

Graduation Project

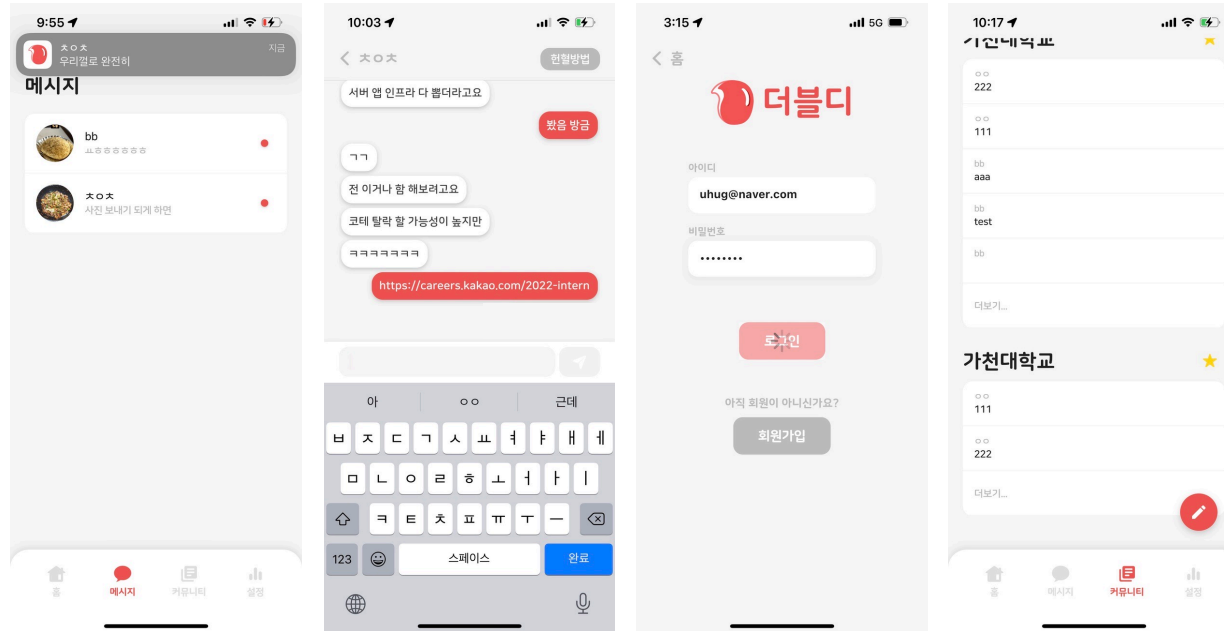
Class 2, Group 5 Final Presentation
2022.04.19

201935006 고효진
201533631 김도균
201935144 최윤찬

I . Project Progress

Front-end

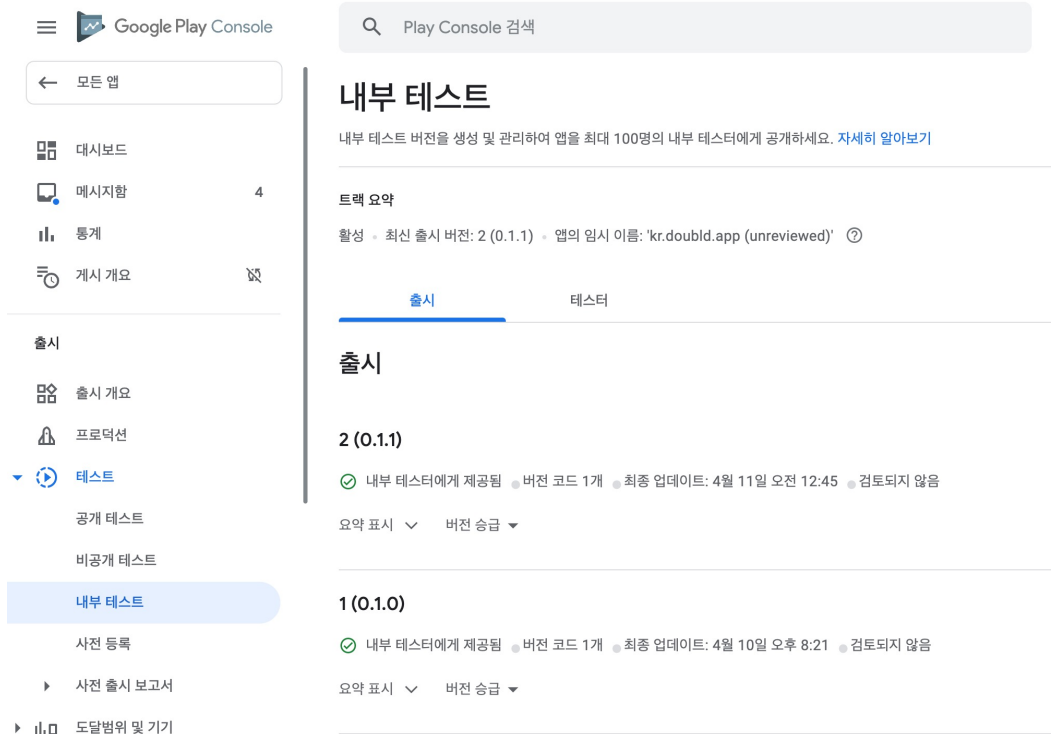
```
✓ lib
  ✓ model
    association.dto.dart
    fcm.dto.dart
    person.dto.dart
    post.dto.dart
    token.dto.dart
  ✓ util
    > chat
    > network
    > noti
    > theme
    global_variables.dart
    preference_manager.dart
    secret.dart
  ✓ view
    > community
    > home
    > signup
    message_view.dart
    setting_text_view.dart
    signin_view.dart
    splash_view.dart
  ✓ widget
    app_bar.dart
    bottom_navigation.dart
    button.dart
    community_item.dart
    dropdown_button.dart
    input_box.dart
    page_title_widget.dart
    generated_plugin_registrant.dart
    main.dart
```



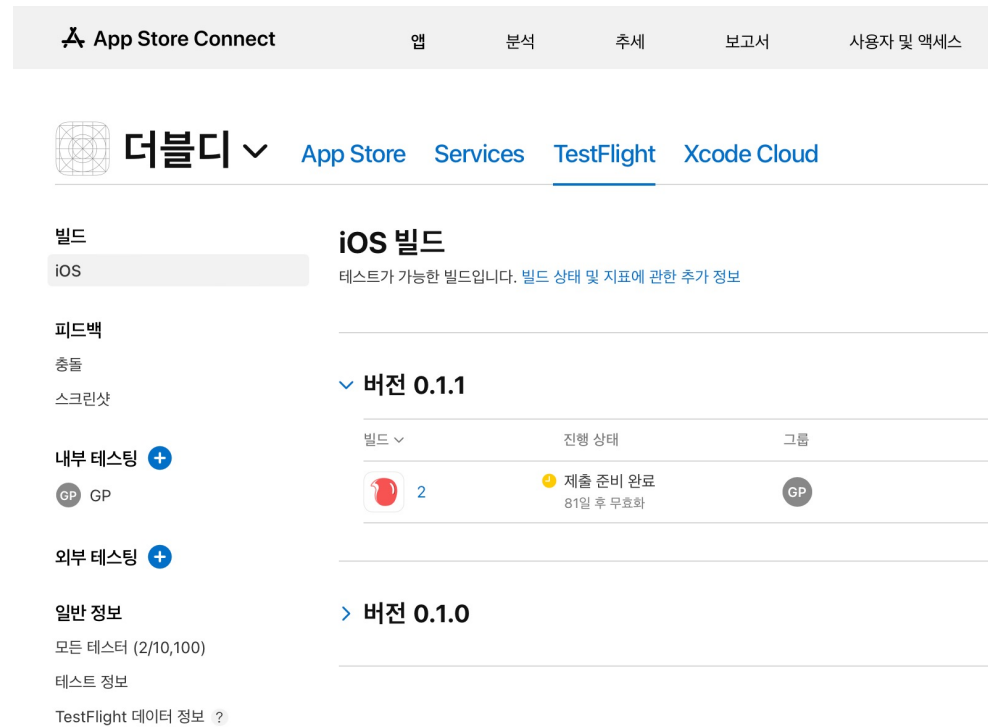
- Fix design system
- Back-end API Communication (Assign, Community)
- State management in local
- Chatting service (through Firebase FireStore and FCM)

Front-end

- Internal Test Deploy
- Available on Android and iOS



The screenshot shows the Google Play Console interface. On the left is a navigation menu with options like '대시보드', '메시지함', '통계', '게시 개요', '출시', '출시 개요', '프로덕션', '테스트', '공개 테스트', '비공개 테스트', '내부 테스트', '사전 등록', '사전 출시 보고서', and '도달범위 및 기기'. The main content area is titled '내부 테스트' (Internal Test) and shows details for version 2 (0.1.1). It includes a '트랙 요약' (Track Summary) section with a status of '완료' (Completed) and a '출시' (Release) section with a status of '완료' (Completed). The version 2 (0.1.1) is marked as '완료' (Completed) and shows a status of '완료' (Completed) with a note '내부 테스터에게 제공됨' (Provided to internal testers).



The screenshot shows the App Store Connect interface. The top navigation bar includes 'App Store Connect', '앱' (App), '분석' (Analytics), '추세' (Trends), '보고서' (Reports), and '사용자 및 액세스' (Users and Access). The main content area is titled 'iOS 빌드' (iOS Build) and shows details for version 0.1.1. It includes a '피드백' (Feedback) section, a '충돌' (Conflict) section, and a '스크린샷' (Screenshot) section. The version 0.1.1 is marked as '완료' (Completed) and shows a status of '완료' (Completed) with a note '테스트가 가능한 빌드입니다' (Build is testable).

Back-end

```

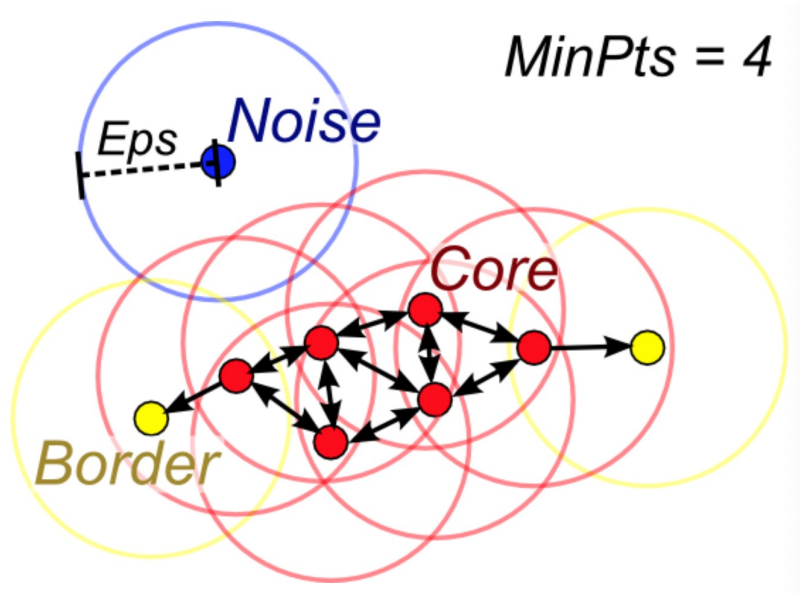
  ✓ blood-donation
    ✓ user
      > GET getUser
      > GET getCurrentUserInfo
      > PATCH updateUser
      > POST changeDormant
      > POST createUserSns
      > GET getUserSnsList
      > DEL deleteUserSns
      > POST validateDuplicate
      > POST changePassword
      > POST lostPassword
      > POST validateEmail
      > POST validateEmailSendCode
    > file
    > post
    > user-associations
    > associations
    > auth
    > liked
    > recommend
    > FLASK
    POST batch

```

- Implement additional functions for FE
- Fix and modify several APIs and ERD
- Add "Likes" feature to collect training dataset
- Add recommendation service through Flask recommendation server

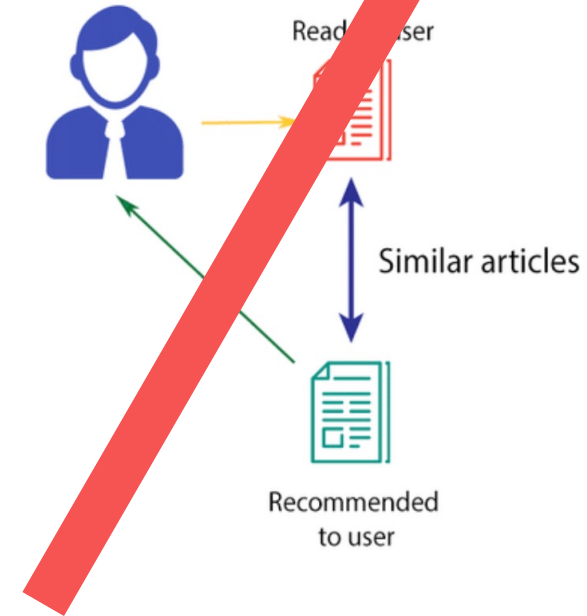
ML (Recommendation Server)

- DBSCAN Clustering



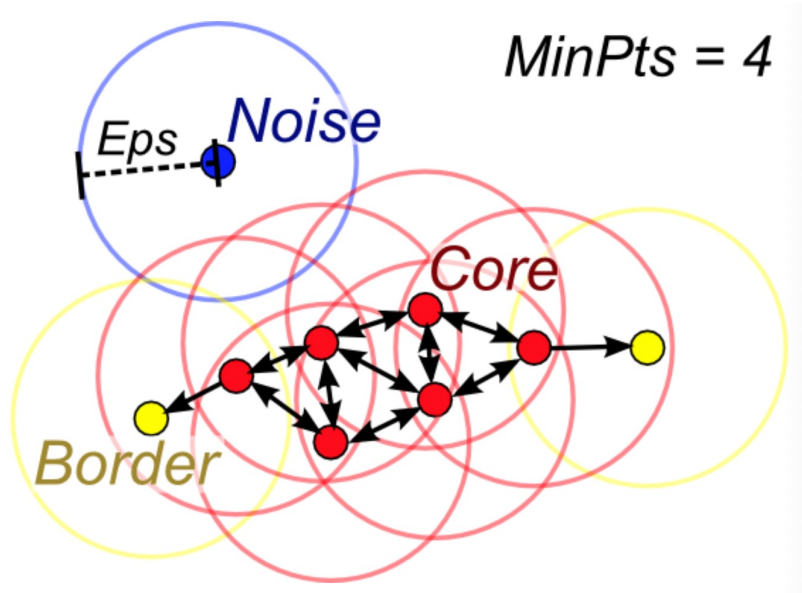
<https://p829911.github.io/2019/01/11/DBSCAN/>

- Content-based filtering



ML (Recommendation Server)

- DBSCAN Clustering



<https://p829911.github.io/2019/01/11/DBSCAN/>

New Solution

- Calculate Euclidian distance in target cluster

ML (Recommendation Server)

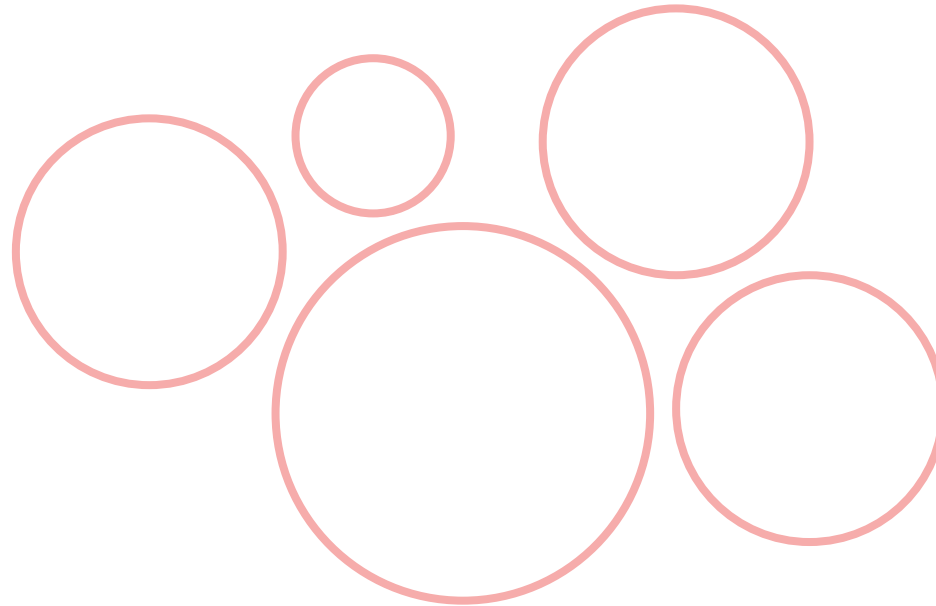
0. DBSCAN Clustering

1. Find cluster of target user
2. Find cluster of "Liked" user who the target user does.
3. Flatten all data from clusters
4. Calculate Euclidian distance and make recommendation list.

ML (Recommendation Server)

0. DBSCAN Clustering

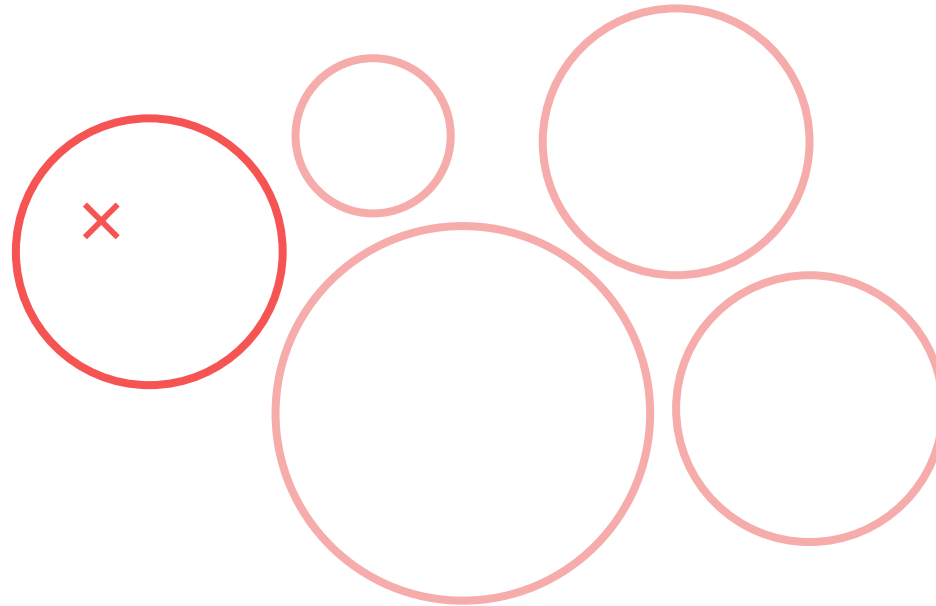
1. Find cluster of target user
2. Find cluster of "Liked" user who the target user does.
3. Flatten all data from clusters
4. Calculate Euclidian distance and make recommendation list.



ML (Recommendation Server)

0. DBSCAN Clustering

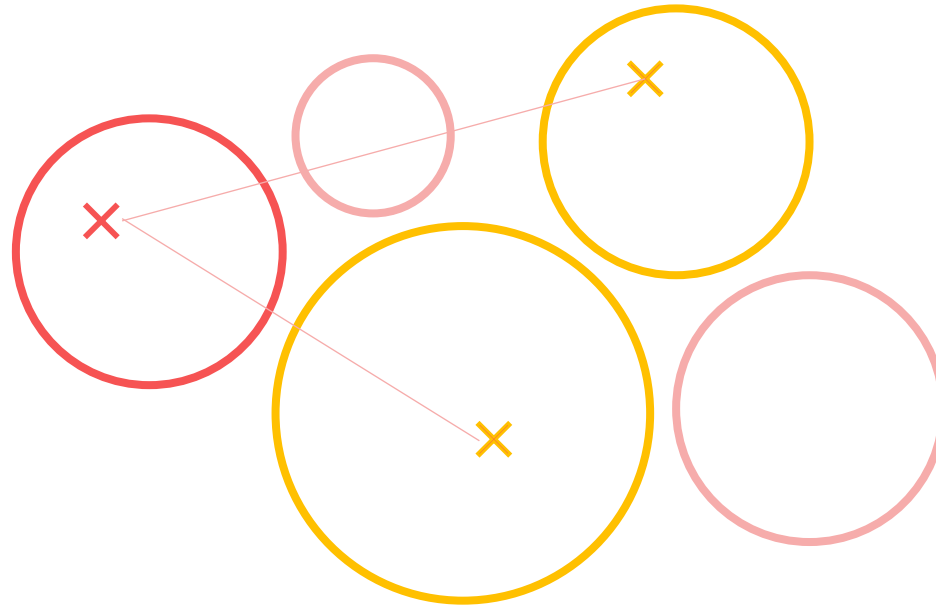
1. Find cluster of target user
2. Find cluster of "Liked" user who the target user does.
3. Flatten all data from clusters
4. Calculate Euclidian distance and make recommendation list.



ML (Recommendation Server)

0. DBSCAN Clustering

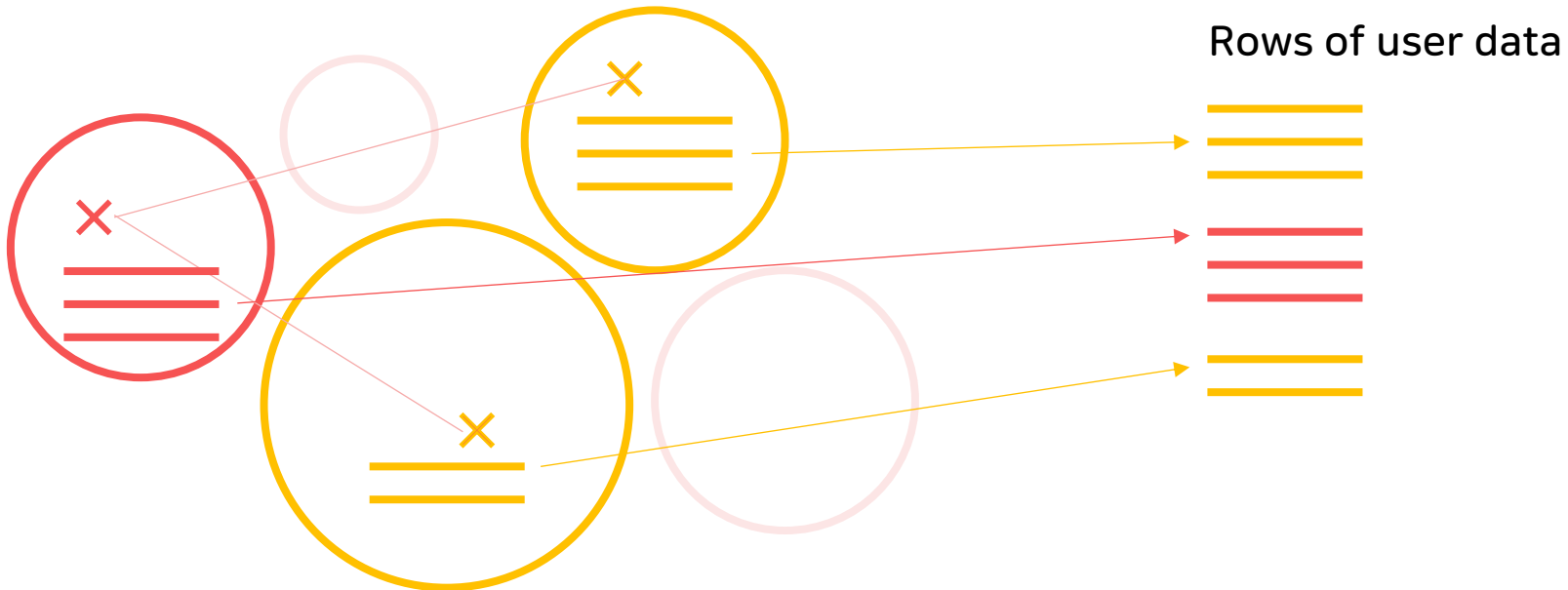
1. Find cluster of target user
2. Find cluster of "Liked" user who the target user does.
3. Flatten all data from clusters
4. Calculate Euclidian distance and make recommendation list.



ML (Recommendation Server)

0. DBSCAN Clustering

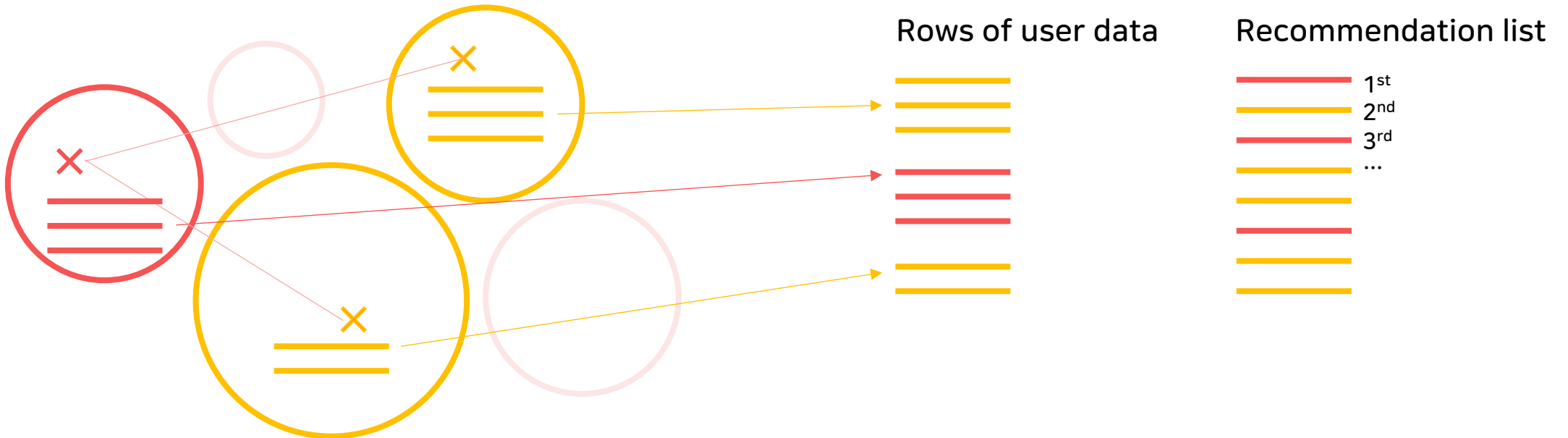
1. Find cluster of target user
2. Find cluster of "Liked" user who the target user does.
3. Flatten all data from clusters
4. Calculate Euclidian distance and make recommendation list.



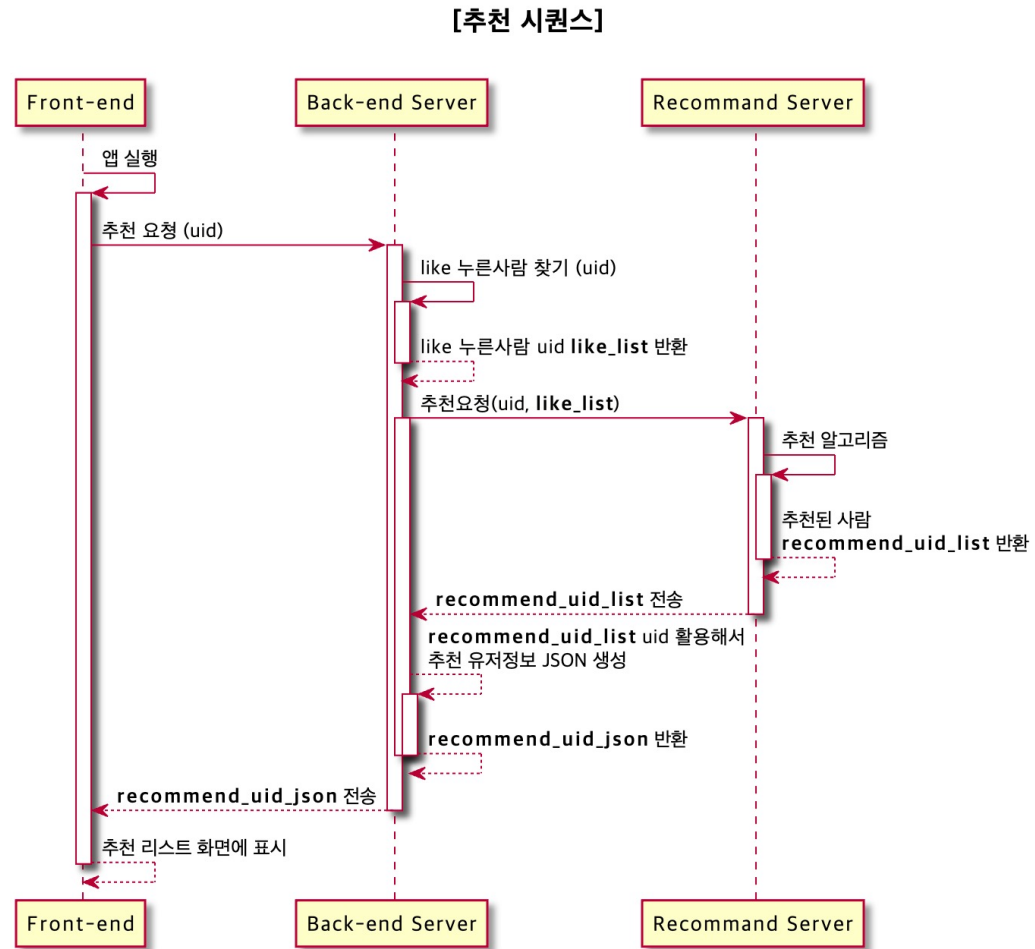
ML (Recommendation Server)

0. DBSCAN Clustering

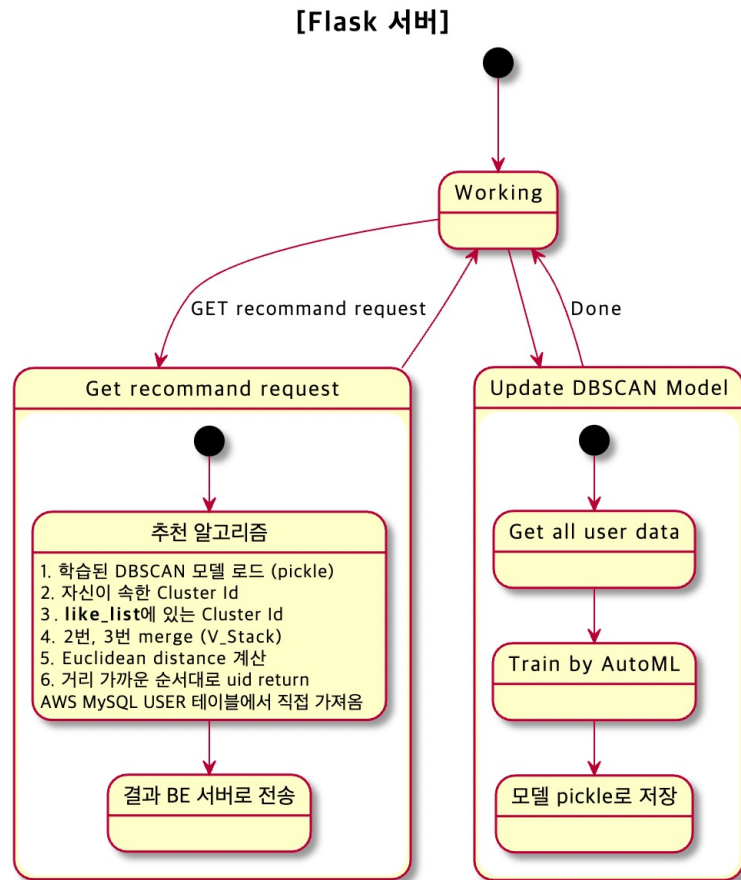
1. Find cluster of target user
2. Find cluster of "Liked" user who the target user does.
3. Flatten all data from clusters
4. Calculate Euclidian distance and make recommendation list.



ML (Recommendation Server)



ML (Recommendation Server)



- Implement tangible recommendation sequence
- Connect Flask server with main service server (Spring)
- Ready for test in practical field

Graduation Project

Class 2, Group 5 Final Presentation
2022.04.19

201935006 고효진
201533631 김도균
201935144 최윤찬