

사용자 감성기반 음악 추천 어플리케이션

팀 BakE

박주원 박은제 이성민 이세연 이해선

CONTENTS

- 001 소프트웨어개요 및 목적
- 002 사용자정보를 이용한 곡 선택알고리즘
 - 심박수 측정 기능
 - 현재 위치에 따른 날씨 정보와 감정 선택 기능
 - 서버:노래 분류 기능
 - 추천 곡의 사용자 평가에 따른 데이터베이스 기능
- 003 기타 기능
- 004 기대효과
- 005 어플리케이션 구현 영상

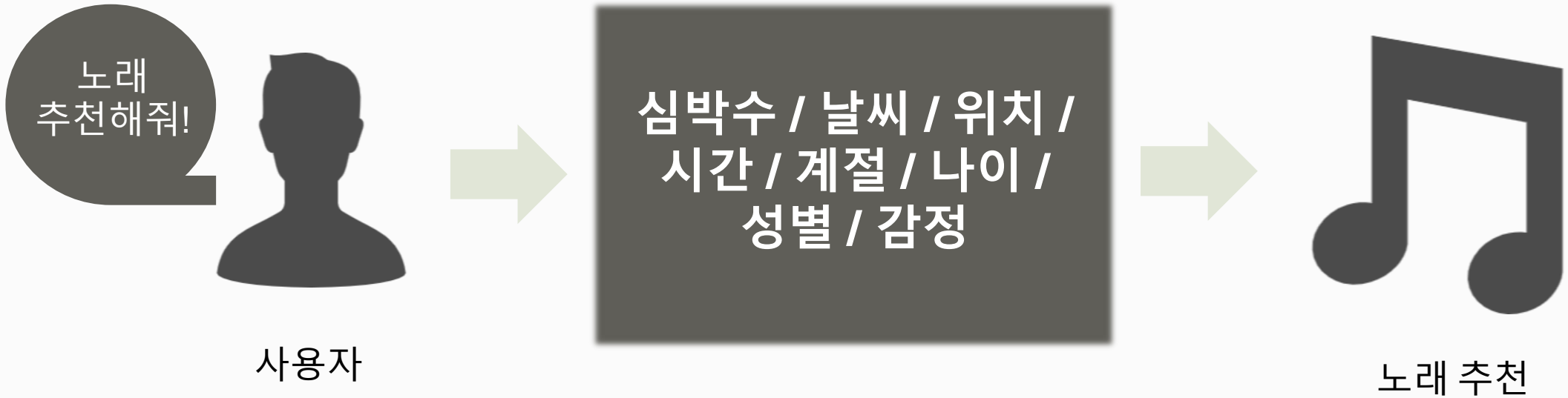


Part 1.

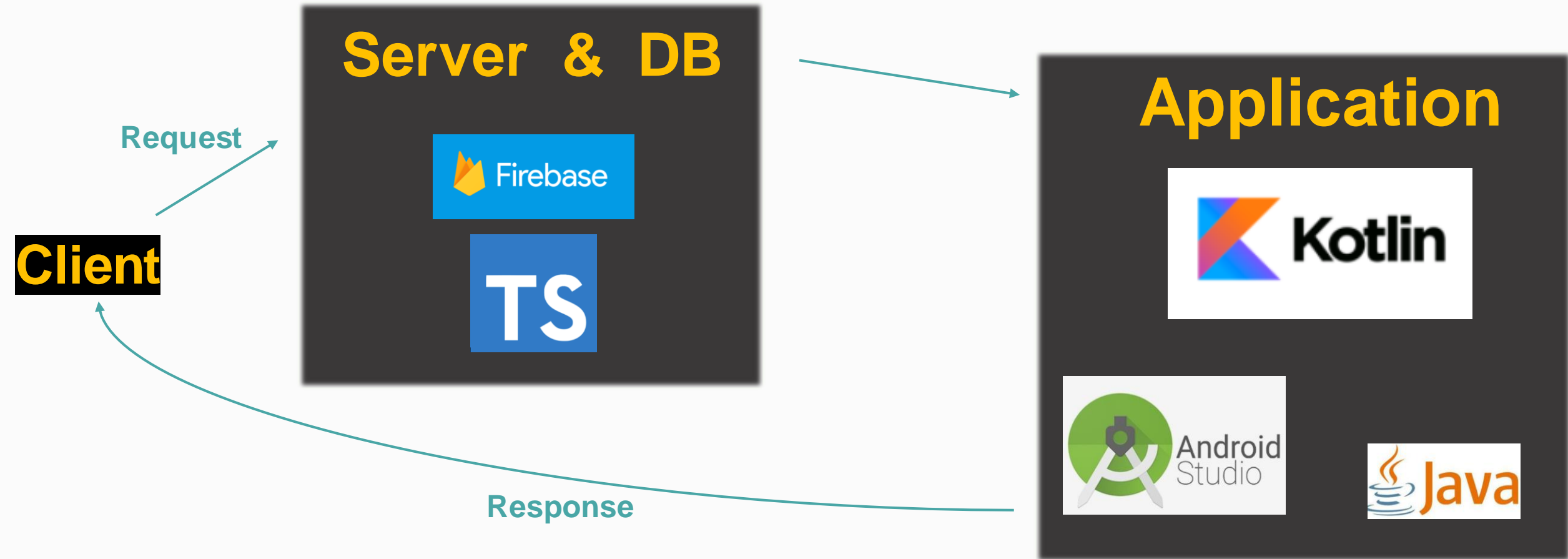
소프트웨어 개요 및 목적 목적



1.1 소프트웨어 개요 및 목적



1.2 시스템 구성도



1.3 역할 분담

박주원	심박수 측정 코드 개발	서류 작업		
박은제	Firebase로 DB 구축	서버 연결		
이세연	날씨, 위치 측정 코드 개발	액티비티 개발	UI/UX 개발	
이해선	노래추천 알고리즘 개발	액티비티 개발	서버 연결	UI/UX 개발
이성민	로그인/회원가입 기능 개발	액티비티 개발		

Part 2.

사용자 정보를 이용한 곡 선택 알고리즘



2.1 심박수 측정 기능



<심박수 측정 팝업>



<카메라 측정 화면>

2.2 현재 위치에 따른 날씨 정보와 감정 선택 기능



<홈 화면>



<감정 선택 팝업>

2.3 서버: 노래 분류 알고리즘

사용자 : (emotion, time, weather, season)
노래추천해주세요!!



Server : requestSongListWithCondition()

→ 노래추천 알고리즘으로 songList를 앱에 보내줌

BACK - END



노래 추천에 영향을 주는 요인

Condition{
 emotion
 season
 time
 weather **}**

```
const defaultCondition: Condition = {
  angry: 50,
  flutter: 50,
  funny: 50,
  happy: 50,
  sad: 50,
  soso: 50,

  sunshine: 50,
  cloudy: 50,
  rainy: 50,
  hot: 50,
  cold: 50,
  snowy: 50,

  spring: 50,
  summer: 50,
  fall: 50,
  winter: 50,
}
```

<conditions>

각 숫자는 노래와 해당
 condition 항목이
 얼마나 어울리는지를 나타냄

Ex) flutter : 90
 sunshine : 80

이고 나머지 속성은 50이라면
 이 노래는 flutter과
 sunshine에 어울리는 노래.

Total count = angry + flutter + ... sunshine+ ... +winter+ ... + morning

angry: 50

cloudy: 50

cold: 50

dawn: 50

evening: 50

fall: 50

flutter: 50

funny: 50

happy: 50

hot: 50

lunch: 50

<conditions>

autoNormalizing()

시스템의 부하를 막기 위해
Condition 값에 변화가
생긴 값에 대해서만 실행하여
Normalized_conditions db
항목에 변경된 값만 업데이트

angry: 0.05

cloudy: 0.05

cold: 0.05

dawn: 0.05

evening: 0.05

fall: 0.05

flutter: 0.05

funny: 0.05

happy: 0.05

hot: 0.05

lunch: 0.05

morning: 0.05

<normalized_conditions>

normalized_angry = angry / total count

normalized_cloudy = cloudy / total count

⋮

⋮

⋮

노래 한 곡에 대한
Normalized_Condition
총 합 : 1

“ 여러 곡들의 condition 속성의
상대적 크기 비교가능 ”

사용자 요청 : flutter, spring, morning, sunshine

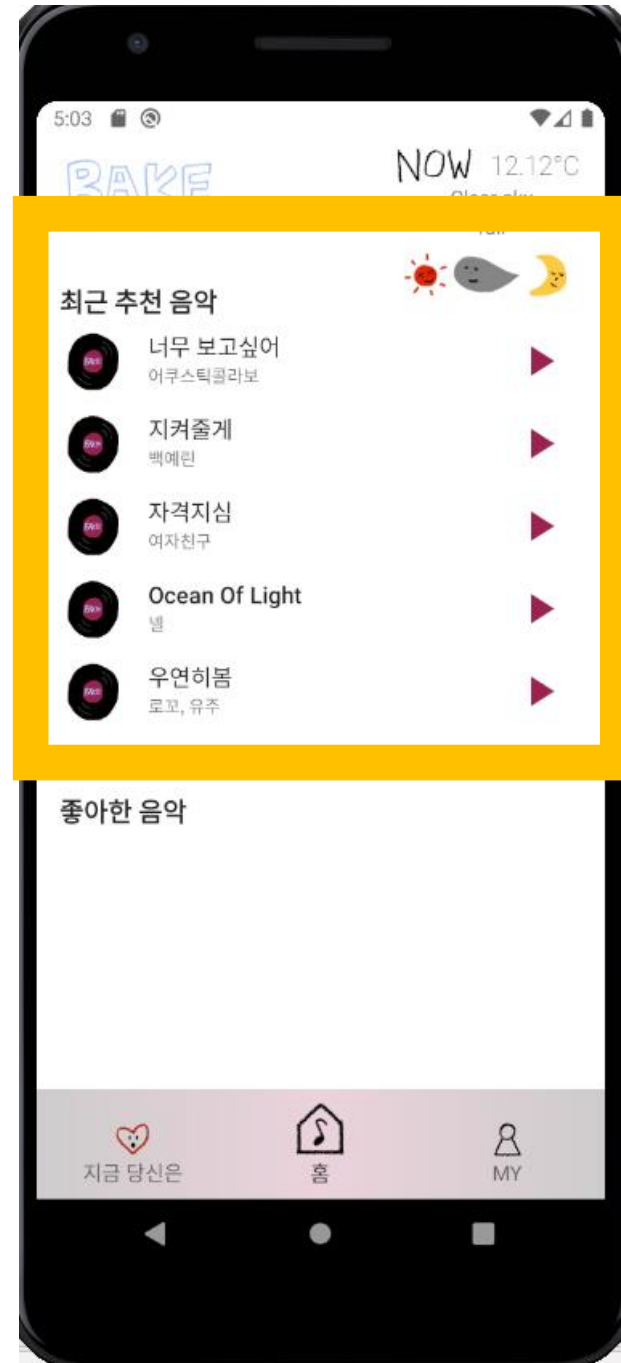
노래1 Normalized_condition	노래2 Normalized_condition	노래3 Normalized_condition	...	Normalized_condition
flutter : 0.1 Spring : 0.3 Morning : 0.2 Sunshine : 0.2	flutter : 0.01 Spring : 0.05 Morning : 0.1 Sunshine : 0.15	flutter : 0.3 Spring : 0.2 Morning : 0.01 Sunshine : 0.13	...	flutter : 0.3 Spring : 0.2 Morning : 0.01 Sunshine : 0.13

flutter 항목 비교 후 점수가 가장 높은 곡 10곡 추출
 Spring 항목 비교 후 점수가 가장 높은 곡 3곡 추출
 morning 항목 비교 후 점수가 가장 높은 곡 3곡 추출
 sunshine 항목 비교 후 점수가 가장 높은 곡 4곡 추출



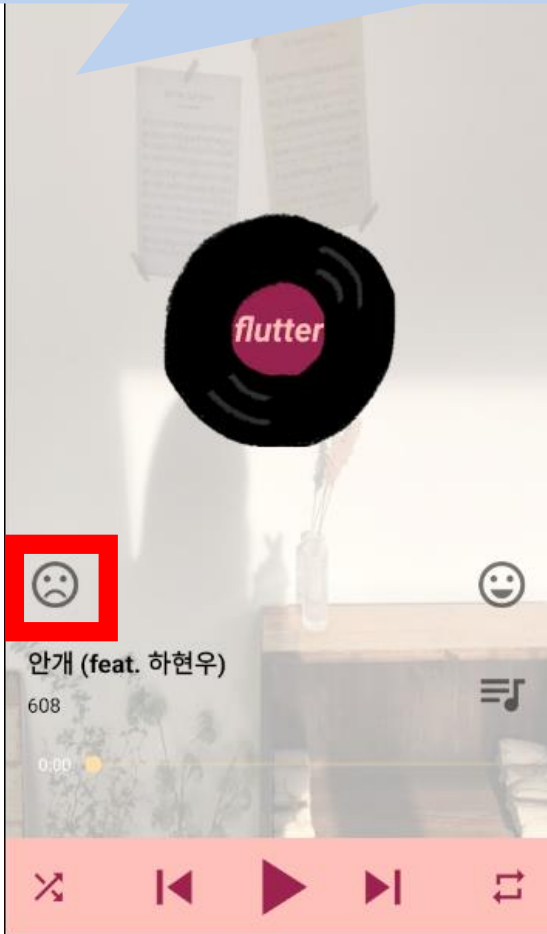
이렇게 뽑힌 곡 중
 랜덤으로 5곡 추천

사용자 요청 : flutter, spring, morning, sunshine



2.4 사용자의 평가에 기반한 실시간 DB 업데이트

사용자 요청 : flutter, spring, morning, sunshine



decreaseCondition()



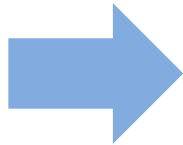
```
condition[requestCondition.emotion] -= 1
condition[requestCondition.weather] -= 1
condition[requestCondition.season] -= 1
condition[requestCondition.time] -= 1
```

데이터베이스의
flutter, spring, morning, sunshine
의 점수가 -1

condition 변화가
생길 때마다 자동으로
autoNormalizing() 실행

2.4 사용자의 평가에 기반한 실시간 DB 업데이트

angry: 30
cloudy: 50
cold: 50
dawn: 50
evening: 50
fall: 50
flutter: 70
funny: 50
happy: 50
hot: 50
lunch: 50
morning: 50



사용자 반응에 대한 실시간 대응

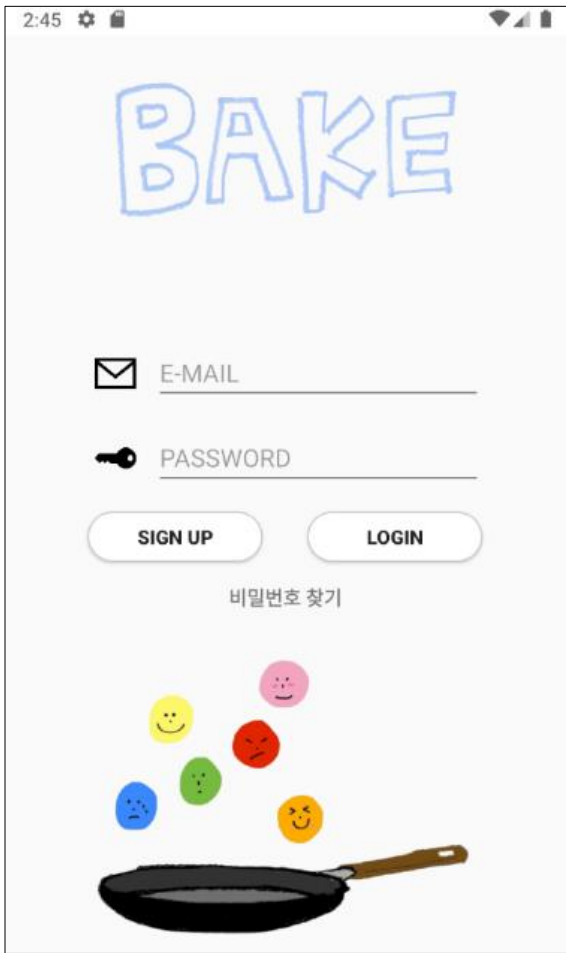
사용자의 입력이 쌓일 수록 데이터의 정확성 상승

Part 3.

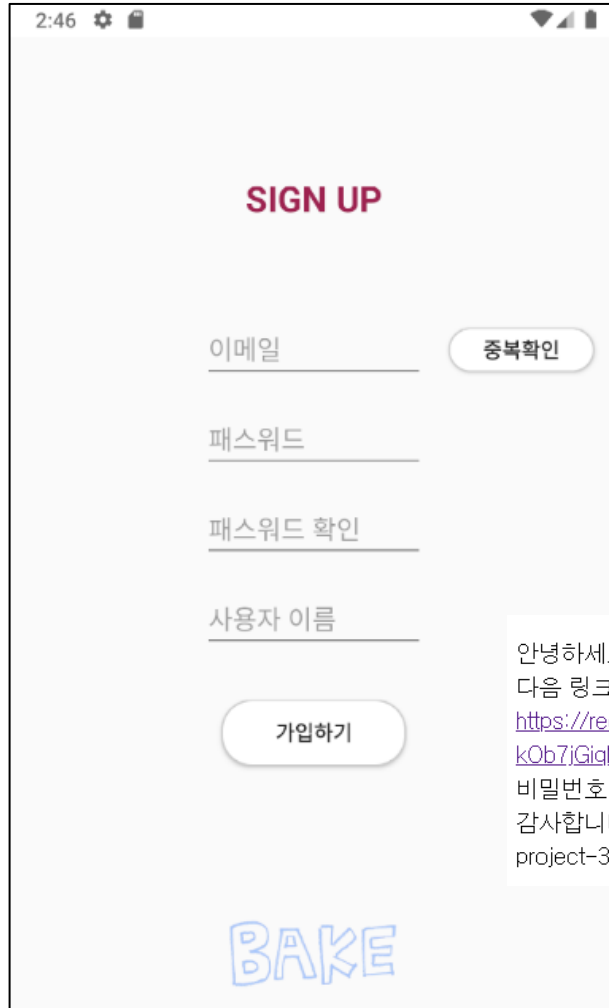
기타기능기능



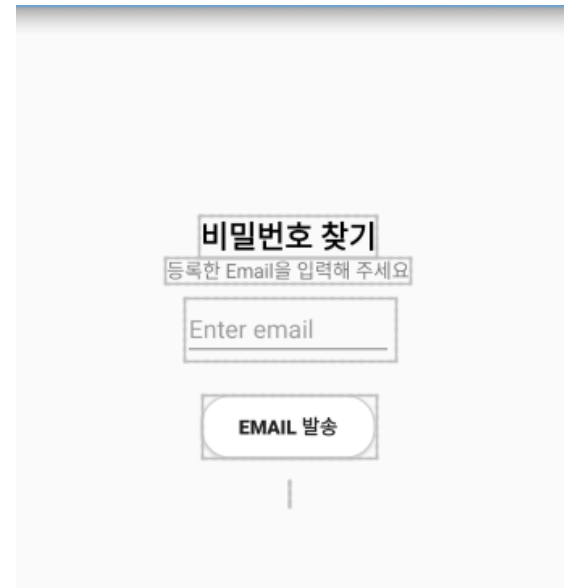
3.1 로그인 기능 / 회원가입 기능



<로그인 화면>



<회원가입 화면>



안녕하세요.

다음 링크를 통해 gildogi@naver.com 계정의 project-312988178575 비밀번호를 재설정하세요.

https://recommended-music.firebaseio.com/_/auth/action?mode=resetPassword&oobCode=NBiWWD8Zy8Ly6N0_loQmopAS_P4Ld-kOb7jGigMB9k4AAAF0rsLwrQ&apiKey=AlzaSyAfwXuJ6-g4JoF9i5RR4Nu79CKIn32Ylac&lang=ko

비밀번호 재설정을 요청하지 않았다면 이 이메일을 무시하셔도 됩니다.

감사합니다.

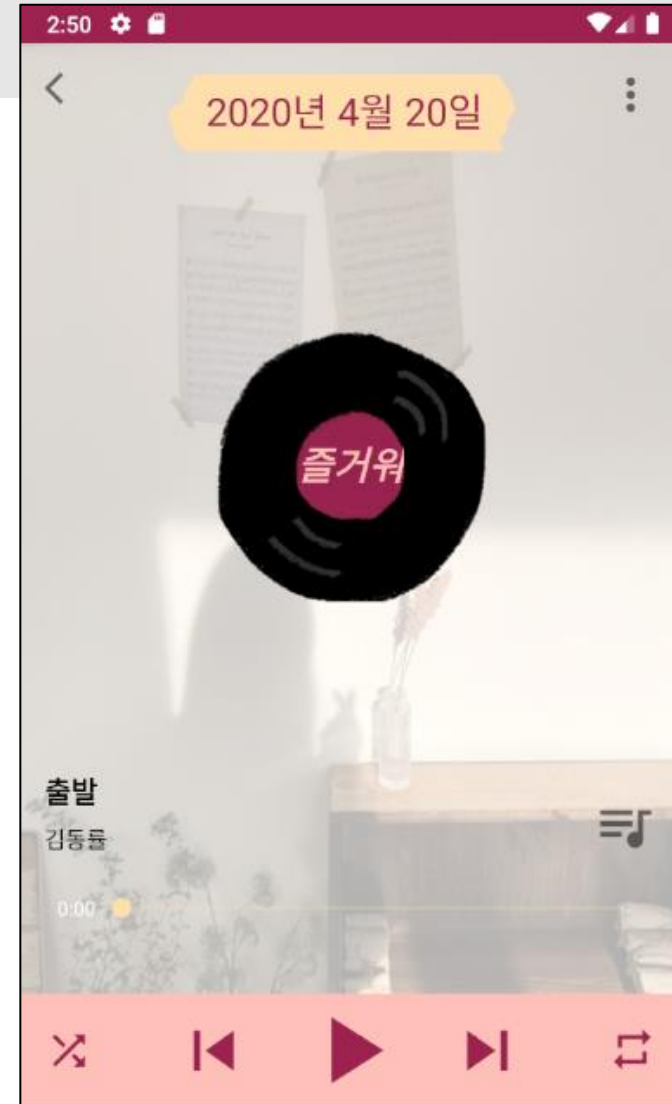
project-312988178575팀

<비밀번호 찾기 화면>

3.2 계정을 통한 노래 관리 기능



<사용자 피드 화면>



<노래 재생 화면>

Part 4.

기대효과



4.1 기대효과

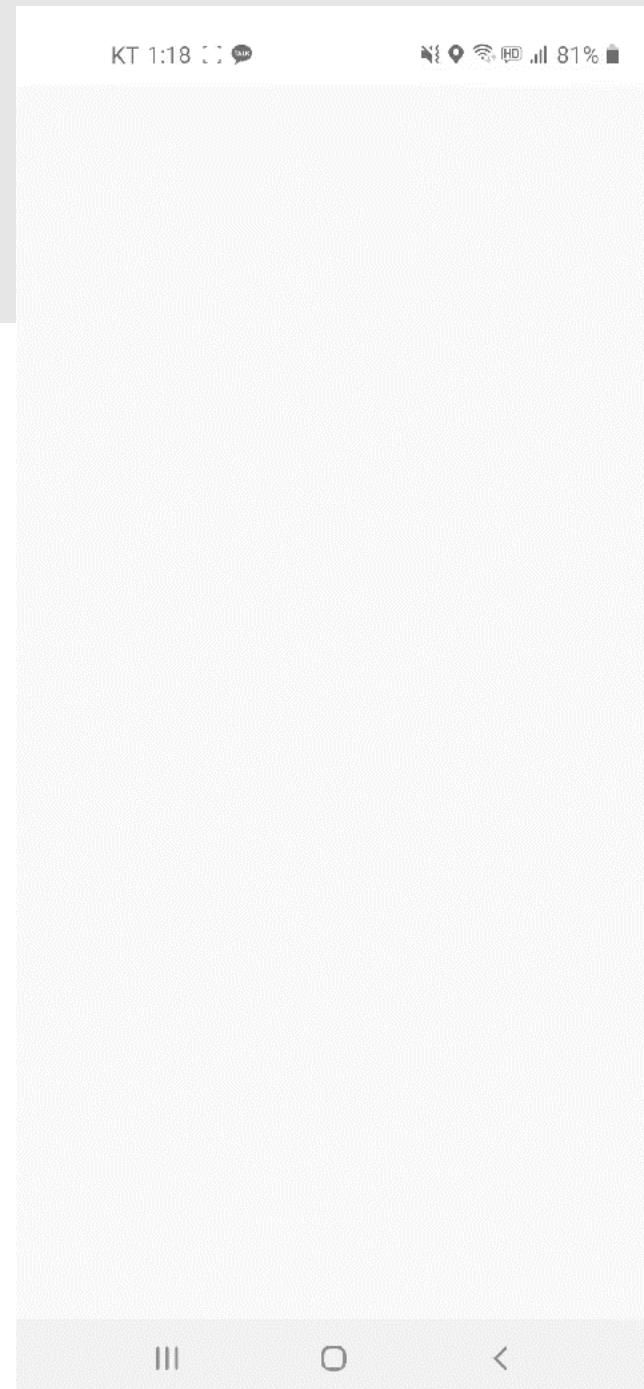
- **사용자의 현재 상태를 기반으로 한 맞춤 노래 추천과 이에 상응하는 감각적인 인터페이스를 통해 사용자의 만족도 및 서비스를 향상할 수 있습니다.**
- **관리자가 직접 분석한 데이터를 통해 개발한 음악 추천 알고리즘으로 인한 더 섬세한 효과를 기대할 수 있습니다.**

Part 5.

어플리케이션 구현 영상 영상



5. 어플리케이션 시연



감사합니다

BAKE