13주차 진행 보고서 세부 사항

2021.05.13~ 2021.05.26 B71116장혜리

코드는 아래 gitnub에서 확인 가능합니다.

[**[github]MBTI-Lovers 안드로이드**](https://github.com/hyerijang/MBTI-Lovers)

[**[github]MBTI-Lovers-webServer**](https://github.com/hyerijang/MBTI-Lovers-webServer)

목차

[UX/UI 디자인 1](#_Toc72931750)

[친구목록 3](#_Toc72931751)

[친구 신청, 차단 기능 구현 3](#_Toc72931752)

[안드로이드 3](#_Toc72931753)

[권한 요청 팝업 구현 3](#_Toc72931754)

# UX/UI 디자인

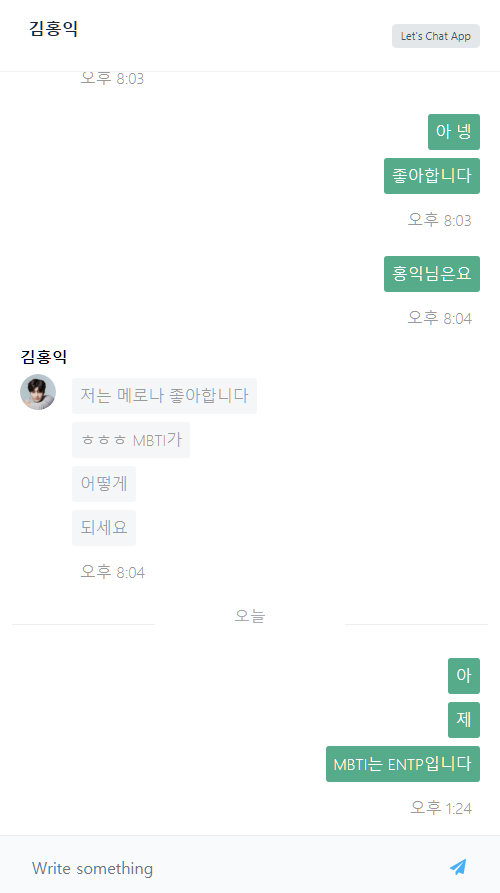
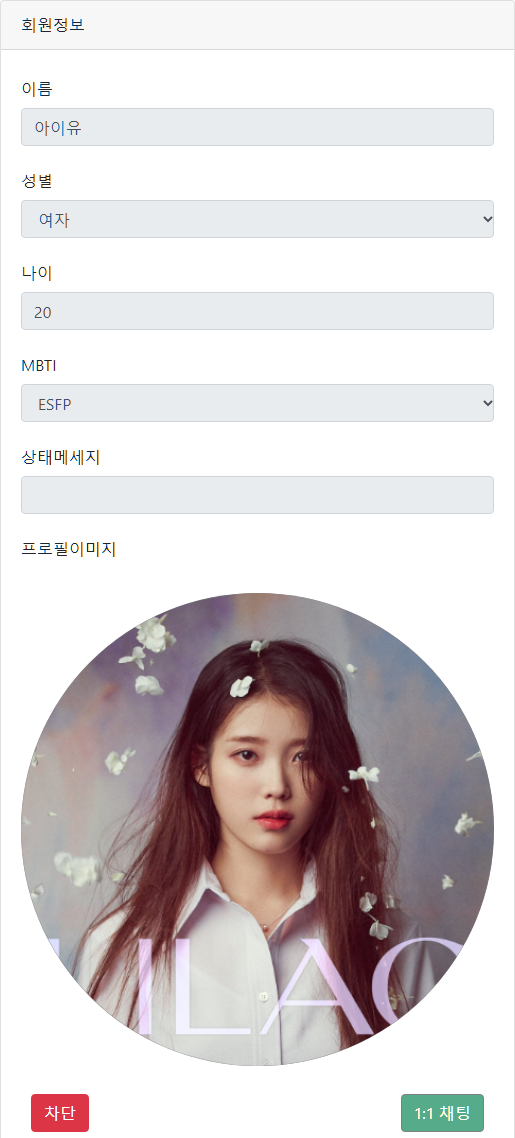
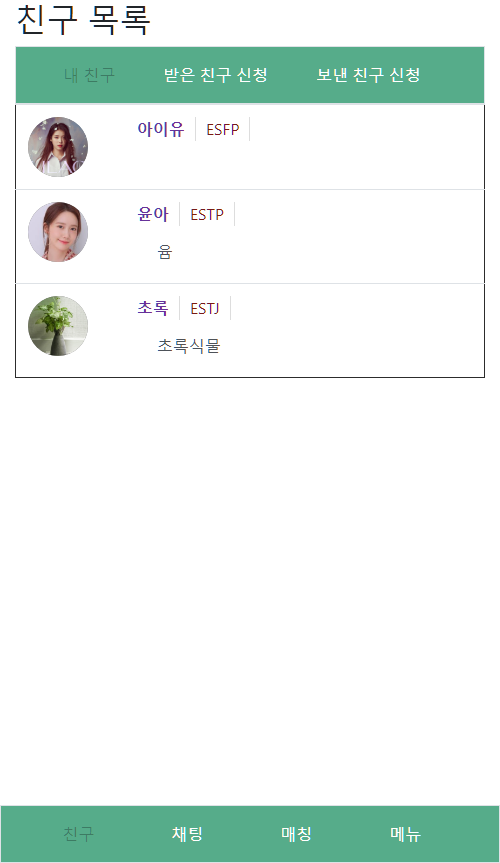
## 팝업 구현

매칭 시 위치 정보를 가져올 수 없는 경우 경고 팝업을 띄웁니다. 유저 마커 클릭 시 유저 정보 팝업을 띄웁니다.

|  |  |
| --- | --- |
| Figure 위치 서비스 경고 팝업 | Figure 유저 정보 팝업 |

## 전체적인 css 수정

채팅 어플이라는 디자인 컨셉에 맞게 UX/UI 디자인을 전반적으로 수정하였습니다. 채팅 시에는 시간 메타데이터를 기준으로 **날짜 및 보낸 시각**(시, 분) 기준으로 메시지가 그룹핑되어 보여집니다.



# 친구목록

## 친구 신청, 차단 기능 구현

친구 관계를 세분화하여 친구 신청 및 차단 기능을 구현하였습니다. Friend컬럼을 삭제할 경우 *친구 유저와의 채팅 기록을 조회할 수 없는 문제*가 발생하기 때문에 차단을 해제하는 경우 컬럼을 그대로 유지하고 *NONE*이라는 관계로 변경하도록 구현하였습니다.

public enum RelationType *{  
  
 FRIEND(*"친구"*)*,  
 *REQUEST(*"신청"*)*,  
 *RECEIVED\_REQUEST(*"신청받음"*)*,  
 *BLOCK(*"차단"*)*,  
 *BLOCKED(*"차단됨"*)*,  
 *NONE(*"아무 관계 아님"*)*;  
 *}*

또한 매칭 시 *친구 유저와 차단된 유저를 제외*하고 보여주기 위해 SQL 쿼리를 새롭게 작성하였습니다.

@Query*(*value = "SELECT r.\* FROM (SELECT m.\* \tFROM member m \tleft join friend f \ton f.uid = ?4 and m.uid = f.fid where f.uid is null or f.relation not in (\"FRIEND\",\"BLOCK\")) as r where r.location is not null order by ST\_Distance(location, POINT(?1, ?2)) limit ?3", nativeQuery = true*)*List*<*Member*>* findNearPointNotFriend*(*double x, double y, int n, String uid*)*;

# 안드로이드

## 권한 요청 팝업 구현

저장소 및 위치권한이 허용되지 않으면 정상적인 실행이 불가능하기 때문에 앱 실행 시 해당 권한들을 요청합니다. 만약 허용되지 않은 권한이 있으면 로그인 되지 않고, 권한 허용을 위한 팝업이 뜹니다. 이 경우 유저가 수동으로 권한을 허용해야하므로 앱 정보 창으로 이동니다. 멀티 퍼미션 라이브러리인 **Dexter**를 이용하여 구현하였습니다.

|  |  |  |
| --- | --- | --- |
| Figure 3최초 권한 요청 | Figure 4 권한 거부 후 로그인 | Figure 앱 정보 창 |