.



자신을 좋아하는 멘티목록을  
확인 후 1대1 채팅이 가능합니다.



마이페이지에서 소개글 수정이  
가능합니다.

