

DB 프로젝트 4

2019019016 서시연

1. 코드 설명

- main

- initDBConnection()을 호출한다.
- 아래 네 가지 메뉴가 있다.
 1. Log in
 - login()을 호출한다.
 2. Sign up
 - signUp()을 호출한다.
 3. Find ID / Password
 - findAccount()를 호출한다.
 4. Exit
 - exit를 true로 만들어 while문을 빠져나가고, 프로그램을 종료한다.

- initDBConnection()

- DB_URL을 jdbc:mariadb//localhost:3306/musicssystem으로 설정한다.
- Class.forName(org.mariadb.jdbc.Driver)를 해준다.
- static Connection connection에 DriverManager.getConnection(DB_URL, "root", [password])에서 리턴된 값을 저장한다.

- login()

1. 아이디와 비밀번호를 입력받는다.
2. 입력받은 아이디,비밀번호를 갖는 manager가 존재한다면 currentUserNo를 해당하는 매니저의 managerNo로 설정한뒤, boolean manager를 true로 설정하고, managerMenu()를 호출한다.
3. 입력받은 아이디,비밀번호를 갖는 user가 존재한다면 currentUserNo를 해당하는 user의 userNo로 설정한뒤, boolean manager를 false로 설정하고, userMenu()를 호출한다.
4. 매니저와 유저 모두에 일치하는 아이디, 비밀번호를 갖는 사람이 없다면 while문을 통해 아이디와 비밀번호를 재입력받는다.
5. 아이디에 -1을 입력하면 while문을 빠져나간다.

- signUp()

- manager 또는 user로 회원가입하는 메소드.
- 1. manager 여부를 입력받아 isManager에 저장한다.
- 2. 이름, 생년월일, 성별을 입력받아 각각 name, bDate, gender에 저장한다.
- 3. 아이디를 입력받는다.
 - 1) isExistId를 true로 초기화해준다.
 - 2) ID를 입력받고 입력받은 id를 갖는 manager나 user가 있는지 확인한다.
 - 3) 만약 해당 아이디가 이미 있거나 -1을 입력한 경우 while문의 처음으로 돌아가 아이디를 다시 입력 받는다

- 로그인시 루프를 빠져나갈 때 아이디에 -1을 입력해서 빠져나가므로, -1을 아이디로 갖지 못하게 한다.
- 4) 아이디가 존재하지 않는다면 while문을 빠져나간다.
- 4. 비밀번호를 입력받는다.
 - 1) 비밀번호를 입력받은 후 확인을 위해 한 번 더 입력받는다.
 - 2) 입력된 두 개의 비밀번호가 서로 같은 경우 while문을 빠져나간다.
 - 3) 비밀번호가 서로 다른 경우 while문의 처음으로 돌아가 다시 입력받는다.
- 5. managerNo를 arraylist인 managerNo에 저장한다.
- 6. 유저로 가입하는 경우
 - 1) 매니저의 수가 0명이면 매니저가 존재하지 않는다는 메시지를 띄우고 회원가입을 종료한다.
 - 2) managerNo에서 랜덤으로 managerNo를 뽑아서 새롭게 가입하는 user의 담당 관리자로 설정한다.
 - 3) 입력받은 정보들을 갖는 user를 추가한다.
- 7. 매니저로 가입하는 경우 입력받은 정보들을 갖는 manager를 추가한다.

- findAccount()

- ID를 찾거나 비밀번호를 변경하는 메소드
- 아래 세 가지 메뉴가 있다.
 1. Find ID
 - 매니저 여부, 이름, 생년월일, 성별을 입력받는다.
 - 매니저인 경우 String user에 “manager”를, 유저인 경우 “user”를 저장한다.
 - 입력받은 정보와 일치하는 정보를 가진 유저/매니저가 존재한다면 ID를 출력하고, 존재하지 않는다면 그런 사람은 존재하지 않는다는 메시지를 출력한다.
 2. Change Password
 - Find ID와 마찬가지로 매니저여부, 이름, 생년월일, 성별을 입력받고, 추가로 ID도 입력받는다.
 - 입력받은 정보와 일치하는 정보를 가진 유저/매니저가 존재하지 않는다면 존재하지 않는다는 메시지를 출력하고 종료한다.
 - 일치하는 유저가 존재한다면 password를 재설정한다.
 3. Go back
 - while문을 빠져나가 함수를 종료한다.

- getResult(sql)

- executeQuery를 호출해 sql에 저장된 쿼리를 실행해서 그 결과를 리턴해준다.

<Manager>

- managerMenu()

- 아래 7 가지 메뉴가 있다.
 1. Manage Musics
 - manageMusics()를 호출한다.
 2. Manage Users
 - manageUsers()를 호출한다.
 3. Manage artists
 - manageArtists()를 호출한다.

4. Manage albums

- manageAlbums()를 호출한다.

5. Edit my Info

- showUserDetail(-1)을 호출한다.

6. Withdrawal

- 정말 탈퇴할 것인지 확인하고, 탈퇴하는 경우 deleteManager()를 호출한다.
- deleteManager에서 리턴되는 값을 withdrawal에 저장한다.
- withdrawal이 1인 경우 while문을 빠져나간다.

7. Logout

- exit를 true로 바꾸고 while문을 빠져나간다.

- manageMusics()

- 아래 4가지 메뉴가 있다.

1. Show music chart

- showMusicChart()를 호출한다.

2. Search music

- searchMusic()을 호출한다.

3. Add new music

- addMusic(-1,-1)을 호출한다.

4. Go back

- exit를 true로 바꾸고 while문을 빠져나간다.

- manageUsers()

1. 전체 유저 목록 (ID + 이름) 을 출력한다.

- 유저가 없는 경우 없다는 메시지를 출력하고, 엔터를 누르면 함수를 종료한다.

2. 관리할 유저를 입력받고, 해당 유저의 userNo를 userNo에 저장한다.

3. showUserDetail(userNo)를 호출한다.

- manageArtists()

- 아래 3가지 메뉴가 있다.

1. Show all artist

- showAllArtist()를 호출한다.

2. Add new artist

- addArtist()를 호출한다.

3. Go back

- exit를 true로 바꾸고 while문을 빠져나간다.

- manageAlbums()

- 아래 3가지 메뉴가 있다.

1. Show all albums

- showAllAlbums()를 호출한다.

2. Add new album

- addAlbum(-1)를 호출한다.

3. Go back

- exit를 true로 바꾸고 while문을 빠져나간다.

<User>

- **userMenu()**

- 아래 7가지 메뉴가 있다.
- 1. Show music chart
 - showMusicChart()를 호출한다.
- 2. Search music
 - searchMusic()을 호출한다.
- 3. Show all public playlists
 - showAllPublicPlayList()를 호출한다.
- 4. Show All Albums
 - showAllAlbums()를 호출한다.
- 5. Show All Artists
 - showAllArtist()를 호출한다.
- 6. My page
 - myPage()를 호출하고, 리턴된 값이 1이면 while문을 빠져나간다.
- 7. Logout
 - exit를 true로 바꾸고 while문을 빠져나간다.

- **myPage()**

- 아래 6가지 메뉴가 있다.
- 1. My like list
 - showMusicLike()를 호출한다.
- 2. My playlist
 - showMyPlayList()를 호출한다.
- 3. My comments
 - showAllCommentForUser()를 호출한다.
- 4. Edit my info
 - showUserDetail(-1)을 호출한다.
- 5. Withdrawal
 - 정말 탈퇴할 것인지 확인하고, 탈퇴한다면 deleteUser(currentUserNo)를 호출하고 1을 리턴한다.
- 6. Go back
 - exit를 true로 바꾸고 while문을 빠져나간다.

- **showAllPublicPlayList()**

1. 공개되어 있는 Public playlist(플레이리스트이름, 소유자 id) 목록을 출력한다.
2. 만약 공개되어 있는 플레이리스트가 없다면 엔터를 입력받은 뒤 함수를 종료한다.
3. 세부 정보를 볼 플레이리스트를 선택받아 showPlayListDetail(해당 플레이리스트의 title, 소유자 id)를 호출한다.

- **showMusicLike()**

1. 유저가 좋아요를 누른 음악(title, artist)의 목록을 출력한다.
2. 만약 좋아요를 누른 음악이 없다면 엔터를 입력받은 뒤 함수를 종료한다.
3. 아래 3가지 메뉴가 있다.

1. Unlike
 - 어떤 음악을 좋아요 취소할지 입력받아서 좋아요를 취소한다.
2. Play music
 - 어떤 음악을 재생할지 입력받아서 재생한다.
3. Go back
 - exit를 true로 바꾸고 while문을 빠져나간다.

- showMyPlayList()

- 해당 유저의 플레이리스트를 목록을 출력한다.
- 아래 4가지 메뉴가 있다.
 1. Add playlist
 - addPlaylist()를 호출한다.
 2. Show playlist detail
 - 세부정보를 볼 플레이리스트를 입력받고 그 플레이리스트의 title과 userNo를 넣어 show PlaylistDetail()을 호출한다.
 3. Delete playlist
 - 삭제할 플레이리스트를 입력받고 그 플레이리스트의 title을 title에 저장해 deletePlaylist (title)을 호출한다.
 4. Go back
 - exit를 true로 바꾸고 while문을 빠져나간다.

- showAllCommentForUser()

1. 현재 사용자가 작성한 모든 댓글들의 목록(음악, 댓글내용, 댓글 작성일)을 출력한다.
 - 사용자가 작성한 댓글이 없다면 메시지를 출력하고 엔터를 입력받으면 함수를 종료한다.
2. 삭제할 댓글을 입력받고 해당 댓글을 삭제한다. (0을 입력할 때까지 반복해서)

- playMusic(musicNo)

- 현재 사용자의 musicNo에 해당하는 음악에 대한 재생횟수를 1 증가시킨다.

<Show>

- showMusicChart()

1. 모든 음악의 title,artist 목록을 출력한다.
 - 단 음악의 총 플레이된 횟수를 기준으로 해서 순위가 높은 순서대로 출력한다.
2. 플레이횟수가 0회인 음악은 play와 join한 결과에 없으므로 위의 목록에 포함되지 않는다. 따라서 위의 목록에 포함되지 않는 음악의 목록을 출력한다.
3. 1,2 모두에 음악이 없다면 메시지를 출력하고 엔터를 입력받은 뒤 함수를 종료한다.
4. 세부정보를 볼 음악을 선택받아 해당음악의 musicNo를 musicNo에 저장한다.
5. 만약 현재 사용자가 manager라면 managerMusic(musicNo)를 호출하고, user라면 userMusic (musicNo)를 호출한다.

- searchMusic()

1. keyword를 입력받아, keyword가 포함된 title을 갖는 music의 title목록을 출력한다.
 - 만약 그런 음악이 없는 경우, 메시지를 출력하고, 엔터를 누르면 함수를 종료한다.

2. title목록에서 세부정보를 볼 음악을 선택받아 해당 음악의 musicNo를 musicNo에 저장한다.
3. 만약 현재 사용자가 manager라면 managerMusic(musicNo)를 호출하고, user라면 userMusic(musicNo)를 호출한다.

- showAllAlbums()

1. 모든 앨범 목록(앨범이름, 참여한 아티스트 이름 목록)을 출력한다.
 - 앨범이 존재하지 않는다면 메시지를 출력하고 엔터를 입력받으면 함수를 종료한다.
2. 세부 정보를 볼 앨범을 입력받아, 해당하는 앨범의 albumNo를 albumNo에 저장한뒤 현재 사용자가 매니저이면 managerAlbum(albumNo)를 호출하고, 유저이면 userAlbum(albumNo)를 호출한다.

- showAllArtist()

1. 모든 아티스트 목록(아티스트이름)을 출력한다.
 - 아티스트가 존재하지 않는다면 메시지를 출력하고, 엔터를 입력받으면 함수를 종료한다.
2. 관리할 아티스트를 입력받고 해당 아티스트의 artistNo를 artistNo에 저장한다.
3. 선택한 아티스트의 artistNo를 artistNo에 저장하고, artistDetail(artistNo)를 호출한다.

- artistDetail(artistNo)

1. 아티스트의 이름과 데뷔일을 출력한다.
2. 현재 사용자가 매니저인 경우 아래 5가지 메뉴가 있다.
 - 1) Show members
 - showMembers(artistNo)를 호출한다.
 - 2) Show musics
 - showArtistsMusics(artistNo)를 호출한다.
 - 3) Show albums
 - showArtistsAlbums(artistNo)를 호출한다.
 - 4) Delete artist
 - 아티스트를 삭제하면, 아티스트의 음악, 앨범도 삭제된다는 경고문을 출력하고, 그래도 삭제할건지 확인한 뒤, 삭제한다면 deleteArtist(artistNo)를 호출한다.
 - 5) Go back
 - exit를 true로 바꾸고 while문을 빠져나간다.
3. 현재 사용자가 유저인 경우 매니저의 경우에서 4번을 뺀 4가지 메뉴가 있다. (동일하게 작동)

- showMembers(artistNo)

- artistNo에 해당하는 아티스트의 멤버 이름 목록을 보여준다.
- 유저인 경우, 엔터를 입력받고 함수를 종료한다.
- 매니저의 경우 아래 3가지 메뉴가 있다.
 1. add member
 - 멤버이름을 입력받아 아티스트의 멤버로 추가한다.
 2. delete member
 - 삭제할 멤버를 입력받아 아티스트의 멤버 목록에서 삭제한다.
 3. go back
 - 함수를 종료한다.

- showArtistsMusics(artistNo)

- artistNo에 해당하는 아티스트가 참여한 음악의 목록(제목, 참여한 아티스트 이름 목록, 발매일)을 보여준다.
- 매니저의 경우 아래 3가지 메뉴가 있다.
 1. add music
 - addMusic(-1,artistNo)를 호출한다.
 2. delete music
 - 삭제할 음악을 입력받아 musicNo에 저장하고 deleteMusic(musicNo)를 호출한다.
 3. go back
 - 함수를 종료한다.
- 유저의 경우 세부정보를 볼 음악을 선택받아 musicNo에 저장하고 userMusic(musicNo)를 호출한다.

- showArtistsAlbums(artistNo)

- artistNo에 해당하는 아티스트가 참여한 앨범 목록(앨범 이름, 참여한 아티스트이름 목록, 발매일)을 보여준다.
- 매니저의 경우 아래 3가지 메뉴가 있다.
 1. add album
 - addAlbum(artistNo)를 호출한다.
 - 2.delete album
 - 삭제할 앨범을 입력받아 해당하는 앨범의 albumNo를 albumNo에 저장한 뒤 delete Album(albumNo)를 호출한다.
 3. go back
 - 함수를 종료한다.
- 유저의 경우 세부정보를 볼 앨범을 선택받아서 해당 앨범의 albumNo를 이용해 userAlbum(albumNo)를 호출한다.

- managerAlbum(albumNo)

- albumNo에 해당하는 앨범의 이름, 참여한 아티스트 목록, 발매일, 앨범에 속한 음악 목록을 보여준다.
- 아래 5가지 메뉴가 있다.
 1. edit album
 - editAlbum(albumNo)를 호출한다.
 2. delete album
 - deleteAlbum(albumNo)를 호출하고 while문을 빠져나간다.
 3. delete music from album
 - 앨범에서 제거할 음악을 입력받아서 제거한다.
 4. add music
 - 앨범에 새로운 음악을 추가하기 위해 addMusic(albumNo,-1)을 호출한다.
 5. go back
 - while문을 빠져나간다.

- userAlbum(albumNo)

- albumNo에 해당하는 앨범의 이름, 참여한 아티스트 목록, 발매일, 앨범에 속한 음악 목록을 보여준다.
- 세부정보를 보고싶은 음악을 입력받고 해당 음악의 musicNo에 대해 userMusic(musicNo)를 호출한다.

- managerMusic(musicNo)

- showMusicDetail(musicNo)를 호출해 해당 음악에 대한 세부정보를 출력한다.

- 아래 4가지 메뉴가 있다.
 1. delete music
 - deleteMusic(musicNo)를 호출하고 while문을 빠져나간다.
 2. manage comments
 - showAllCommentForMusic(musicNo)를 호출한다.
 3. edit music
 - editMusic(musicNo)를 호출한다.
 4. go back
 - while문을 빠져나간다.

- userMusic(musicNo)

- showMusicDetail(musicNo)를 호출해 해당 음악에 대한 세부정보를 출력한다.
- 아래 5가지 메뉴가 있다.
 1. Play music
 - playMusic(musicNo)를 호출한다.
 2. show comments
 - showAllCommentForMusic(musicNo)를 호출한다.
 3. like / unlike
 - 유저가 해당 음악에 좋아요를 이미 누른 경우 좋아요를 취소하고, 아닌 경우 좋아요를 누른다.
 4. add to playlist
 - addToPlayList(musicNo)를 호출한다.
 5. go back
 - while문을 빠져나간다.

- showMusicDetail(musicNo)

- 음악의 제목, 참여한 아티스트 목록, 장르, 속한 앨범, 재생시간, 발매일, 좋아요수, 재생횟수, 댓글수, 가사를 출력한다.

- showAllCommentForMusic(musicNo)

- 넘겨받은 musicNo를 갖는 음악의 모든 댓글 목록(댓글 내용, 작성자 이름, 작성일)을 보여준다.
- 매니저인 경우: 댓글을 삭제할 수 있다.
 - 삭제할 댓글을 입력받아 그 댓글을 삭제한다.
- 매니저가 아닌 (일반 유저의) 경우 아래 두 가지 메뉴가 있다.
 1. write comment
 - comment 내용을 입력받아 댓글을 추가한다.
 2. go back
 - 함수를 종료한다.

- showUserDetail(userNo)

- userNo에 해당하는 유저의 정보를 출력한다.
- 현재사용자가 userNo에 해당하는 유저라면 editUser(userNo)를 호출한다.
- 현재 사용자가 userNo에 해당하는 유저가 아니고, 현재 사용자가 매니저라면 아래 3가지 메뉴가 있다.
 1. edit user
 - editUser(userNo)를 호출한다.
 2. delete user

- deleteUser(userNo)를 호출하고 while문을 빠져나간다.
3. go back
- while문을 빠져나간다.

- showPlaylistDetail(title,userNo)

- userNo인 유저가 갖는 이름이 title인 플리이리스트에 대한 정보를 출력한다.
 - 플레이리스트 공개여부, 소유자 아이디, 참여한 아티스트 목록, 플레이리스트에 속한 음악
- 현재 사용자가 해당 플레이리스트의 소유자인 경우 아래 6가지 메뉴가 있다.
 1. Change to Private/Public playlist
 - 플레이리스트의 공개여부 설정을 변경한다.
 2. edit title
 - 플레이리스트의 새로운 이름을 입력받아 이름을 변경한다.
 3. delete music in playlist
 - 어떤 음악을 삭제할지 입력받아 그 음악을 플레이리스트에서 삭제한다.
 4. play music
 - playMusic(musicNo)를 호출한다.
 5. like
 - 만약 좋아요를 누르지 않은 음악이라면 좋아요를 누르고, 이미 좋아요를 누른 음악이라면 좋아요를 취소할지 확인하고 취소한다.
 6. go back
 - while문을 빠져나간다.
- 현재 사용자가 플레이리스트의 소유자가 아닌 경우 아래 4가지 메뉴가 있다.
 1. add music to my playlist
 - 어떤 음악을 추가할지 입력받아 addToPlayList(musicNo)를 호출한다.
 2. play music
 - 위의 4번과 동일
 3. like
 - 위의 5번과 동일
 4. go back

<ADD>

- addMusic(albumNo,artistNo)

- 음악의 제목, 아티스트 목록, 장르, 재생시간, 발매일, 속한 앨범, 가사를 입력받는다.
 - artistNo가 -1인 경우에만 아티스트 목록을 입력받는다. 아티스트 목록을 입력받을 때는 입력받은 키워드를 포함하는 아티스트 목록을 출력하고 그 중 한 아티스트를 선택받는다. q를 입력할 때까지 이 과정을 반복한다.
 - albumNo가 -1인 경우에만 앨범을 입력받는다. 앨범을 입력받을 때는 입력받은 키워드를 포함하는 앨범 목록을 출력하고 그 중 한 앨범을 선택받는다.
 - 만약 앨범에 속해있지 않은 음악이라면 n을 입력하도록 한다.
 - 가사를 입력받을 때는 여러 줄을 입력할 수 있는데, q를 입력하면 입력받는 것을 멈춘다.
- 입력받은 정보를 갖는 음악을 추가해준다.

- addAlbum(artistNo)

- 앨범이름, 발매일을 입력받고, 앨범에 참여한 아티스트 목록을 입력받는다.

- artistNo가 -1인 경우에만 아티스트 목록을 입력받고, 아티스트 목록을 입력받는 방식은 addMusic과 동일하다
- 입력받은 정보를 갖는 앨범을 추가해준다.

- addArtist()

- 아티스트 이름, 데뷔일, 멤버목록을 입력받는다.
- 입력받은 이름, 데뷔일을 갖는 아티스트를 추가하고, 멤버도 추가한다.

- addPlaylist()

- 플레이리스트의 이름, 공개여부를 입력받아 플레이리스트를 추가한다.

<Edit>

- editMusic(musicNo)

- 제목, 재생시간, 장르, 발매일, 가사 중 수정할 것을 선택받고 선택한 것을 입력받아 그것으로 수정한다.

- editUser(userNo)

- 이름, 비밀번호 중 수정할 것을 선택받고, 수정할 내용을 입력받아 수정한다.

- editAlbum(albumNo)

- 새로운 제목, 발매일 중 수정할 것을 선택받고 수정할 내용을 입력받아 수정한다.

- addToPlayList(musicNo)

- 현재 사용자가 소유한 플레이리스트의 목록을 출력한다.
- 만약 플레이리스트가 없는 경우 2가지 메뉴가 있다.
 1. add playlist
 - addPlaylist()를 호출해 음악을 추가하기 위해 새롭게 만든 플레이리스트의 이름을 title에 저장한다.
 2. go back
 - 함수를 종료한다.
- 플레이리스트가 있는 경우 음악을 추가할 플레이리스트를 입력받고, 그 플레이리스트의 이름을 title에 저장한다.
- 현재사용자가 소유한 title을 이름으로 갖는 플레이리스트에 musicNo를 갖는 음악을 추가한다.
 - 이미 플레이리스트에 있는 경우에는 메시지를 출력한뒤 함수를 종료한다.

<Delete>

- deleteUser(userNo)

- userNo를 갖는 유저를 삭제한다.
- 유저가 쓴 댓글, 음악 재생기록, 좋아요, 플레이리스트도 함께 삭제한다.

- deleteManager()

- 매니저가 회원탈퇴를 하는 경우 호출되는 함수

- 모든 매니저의 managerNo를 가져와서 현재 사용자의 managerNo를 제외하고 ArrayList에 넣는다.
 - 만약 현재 사용자가 마지막 매니저라면 탈퇴할 수 없다는 메시지를 출력하고 함수를 종료한다.
- 현재 사용자가 담당 관리자인 음악, 유저의 담당관리자에 현재 사용자를 제외한 매니저 중 한 명을 랜덤으로 배정한다.
- 현재사용자(매니저)를 삭제한다.

- deleteMusic(musicNo)

- musicNo를 갖는 음악을 삭제한다.
- 음악에 대한 댓글, 재생기록, 좋아요도 함께 삭제한다.

- deleteArtist(artistNo)

- artistNo를 갖는 아티스트를 삭제한다.
- 아티스트의 음악, 앨범도 함께 삭제한다.

- deleteAlbum(albumNo)

- albumNo를 갖는 앨범을 삭제한다.
- 해당 앨범에 속한 음악을 앨범에 속하지 않은 앨범으로 바꿔준다 (음악의 albumNo를 null로 변경)

- deltePlaylist(title)

- 현재사용자가 소유한 이름이 title인 플레이리스트를 삭제한다.

2. 실행 예시

로그인

```
C:\Users\tjtld\database>java -classpath .;mariadb-java-client-2.7.1.jar MusicSystem

===== Welcome to Music System =====
1. Log In
2. Sign Up
3. Find ID / Password
4. Exit
-----
Input command: 1

===== Login =====
ID: siun
Password: siun
login succeed!
```

회원가입

```
===== Sign Up =====
Are you manager? (manager: enter 1, user: enter 0) : 0
Name: user
Birth Date (Format= yyyy-mm-dd): 2020-12-07
Gender (F/M): F
ID: user
Password: user
Input Password Again: user
sign up succeed!
```

아이디 찾기

```

===== Find ID/Password =====
1. Find ID
2. Change Password
3. Go back
=====
Input command: 1
Manager or User? (manager = Enter 1, User = Enter 0) 0
Name: user
Birth Date (Format = yyyy-mm-dd): 2020-12-07
Gender (F/M): F
ID: user

```

비밀번호 변경

```

===== Find ID/Password =====
1. Find ID
2. Change Password
3. Go back
=====
Input command: 2
Manager or User? (manager = Enter 1, User = Enter 0) 0
ID: user
Name: user
Birth Date (Format = yyyy-mm-dd): 2020-12-07
Gender (F/M): F
Input new Password: newpass
Input new Password Again: newpass

```

<유저>

유저 메인 화면

```

===== Main Menu =====
1. Show music chart
2. Search music
3. Show All Public PlayLists
4. Show All Albums
5. Show All Artists
6. My Page
7. Logout
=====
Input command: 1

```

음원 순위

```

===== Show Music Chart =====
1. palette - IU
2. happy together - ParkHyosin
3. title - AKMU
4. unlucky - IU
5. bluering - IU
6. music - AKMU, BLACKPINK
7. Black out - IU
8. goodbye - ParkHyosin
=====
Select music to see detail: 1

```

음원 세부 정보 -> 음원 재생

```

===== Music Detail =====
Title: palette
Artist: IU
Genre: ggg
Album: palette
Time: 00:03:38      Release Date: 2017-04-21
Like: 2      Play Time: 14      Comments: 2
Lyrics:
I like it Im twenty five
I got this Im truly fine

1. Play music
2. Show Comments
3. Unlike
4. Add to Playlist
5. Go Back

-----
Input command: 1

[ palette ]
0:01 -o----- 00:03:38
      <<      ||      >>
Enter to go back

```

댓글 작성

```

===== Show Comments =====
1. good! - name - 2020-12-07
2. comment - seosiun - 2020-12-07

1. Write Comment
2. Go back

-----
Input command: 1
Comment: good song~

```

좋아요 취소, 좋아요

```

===== Music Detail =====
Title: palette
Artist: IU
Genre: ggg
Album: palette
Time: 00:03:38      Release Date: 2017-04-21
Like: 2      Play Time: 15      Comments: 3
Lyrics:
I like it Im twenty five
I got this Im truly fine

1. Play music
2. Show Comments
3. Unlike
4. Add to Playlist
5. Go Back

-----
Input command: 3

===== Music Detail =====
Title: palette
Artist: IU
Genre: ggg
Album: palette
Time: 00:03:38      Release Date: 2017-04-21
Like: 1      Play Time: 15      Comments: 3
Lyrics:
I like it Im twenty five
I got this Im truly fine

1. Play music
2. Show Comments
3. Like
4. Add to Playlist
5. Go Back

-----
Input command: 3

```

음악 검색 후 플레이리스트에 추가

```

===== Search Music =====
Input keyword: m
1. blueming
2. music

-----
Select music number to see detail: 2

===== Music Detail =====
Title: music
Artist: AKMU / BLACKPINK
Genre: genre
Album: a
Time: 00:03:40      Release Date: 2000-02-12
Like: 1      Play Time: 1      Comments: 0
Lyrics:
lalala
fkfkfk

1. Play music
2. Show Comments
3. Unlike
4. Add to Playlist
5. Go Back

-----
Input command: 4

===== Add Music to PlayList =====
1. pp
2. siuns playlist
3. yeh~~

-----
Select playlist: 2

```

공개 플레이리스트 목록 -> 플레이리스트 세부 정보 (자신의 플레이리스트가 아닐 때)

```

===== Show All Public PlayList =====
1. 222 - 2
2. pp - siun
3. siuns playlist - siun
4. yeh~~ - siun
5. p - u
6. playlist - u

-----
Select playList to see detail (if you want to go back, enter 0): 1

===== PlayList : 222 =====
Public PlayList.
Owner ID: 2
1. palette - IU
2. title - AKMU
3. happy together - ParkHyosin
4. blueming - IU

1. Add music to my playlist
2. Play music
3. Like
4. Go back

```

모든 앨범 목록 -> 앨범 세부 정보

```

===== Show All Album List =====
1. palette - IU
2. Love poem - IU
3. a - AKMU
4. HAPPY TOGETHER - IU, ParkHyosin
5. album - Teayeon, BLACKPINK

-----
Select album to see detail (if you want to go back, enter 0): 4

===== Album Detail =====
Title: HAPPY TOGETHER
Artist: IU, ParkHyosin
Release Date: 2014-11-24
1. happy together - ParkHyosin

-----
Select music to show detail (if you want to go back, Enter 0): 1

```

전체 아티스트 목록 -> 아티스트 세부 정보

```

===== Artist List =====
1. IU
2. AKMU
3. Teayeon
4. ParkHyosin
5. BLACKPINK

-----
Select artist (if you want to go back, enter 0): 1

===== Artist Detail =====
Name: IU, Debut Date: 2008-09-23

1. Show members
2. Show musics
3. Show albums
4. Go back

-----
Input command: 1

```

아티스트의 멤버 목록

```

===== Artist's Member List =====
1. IU
2. jieun

Enter to go back

```

아티스트의 음악 목록

```

===== Artist's Musics =====
1. unlucky - IU
2. palette - IU
3. Black out - IU
4. blueming - IU

-----
Select music number to see detail (Enter 0 to go back): 1

```

아티스트의 앨범 목록

```

===== Artist's Album List =====
1. palette - IU
2. Love poem - IU
3. HAPPY TOGETHER - IU, ParkHyosin

-----
Select album to see detail (If you want to go back, Enter 0): 2

```

my page

```

===== My Page =====
1. My Like list
2. My PlayLists
3. My comments
4. Edit My Info
5. Withdrawal
6. Go back
-----
Input command: 1

```

유저의 좋아요 목록

```

===== Show My Like List =====
1. unlucky - IU
2. palette - IU
3. title - AKMU
4. happy together - ParkHyosin
5. blueming - IU
6. music - AKMU, BLACKPINK

1. Unlike
2. Play music
3. Go back
-----
Input command: 2
Select music: 6

```

유저의 플레이리스트 목록

```

===== Show My PlayList =====
1. pp
2. siuns playlist
3. yeh~~

1. Add PlayList
2. Show PlayList detail
3. Delete PlayList
4. Go back
-----
Input command: 1

```

새로운 플레이리스트 추가

```

===== Add Playlist =====
Title: list
Private(open for only you) or Public(open for everyone) : Public

===== Show My PlayList =====
1. list
2. pp
3. siuns playlist
4. yeh~~

```

플레이리스트 세부 정보 (본인의 플레이리스트일 때)

```

===== Playlist : pp =====
Public PlayList.
Owner ID: siun
1. palette - IU
2. title - AKMU
3. Black out - IU
4. happy together - ParkHyosin
5. music - AKMU / BLACKPINK

1. Change to Private playlist
2. Edit title
3. Delete music in playlist
4. Play music
5. Like
6. Go back
-----
Input Command: 1

```


플레이리스트 이름 변경

```
1. Change to Public playlist
2. Edit title
3. Delete music in playlist
4. Play music
5. Like
6. Go back

-----
Input Command: 2
Input new Title: playy
===== PlayList : playy =====
```

플레이리스트에서 음악 제거

```
1. palette - IU
2. title - AKMU
3. Black out - IU
4. happy together - ParkHyosin
5. music - AKMU / BLACKPINK

1. Change to Public playlist
2. Edit title
3. Delete music in playlist
4. Play music
5. Like
6. Go back

-----
Input Command: 3
Select music: 5

===== PlayList : playy =====
Private PlayList.
Owner ID: siun
1. palette - IU
2. title - AKMU
3. Black out - IU
4. happy together - ParkHyosin
```

플레이리스트 제거

```
===== Show My PlayList =====
1. list
2. playy
3. siuns playlist
4. yeh~~

1. Add PlayList
2. Show PlayList detail
3. Delete PlayList
4. Go back

-----
Input command: 3
Select playlist: 1

===== Show My PlayList =====
1. playy
2. siuns playlist
3. yeh~~
```

유저가 작성한 댓글 목록

```
===== Show My Comments =====
1. unlucky
   good! - 2020-12-06
2. palette
   comment - 2020-12-07
3. palette
   good song~ - 2020-12-07
4. happy together
   pigon.. - 2020-12-07

-----
Select comment to delete (If you want to go back, enter 0): 3
```

유저 세부 정보

```
===== User Detail =====  
ID: siun  
Name: seosiu  
Birth Date: 2000-08-16  
Gender: F
```

유저 정보 수정

```
===== Edit My Info =====  
1. Name  
2. Password  
3. Go back  
-----  
Input command: 1  
Input new Name: siun
```

회원탈퇴

```
===== My Page =====  
1. My Like list  
2. My PlayLists  
3. My comments  
4. Edit My Info  
5. Withdrawal  
6. Go back  
-----  
Input command: 5  
Really want to withdrawal...? (Enter 'yes' for withdrawal): yes
```

<매니저>

매니저 메인 화면

```
===== Main Menu =====  
1. Manage Musics  
2. Manage Users  
3. Manage artists  
4. Manage albums  
5. Edit My Info  
6. Withdrawal  
7. Logout  
-----  
Input command: 1
```

음악 관리

```
===== Manage Musics =====  
1. Show music chart  
2. Search music  
3. Add new music  
4. Go back  
-----  
Input command: 1
```

음악 추가

```

===== Add Music =====
Title: newMusic
Artist (If you want to stop, Enter q): p
1. ParkHyosin 1999-11-04
2. BLACKPINK 2016-08-08
Select artist: 2
Artist (If you want to stop, Enter q): q
Genre: g
Play Time (format = mm:ss): 3:30
Release Date (format = yyyy-mm-dd): 2020-12-07
Album (if no album, input n) : a
1. palette - IU - 2020-01-01
2. a - AKMU - 2014-11-13
3. HAPPY TOGETHER - IU / ParkHyosin - 2014-11-24
4. album - Teayeon / BLACKPINK - 2020-12-06

Select album: 4
Lyrics (if want to stop input, enter 'q')
lyrics
lyrics
q

```

매니저 - 음악 세부정보

```

===== Music Detail =====
Title: palette
Artist: IU
Genre: ggg
Album: palette
Time: 00:03:38      Release Date: 2017-04-21
Like: 2             Play Time: 15      Comments: 2
Lyrics:
I like it Im twenty five
I got this Im truly fine

1. Delete Music
2. Manage Comments
3. Edit Music
4. Go Back

Input command: 2

```

댓글 관리

```

===== Show Comments =====
1. good! - name - 2020-12-07
2. comment - siun - 2020-12-07

Select comment to delete (if you want to go back, enter 0): 2

```

음악 세부정보 수정

```

===== Edit Music =====
1. Title
2. Time
3. Genre
4. Release Date
5. Lyrics
6. Go Back

Input command: 3
Input new Genre: genre

```

유저 목록

```

===== User List =====
1. ID: 2, name: name
2. ID: u, name: u
3. ID: siun, name: siun

Select user to manage (if you want to go back, enter 0): 1

```

유저 세부 정보 - 매니저

```

===== User Detail =====
ID: 2
Name: name
Birth Date: 1999-01-01
Gender: F

1. Edit user
2. Delete user
3. Go Back
-----
Input command: 1

```

아티스트 관리

```

===== Manage Artists =====
1. Show All Artist
2. Add new Artist
3. Go back
-----
Input command: 1

```

아티스트 세부 정보

```

===== Artist Detail =====
Name: IU, Debut Date: 2008-09-23

1. Show members
2. Show musics
3. Show albums
4. Delete artist
5. Go back
-----
Input command: 1

```

아티스트 멤버 목록 -> 멤버 추가

```

===== Artist's Member List =====
1. IU
2. jieun

1. Add Member
2. Delete Member
3. Go back
-----
Input command: 1
Input member name: i

===== Artist's Member List =====
1. i
2. IU
3. jieun

1. Add Member
2. Delete Member
3. Go back
-----
Input command: 2
Input member number to delete: 1

```

아티스트의 음악 목록

```

===== Artist's Musics =====
1. unlucky - IU
2. palette - IU
3. Black out - IU
4. blueming - IU

1. Add Music
2. Delete Music
3. Go back
-----
Input command: 3

```

아티스트의 앨범 목록

```
===== Artist's Album List =====
1. palette - IU
2. Love poem - IU
3. HAPPY TOGETHER - IU, ParkHyosin

1. Add Album
2. Delete Album
3. Go back
=====
Input command: 3
```

아티스트 추가

```
===== Add Artist =====
Name: artist
Debut date (Format = yyyy-mm-dd): 2020-12-07
Member list (separate by comma(,))
mem,ber
```

3. SQL문 명세

- 매니저 로그인
 - select * from manager where id = '[id]' and password = '[password]'
- 유저 로그인
 - select * from user where id = '[id]' and password = '[password]'
- 아이디 중복 확인 (유저와 매니저를 통틀어서 중복이 없어야 한다.)
 - select (select count() from user where user.id='[id]')+(select count(*) from manager where id='[id]')
- 매니저 인원 수 확인
 - select managerNo from manager
 - 매니저가 없으면 유저 가입을 막고, 마지막 남은 매니저가 탈퇴하는 것을 막기 위해
 - 유저가 회원가입할 때 매니저 중 한 명을 랜덤으로 담당 관리자로 배정하기 위해
- 유저 회원 가입
 - insert into user(id,password,name,birth_date,gender,managerNo) values('[id]', '[password]', '[name]', '[birth_date]', '[gender]', [managerNo])
 - managerNo는 현재 존재하는 매니저 중 한 명을 랜덤으로 골라서 정한다.
- 매니저 회원 가입
 - insert into manager(id,password,name,birth_date,gender) values('[id]', '[password]', '[name]', '[birth_date]', '[gender]')
- 아이디 찾기
 - select id from [manager/user] where name='[name]' and birth_date='[birth_date]' and

gender='[gender]'

- 매니저/유저의 존재 여부 확인
 - select * from [manager/user] where name='[name]' and birth_date='[birth_date]' and gender='[gender]' and id='[id]'
- 비밀번호 변경
 - update [manager/user] set password='[password]' where id='[id]'
- 전체 유저 정보
 - select * from user
- 공개 플레이리스트 목록
 - select title, id, user.userNo from playlist, user where user.userNo=playlist.userNo and isOpen=1
- 유저가 '좋아요'한 음악 목록
 - select title, musicNo from music where musicNo in (select musicNo from likes where userNo=[userNo])
- 음악에 참여한 아티스트 목록
 - select name from artist, make_music where artist.artistNo=make_music.artistNo and musicNo=[musicNo]
- 좋아요
 - insert int likes values([musicNo], [userNo])
- 좋아요 취소
 - delete from likes where userNo=[userNo] and musicNo=[musicNo]
- 유저의 플레이리스트 목록
 - select * from playlist where userNo=[userNo]
- 특정 유저가 작성한 댓글 목록 (작성일 순으로)
 - select title, content, writeDate, music.musicNo from music, comment where music.musicNo=comment.musicNo and userNo=[userNo] ordered by writeDate
- 댓글 삭제
 - delete from comment where musicNo=[musicNo] and userNo=[userNo] and content='[content]'
- 특정 음악 정보
 - select * from music where musicNo=[musicNo]
- 유저가 음악을 재생한 적 있는지 확인
 - select count(*) from play where musicNo=[musicNo] and userNo=[userNo]

- 유저가 처음 음악을 재생하는 경우
 - insert into play values([musicNo], [userNo], 0)
- 유저가 음악을 재생하는 경우, 재생횟수 1 증가
 - update play set cnt_play=cnt_play+1 where musicNo=[musicNo] and userNo=[userNo]
- 전체 음악 목록 (재생 횟수 순서로 (내림차순) →순위)
 - select title, music.musicNo, sum(cnt_play) from music join play on music.musicNo = play.musicNo group by music.musicNo order by sum(cnt_play) desc
 - 같은 musicNo를 가지는 토플이 생기기 때문에 musicNo로 group by하고, sum(cnt_play)를 기준으로 내림차순으로 가져온다.
- 한 번도 재생된 적 없는 음악 목록
 - select title, music.musicNo from music where music.musicNo not in(select music.musicNo join play on music.musicNo=play.musicNo)
 - 한 번도 재생된 적 없는 음악은 play와 music을 조인한 목록에 없으므로 따로 가져와준다.
- 키워드로 음악 검색
 - select title, musicNo from music where title like '%[keyword]%'
- 전체 앨범 목록
 - select title, albumNo from album
- 앨범에 참여한 아티스트 목록
 - select name from artist, make_album where artist.artistNo=make_album.artistNo and albumNo=[albumNo]
- 전체 아티스트 목록
 - select * from artist
- 특정 아티스트 정보
 - select * from artist where artistNo=[artistNo]
- 아티스트의 멤버 목록
 - select mName from artist_member where artistNo=[artistNo]
- 아티스트 멤버 추가
 - insert into artist_member values([artistNo],[mName])
- 아티스트의 멤버 삭제
 - delete from artist_member where artistNo=[artistNo] and mName='[mName]'
- 특정 아티스트의 음악 목록
 - select title,music.musicNo from artist,make_music,music where make_music.artistNo =

artist.artistNo and artist.artistNo=[artistNo] and make_music.musicNo=music.musicNo

- 특정 아티스트의 앨범 목록
 - select title,album.albumNo from artist,make_album,album where make_album.artistNo = artist.artistNo and artist.artistNo=[artistNo] and make_album.albumNo=album.albumNo
- 특정 앨범 정보
 - select title,releaseDate from album where albumNo=[albumNo]
- 특정 앨범의 음악 목록
 - select title, musicNo from music where albumNo=[albumNo]
- 앨범에서 음악 제거
 - update music set albumNo=null where musicNo=[musicNo]
- 유저가 특정 음악을 좋아요 했는지 여부 확인
 - select count(*) from likes where userNo=[userNo] and musicNo=[musicNo]
- 특정 음악이 속해있는 앨범확인
 - select album.title from album,music where music.albumNo=album.albumNo and musicNo=[musicNo]
- 특정 음악의 좋아요 수
 - select count(*) from likes where musicNo=[musicNo] group by musicNo
- 특정 음악의 재생횟수
 - select sum(cnt_play) from play where musicNo=[musicNo] group by musicNo
- 특정 음악의 댓글 수
 - select count(*) from comment where musicNo=[musicNo] group by musicNo
 - 좋아요, 재생횟수, 댓글 수 모두 특정 음악에 대한 것이므로 group by musicNo를 해준다.
- 특정 음악에 대한 댓글 목록 (작성일 순으로)
 - select content,name,writeDate,user.userNo from comment,user where comment.userNo = user.userNo and musicNo=[musicNo] order by writeDate
- 댓글 작성
 - insert into comment values([musicNo],[userNo],[writeDate],[content])
- 특정 매니저 정보
 - select * from manager where managerNo=[managerNo]
- 특정 유저 정보
 - select * from user where userNo=[userNo]

- 특정 플레이리스트 정보
 - select title,isOpen,id from playlist,user where title='[title]' and user.userNo=playlist.userNo and playlist.userNo=[userNo]
- 특정 플레이리스트에 있는 음악 목록
 - select title, musicNo from music where musicNo in (select musicNo from add_to_playlist where title='[title]' and userNo=[userNo])
- 특정 플레이리스트의 공개여부 변경
 - update playlist set isOpen=[isOpen] where title = '[title]' and userNo=[userNo]
- 플레이리스트 이름 변경
 - update playlist set title=[new title] where title = '[title]' and userNo=[userNo]
- 플레이리스트에서 음악 제거
 - delete from add_to_playlist where musicNo=[musicNo] and userNo=[userNo] and title='[title]'
- 특정 키워드를 포함하는 아티스트 목록
 - select * from artist where name like '%[keyword]%'
- 특정 키워드를 포함하는 앨범 목록
 - select album.albumNo, releaseDate,title,name from album,make_album,artist where artist.artistNo = make_album.artistNo and make_album.albumNo=album.albumNo and title like '%[keyword]%' group by albumNo
 - 아티스트가 여러 명인 경우 여러 개의 토플이 포함되므로 group by albumNo를 넣어준다.
- 음악 추가 (앨범에 포함되어있지 않은)
 - insert into music(genre,title,lyrics,playtime,releaseDate,managerNo) values('[genre]', '[title]', '[lyrics]', '[playtime]', '[releaseDate]',[managerNo])
- 음악 추가 (앨범에 포함된)
 - insert into music(genre,title,lyrics,playtime,releaseDate,albumNo,managerNo) values('[genre]', '[title]', '[lyrics]', '[playtime]', '[releaseDate]',[albumNo],[managerNo])
- 방금 추가한 음악의 musicNo
 - select max(musicNo) from music
 - 새로운 음악을 추가할 때, 1씩 증가된 값이 musicNo에 할당된다.
- 아티스트의 음악 추가
 - insert into make_music values([artistNo],[musicNo])
- 앨범 추가
 - insert into album(title,releaseDate) values('[title]', '[releaseDate]')
- 방금 추가한 앨범의 albumNo

- select max(albumNo) from album
- 새로운 앨범을 추가할 때, 1씩 증가된 값이 albumNo에 할당된다.

- 아티스트의 앨범 추가
 - insert into make_album values([artistNo],[albumNo])

- 아티스트 추가
 - insert into artist(name,debutDate) values('[name]', [debutDate])

- 방금 추가한 아티스트의 artistNo
 - select max(artistNo) from artist
 - 새로운 앨범을 추가할 때, 1씩 증가된 값이 artistNo에 할당된다.

- 플레이리스트 추가
 - insert into playlist values('[title]', '[userNo],[isOpen])

- 음악 수정
 - update music set title='[title]', playtime='[playtime]', genre='[genre]', releaseDate='[releaseDate]', lyrics='[lyrics]' where musicNo=[musicNo]

- 유저정보 수정
 - update user set name='[name]', password='[password]' where userNo=[userNo]

- 매니저 정보 수정
 - update manager set name='[name]', password='[password]' where managerNo=[managerNo]

- 앨범 정보 수정
 - update album set title='[title]', releaseDate='[releaseDate]' where albumNo=[albumNo]

- 플레이리스트에 음악추가시 이미 추가되어있는 음악인지 확인
 - select count(*) from add_to_playlist where musicNo=[musicNo] and userNo=[userNo] and title='[title]'

- 플레이리스트에 음악 추가
 - insert into add_to_playlist values([musicNo],[userNo],[title])

- 유저 삭제
 - delete from user where userNo=[userNo]

- 특정 매니저가 담당 관리자인 유저의 매니저 변경
 - update user set managerNo=[new managerNo] where managerNo=[managerNo]

- 특정 매니저가 담당 관리자인 음악의 매니저 변경
 - update music set managerNo=[new managerNo] where managerNo=[managerNo]

- 매니저 삭제
 - delete from manager where managerNo=[managerNo]
- 아티스트 삭제
 - delete from artist where artistNo=[artistNo]
- 음악 삭제
 - delete from music where musicNo=[musicNo]
- 앨범 삭제
 - delete from album where albumNo=[albumNo]
- 플레이리스트 삭제
 - delete from playlist where title='[title]' and userNo=[userNo]