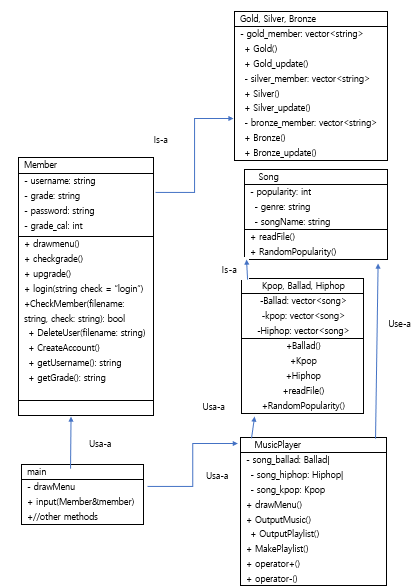
객체지향 텀 프로젝트

유동균 유지원 권태열 이충훈

•문제 명세

저희 팀의 프로젝트 주제는 음악 플레이어입니다. 음악 플레이어에 들어가는 기능으로는 플레이어를 이용하는 회원등록 및 관리, 음악보기, 나만의 플레이리스트 만들기입니다.

회원등록 기능을 만들어 이용자가 회원가입, 로그인 및 회원탈퇴가 가능하도록 만듭니다. 또한 이용자에 등급을 매겨 이후 기능 사용에 차별점을 넣었습니다. 음악보기 기능은 모든 장르의 음악을 출력해 볼 수 있으며, 나의 플레이리스트 기능은 플레이리스트를 만들어 노래를 추가하거나 삭제하고 플레이리스트의 노래 모음을 조회 가능할 수 있는 기능까지 만들었습니다.

•분석

Class Bronze, Silver, Gold는 사용자의 등급을 관리하기 위해 사용되는 클래스입니다. class Member에서 등급 클래스를 “is-a”관계로 상속받아 checkgrade()와 upgrade()함수로 사용자의 등급을 업데이트할 수 있습니다. Class Member는 등록한 회원의 아이디와 비밀번호를 멤버변수로 가지고 있는 클래스로, 회원의 정보를 처리하기 위해 위와 같은 함수를 사용합니다.

class Song은 멤버변수로 popular, songName, genre라는 변수로 받아 class Ballad, Hiphop, K-pop을 “is-a” 관계로 사용됩니다. 이와 같은 장르 클래스는 다시 class MusicPlayer에서 “has-a”관계로써 사용되어 각 장르별 노래 리스트를 출력합니다. 또한 MusicPlayer 클래스는 Member 클래스와 “is-a”관계로써 사용자의 등급을 사용하여 플레이리스트에 추가할 수 있는 노래의 수를 제한합니다.

main()함수는 위의 클래스를 사용하여 프로그램을 실행시키기 때문에 “Use-a”관계를 사용하였습니다.

IOPE

I:

콘솔창에서 사용자 입력 메뉴로 로그인, 회원가입, 회원탈퇴, 종료 4개의 선택지 중 하나를 입력을 받습니다. 회원가입, 회원탈퇴는 아이디와 비밀번호를 입력 받아 처리하고, 로그인을 입력하면 다음 메뉴로 넘어갑니다.

로그인 이후에는 플레이리스트, 노래보기 두 개의 선택지가 존재하며, 플레이리스트는 사용자의 플레이리스트를 확인 및 수정이 가능하며 노래보기는 노래가 인기도 순서대로 출력됩니다.

O:

로그인 성공시에는 “로그인 성공”과 사용자의 현재 등급이 출력되며 메뉴의 구성이 플레이리스트, 노래보기 선택이 가능하도록 출력합니다. 플레이리스트를 선택하면 아이디와 같은 이름의 텍스트 파일 안의 노래의 정보를 출력하고, 노래보기를 선택하면 모든 노래의 정보가 담긴 텍스트 파일을 읽어 랜덤 한 인기도를 부여하고 인기도 순서대로 출력합니다.

P:

화면을 그리는 UI는 외부함수 Draw\_Ui와 각 클래스의 drawMenu를 사용하여 그린다. Member클래스는 사용자의 username(ID)와 password(비밀번호)를 입력 받고 객체로 갖으며, drawMenu(), checkgrade(), CheckMemeber(), CreateAccount(), DeleteUser() 등의 함수를 멤버로 사용합니다. “login.txt”파일에 회원 정보를 작성하거나 읽어 로그인, 회원가입, 회원탈퇴를 구현합니다. 만약 같은 아이디로 일정 횟수 이상 로그인을 하였을 경우 등급이 자동으로 업데이트됩니다.

Song클래스는 노래의 제목 및 장르, 인기도를 활용하기 위해 만들었습니다. Song클래스는 Ballad, Hiphop, K-pop 클래스에 상속되어 장르별로 노래를 저장하고 인기도를 부여합니다.

각 장르별 클래스의 인스턴스를 객체로 MusicPlayer클래스를 작성하고, Memeber클래스의 username을 가져와 사용자의 이름으로 플레이리스트를 만들어 노래를 저장 및 삭제할 수 있도록 합니다.

E:

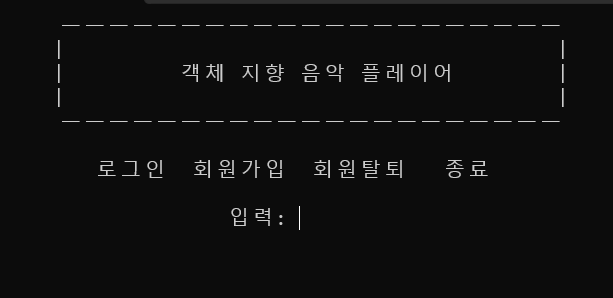
회원가입에서 username: abc, password: 123으로 하여 텍스트 파일에 저장합니다. 다시 abc와 123을 입력하여 로그인하여 플레이리스 또는 노래보기를 선택합니다. 노래보기를 선택하면 랜덤으로 인기도가 부여된 노래들이 인기도 순서로 출력됩니다. 플레이리스트를 입력하면 사용자의 아이디로 된 플레이리스트를 생성하고 노래의 제목을 입력 받아 노래를 추가, 삭제가 가능하며, 만약 다시 로그인을 하여 자신의 플레이리스트를 보고싶다면 “abc.txt”을 불러옵니다.

•설계

**Class Gold, Silver, Bronze**

Gold, Silver, Bronze클래스는 각각 회원의 등급을 나타냅니다. 생성자함수에 등급별 회원 파일을 읽는 함수를 넣었고 그 함수를 통해 해당등급의 멤버가 들어간 문자열벡터에 넣어 초기화한다. 여기서 생성한 벡터를 사용하여 등급의 멤버 상태를 최신화하며 upadate() 함수를 통해 벡터의 새로운 회원으로 텍스트 파일을 수정합니다.

**Class Member**



Member 클래스는 회원의 아이디, 비밀번호, 등급을 관리하는 클래스이며, 등급 관련 클래스(Class Gold, Silver, Bronze)를 상속받습니다. 멤버로 username(회원ID), password(비밀번호), grade(등급), grade\_cal(회원 등급 계산)을 갖고 있습니다. drawMemu()함수를 사용하여 시작화면 하단의 메뉴를 그립니다.

checkgrade()함수는 각 등급에 해당하는 메모장에 있는 명단을 통해서 현 Member클래스의grade와 grade\_cal에 등급과 숫자를 부과합니다. upgrade() 함수는 grade\_cal이 일정 기준을 넘어가게 되면 하위 등급 파일에서 상위 등급 파일로 사용자 아이디를 이동시켜 등급을 올려주는 함수입니다. 등급업을 할 수 있는 방법은 로그인을 할 때마다 grade\_cal이 증가하며 grade\_cal을 기준으로 잡았습니다. login() 함수는 사용자의 로그인 여부를 확인하며 CheckMember함수를 통해 login 파일을 읽어 사용자가 입력한 정보가 존재하는지 확인합니다.

DeleteUser ()함수는 회원탈퇴 시 사용되는 함수로, 입력 받은 아이디와 비밀번호를 파일에서 삭제하고 사용자의 플레이스 또한 삭제하는 함수로 입력 받은 정보를 제외한 정보를 임시파일에 작성 후, 기존 파일을 삭제하고 임시 파일을 기존 파일의 이름으로 변경하는 형식으로 작동합니다. CreateAccount 함수는 입력받은 정보를 파일의 끝에 추가하는 함수로 회원가입에 사용됩니다. 그리고 새로운 회원을 생성하면 자동으로 bronze.txt에 들어가도록 하였습니다

**Class Song**

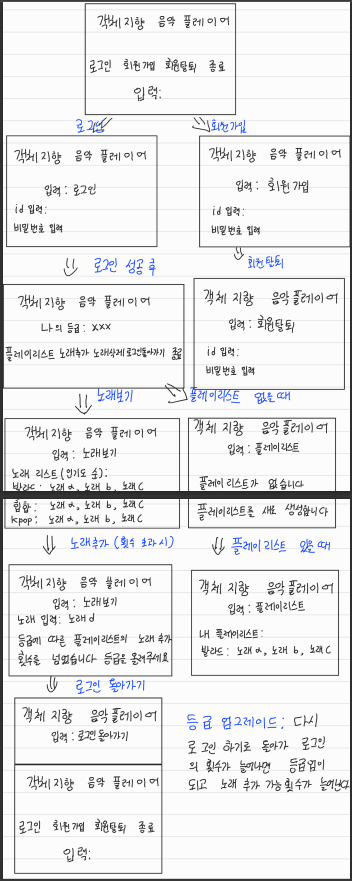
Song 클래스는 노래의 인기도, 장르, 제목을 관리하는 클래스입니다. class Song은 각 장르의 클래스(Ballad, Hiphop, K-pop)에 상속됩니다. 각 장르 (Ballad, Hiphop, K-pop)는 Song 클래스를 사용하여 객체 배열을 벡터로 만들어 노래의 내용을 담았습니다. music 파일에서 각 장르의 음악을 추출하여 RandomPopularity 함수를 통해 1부터 5까지의 인기도를 노래에 무작위로 적용하여 벡터에 저장합니다.

**Class MusicPlayer**

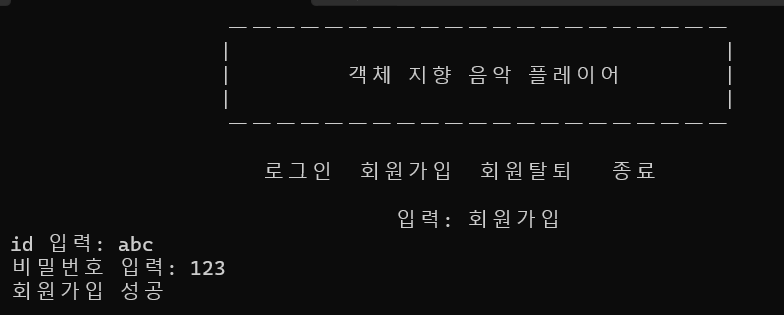
MusicPlayer클래스는 사용자의 플레이리스트를 관리하는 기능을 담당합니다. 장르 클래스의 인스턴스를 생성하고, 노래의 인기도 순서대로 정렬하기위해 algorithm 헤더파일에 있는 sort함수를 통해 정렬합니다. Makeplaylist함수로 로그인 후 첫 플레이리스트 메뉴를 선택하였을 경우 자동으로 사용자의 아이디와 같은 이름(ex)abc\_playlist)으로 플레이리스트 파일을 생성합니다. Outputmusic()함수를 통해 전체 노래를 출력하거나 OutputPlaylist()함수로 플레이리스트에 저장한 노래를 출력 할 수 있습니다. 플레이리스트에 음악을 추가하기 위해 “+”연산자 중복을 사용하여 “mp(플레이리스트) + 노래제목”형식으로 추가할 수 있습니다. 또한, 연산자 중복 함수에는 songCount라는 변수를 갖고 있는데 이는 등급별로 플레이리스트에 추가할 수 있는 음악의 수를 제한하기 위해 사용됩니다. 이와 반대로 “-“연산자를 사용하여 입력 받은 음악의 노래를 “mp - 노래제목”형식을 사용하여 플레이리스트에서 삭제하도록 하였습니다.

**외부함수**

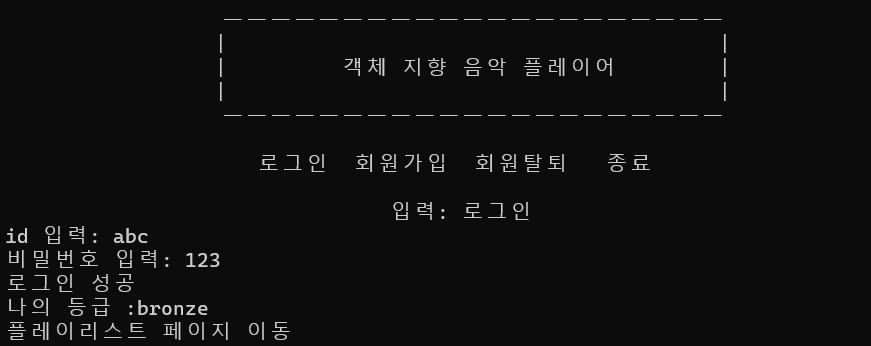
외부 함수로는 Ui를 그리는 Draw\_Ui() 그리고 id와 비밀번호 입력에 사용하는 input()함수, sort에 사용할 compare()함수 그리고 main에 위 클래스와 함수를 사용해 구성한 내용을 Login\_Check(), Playlist\_Check()로 만들어 사용에 조금 더 용이하게 만들었습니다.

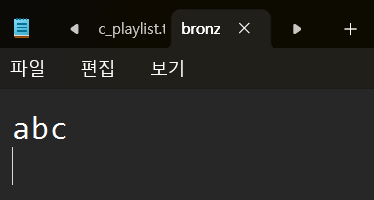
•구현 및 결과

위 그림은 프로그램을 구상하면서 그린 프로그램의 실행 예상도이며 아래의 사진은 완성된 프로그램의 실행화면을 단계별로 캡쳐한 사진입니다. 아래의 사진을 보면 예상도와 유사하게 프로그램이 작동하는 것을 볼 수 있습니다.

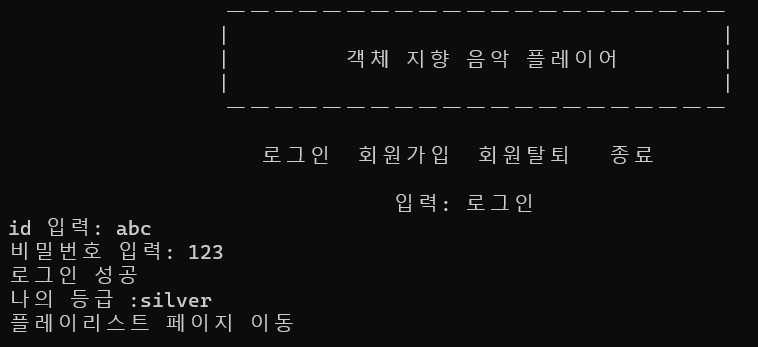


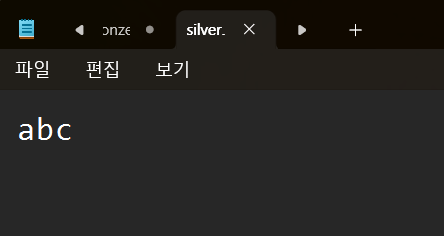
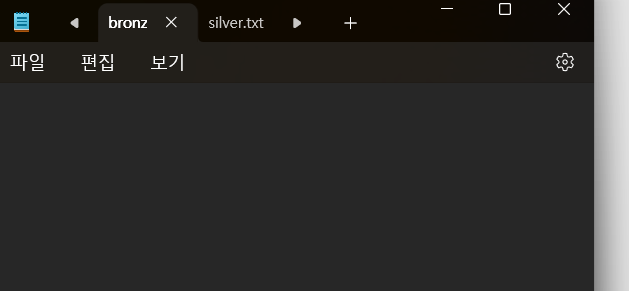
회원가입





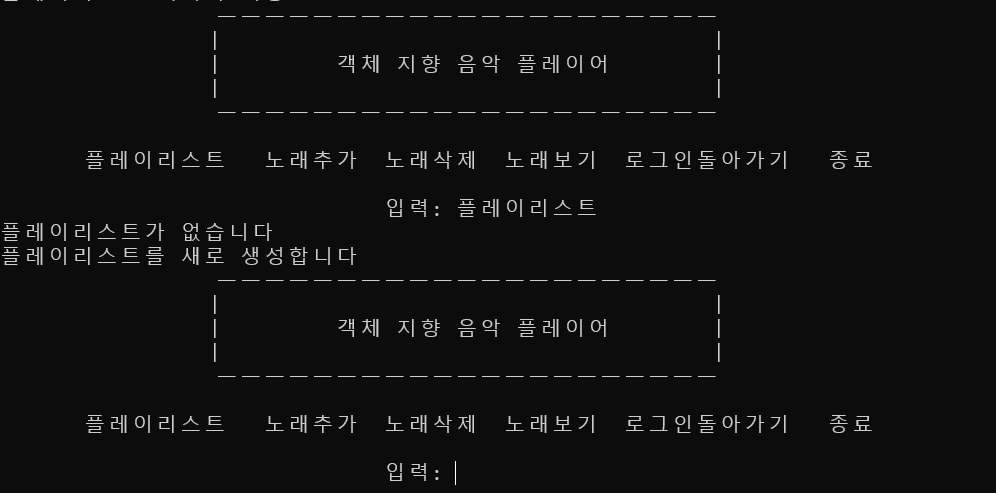
로그인 이후 화면, 브론즈 등급 리스트에 abc가 추가됨



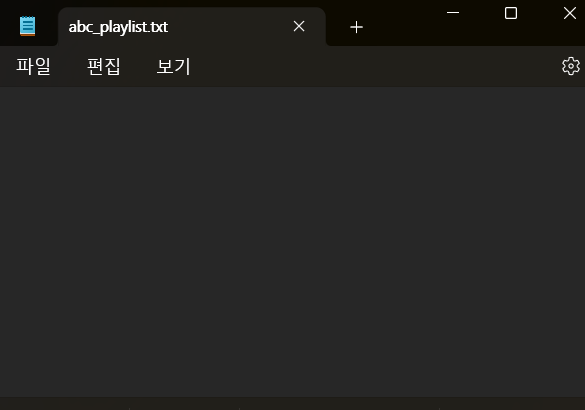
 

일정 횟수 이상 로그인 하였을 경우 silver로 업데이트

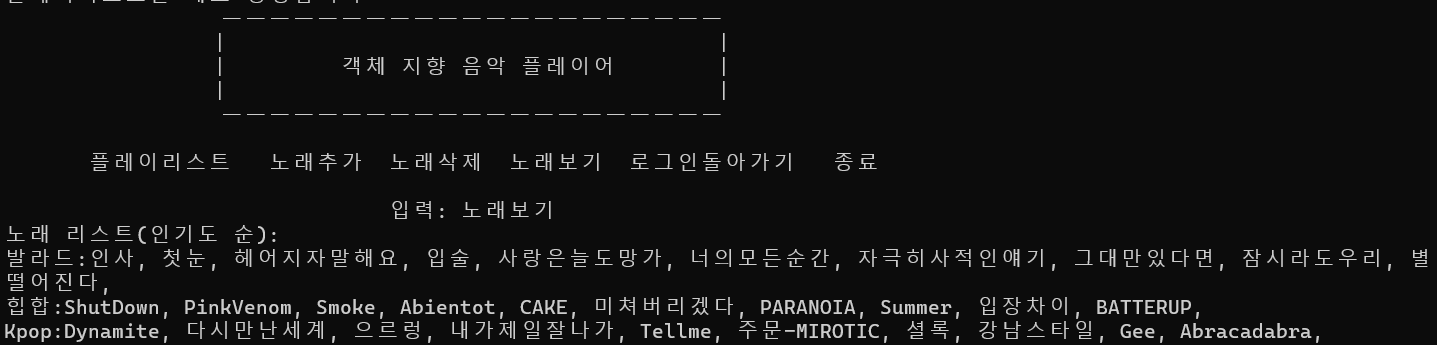
bronze에서 abc가 삭제되고 silver에 abc가 생김



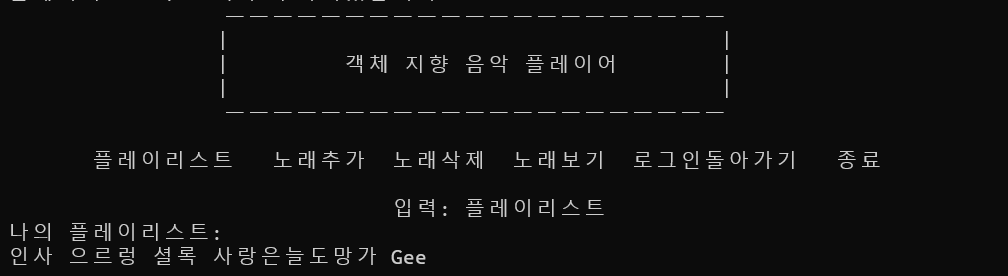
플레이리스트 메뉴를 로그인 이후 처음 입력하였을 경우

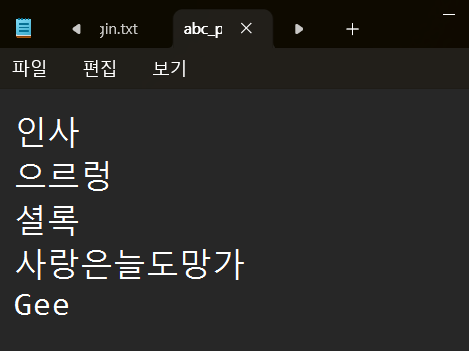


abc\_playlist.txt로 플레이리스트가 생성됨

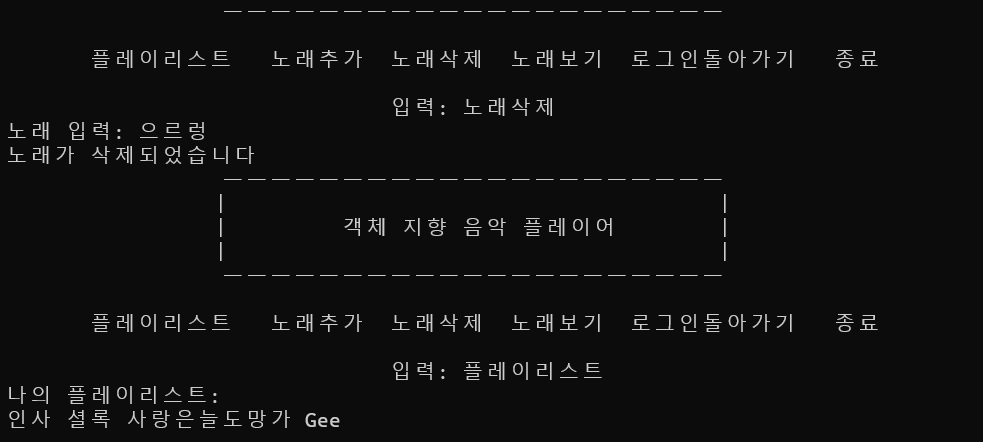


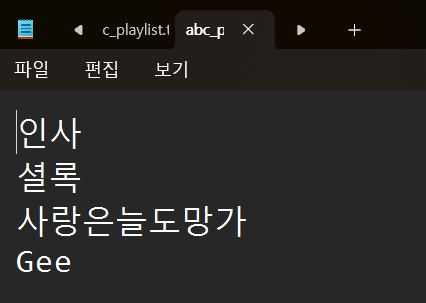
노래 보기를 입력하였을 때 “music.txt”에 저장 되어있는 노래가 장르별 인기도 순서대로 출력됨



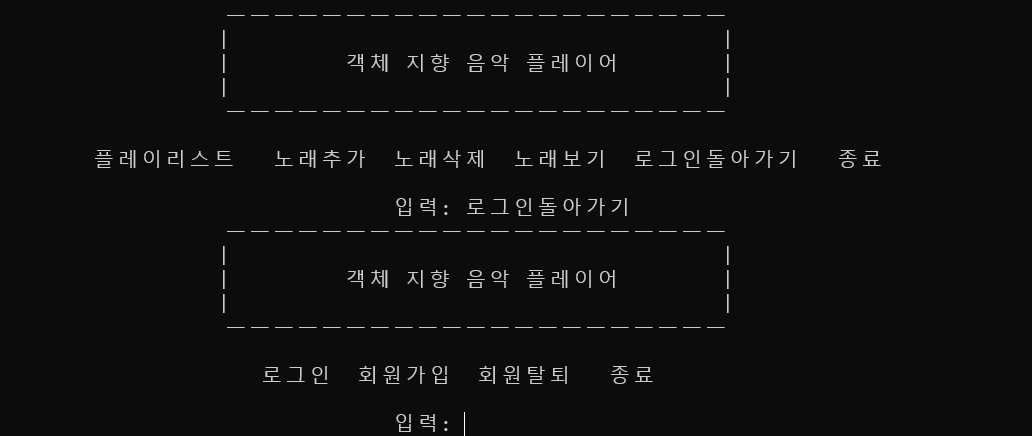


노래추가 후 플레이리스트 확인

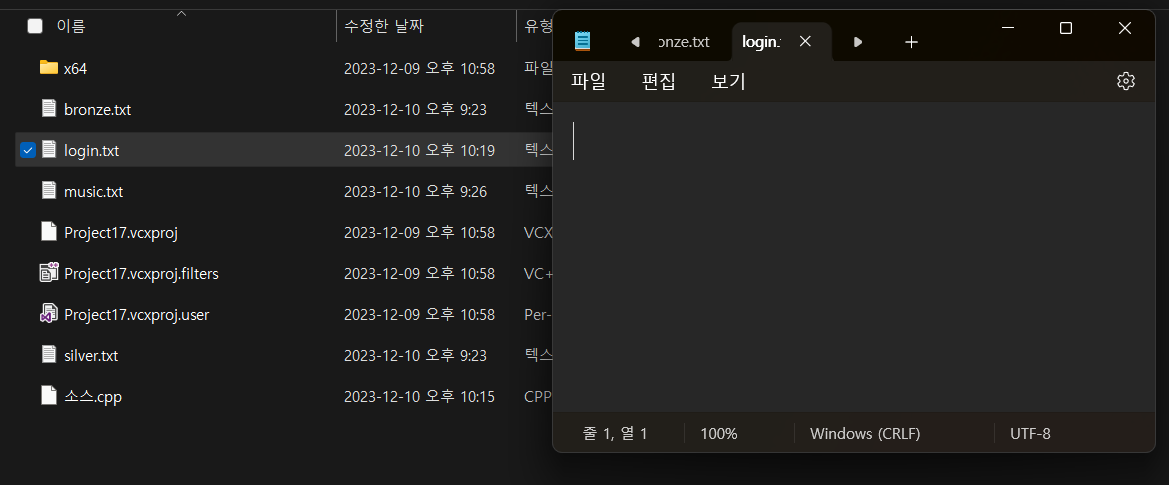
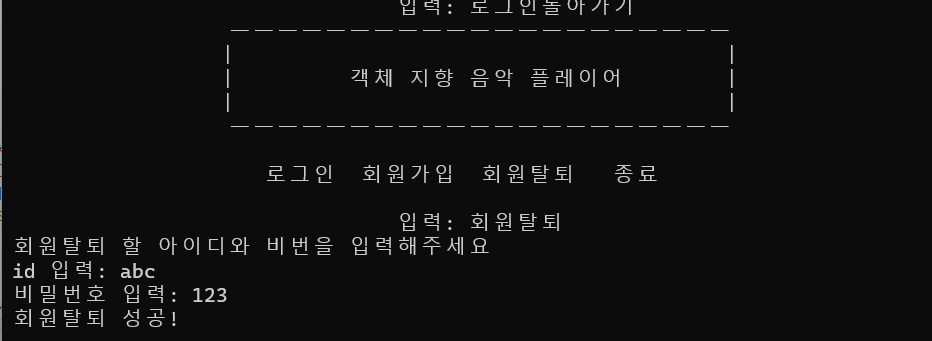




노래삭제 후 플레이리스트의 모습

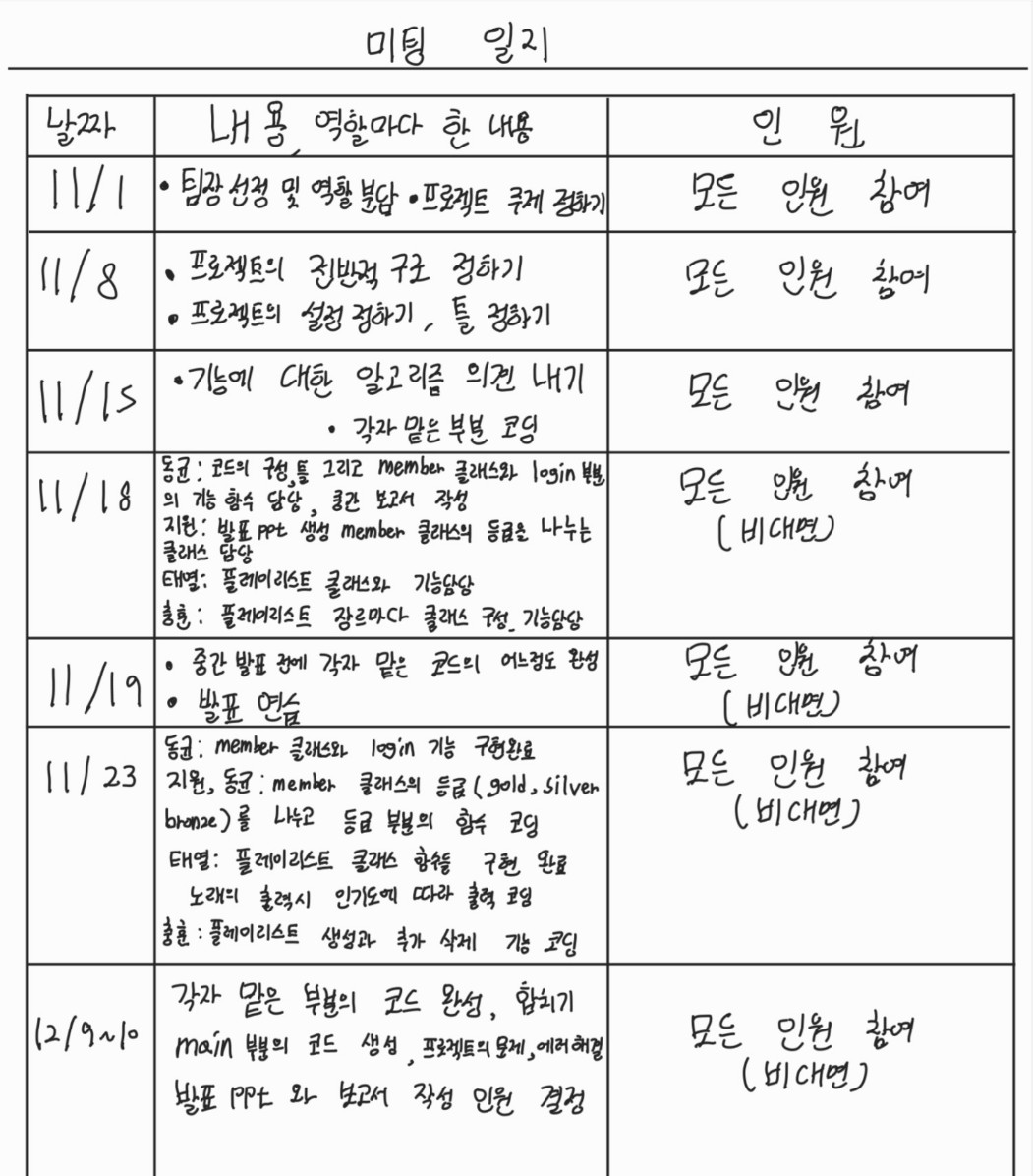


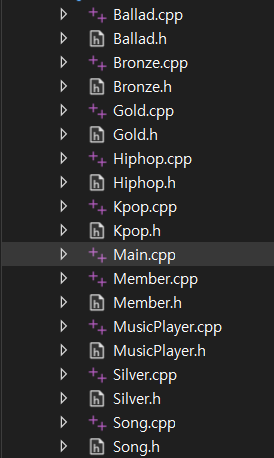
로그인 돌아가기를 입력하였을 경우



회원탈퇴 후 login파일에 abc 삭제 및 “abc\_playlist.txt”삭제

•팀 구성 및 역할분담





팀 구성은 조장 유동균 그리고 조원 유지원, 이충훈, 권태열로 구성되어 있습니다.

중간보고서 전 주제와 주제의 세부내용은 팀원 모두가 모여 의견을 냈습니다. 주제는 권태열 조원이 의견을 내서 모두 동의를 하였고 주제의 세부내용, 기능은 이충훈 조원 그리고 다른 모든 인원이 서로 주제의 세부 기능들의 의견을 제시하고 알고리즘을 생각해 봤습니다.

동균: 각 클래스의 틀과 Member클래스 그리고 안의 여러 기능을 하는 함수의 코드 작성, 코드 완성 후 main작성과 프로그램의 점검, 오류해결

지원: 각 등급마다 세부 클래스(Gold, Silver, Bronze)를 나누어 해당하는 함수와 배열 생성 코드 작성

태열: MusicPlayer클래스의 여러 기능의 함수 생성, 노래출력의 함수들을 작성

충훈: 노래마다 각 장르의 클래스를 생성 후 노래의 추가, 삭제 함수들을 연산자 중복을 사용해 해결, 작성

동균: 중간보고서 작성, 기말 발표 / 지원: 중간발표 ppt 작성, 중간발표 / 태열: 기말보고서 작성 / 충훈: 기말발표 ppt 작성

각자 위에 표시하고 작성했다시피 각각의 내용을 맡아 하였습니다.

미팅은 주제와 세부내용을 정할 때까지는 모든 조원이 직접 만나 정했습니다. 이후 클래스와 코드의 세부적인 내용은 통화와 카톡을 통한 비대면 소통으로 코드를 절충해 나아갔습니다.

•의견

유동균

팀원과 함께 프로그램을 구성하고 알고리즘을 생각하며 코딩하는 부분에서 다양한 의견과 여러가지를 배울 수 있어서 좋았다. c++언어 코딩으로 기능들을 구현하면서 구현 성공한 기능들에 대해서는 다양한 방법과 알고리즘, 객체지향을 사용해서 만들어 좋았지만 조금 더 다양하고 많은 기능과 콘솔창이 아닌 다른 GUI를 만들지 못한 부분이 조금 아쉬웠다. 프로그램 제작을 팀플로 진행함으로 서 혼자 하는 것과 많은 다른 점 있다는 것을 배웠고 객체지향에 대해 공부가 되고 좋은 과제경험이 되었다.

권태열

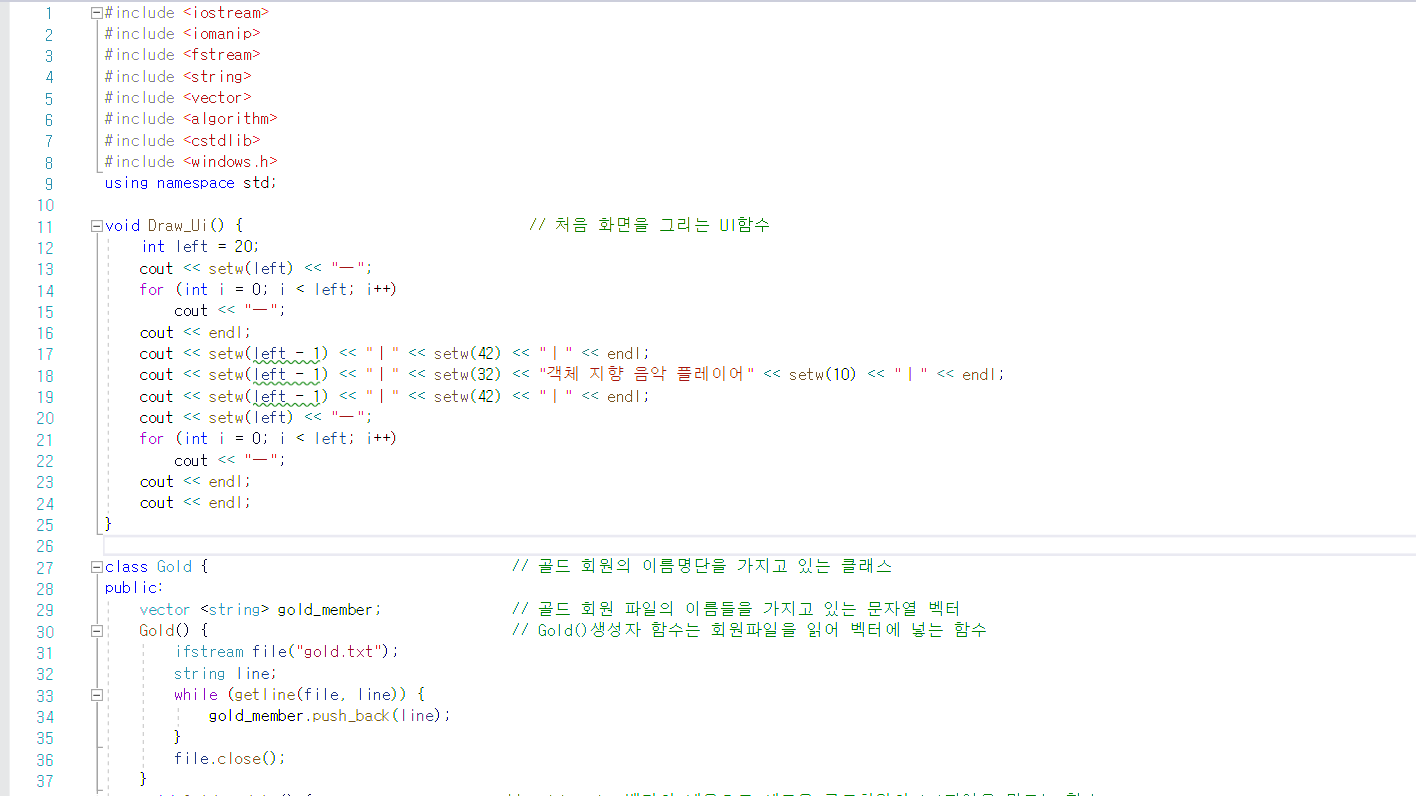
팀원들과 함께 프로그램을 구성한다는 부분에서 다양한 마찰이 있을 것 같아 걱정이 많았었다. 하지만 우려와는 반대로 다양한 의견이 나뉘면서 생각도 못 했던 아이디어를 얻을 수 있었으며, 혼자 프로그래밍을 진행하는 것 보다 각자 파트를 나누어 진행하다 보니 이런 방법으로도 구현할 수 있구나 라는 것을 배울 수 있었다. 프로그램을 제작하다 보니 한번에 다양한 노래를 추가하거나 삭제할 수 있게 만들지 못한점과 타 음원 서비스의 데이터를 가져오고 싶었지만 그러지 못한점이 아쉬웠다.

이충훈

첫 팀 프로젝트를 하면서 처음 구상부터 마지막 회의까지 쉬운 과정은 아니였지만 팀원 모두에게 배울 점이 있었던 것 같습니다. c++언어에 대해 흥미롭게 공부할 수 있는 계기가 되었고 특히나 수업시간에 배운 상속, 연산자 중복등을 직접 활용할 수 있던 게 인상 깊었습니다. 실력 부족으로 풍부하게 구현하지 못한 게 아쉽지만, 각자 역할을 맡은 바를 해내는 과정에서 공부에 큰 도움이 된 것 같습니다.

유지원

객체지향 관련한 팀프로젝트를 진행하면서 각 팀원마다 잘 이해하지 못하는 부분들을 서로 알려주면서 코드를 완성해가는 과정이 좋았습니다. 또한 만들어온 코드들을 설명하면서 서로 보완해야할 부분들을 이야기하면서 주고받는 과정도 좋았습니다. 강의에서 배웠던 C++기능들을 알고 구현하는 부분에 있어서 이해도가 좀 떨어졌던 부분들이 많았던 것이 아쉬웠습니다.

**코드:**

