Song Player 명세서

프로그램의 구조는 application class에 Member List가 선언되어져 있다. Application class에서 메뉴를 출력하고 수행할 기능을 선택받아서 적절한 함수를 호출한다. 모든 자료들은 링크드리스트로 관리된다. 고 ItemType으로는 User, Album, Song이 있다. Application 안에는 UserManager list, AlbumManager list, SongManager list가 선언되어 있다.

프로그램 실행 시 메인 메뉴가 출력되고 메인 메뉴의 내용은 다음과 같다.

1. 로그인
2. 회원가입
3. 회원탈퇴
4. ID 찾기
5. PassWord 찾기
6. 프로그램 종료

로그인 선택 하여서 회원ID로 로그인할 경우 회원 메뉴를, 사전에 등록된 관리자ID로 로그인 할경우 관리메뉴를 출력 한다. 회원 메뉴는 다음과 같다.

1. 앨범추가
2. 앨범삭제
3. 앨범에 곡 추가
4. 앨범에서 곡 삭제
5. 앨범 검색
6. 앨범 출력
7. 음악 검색

관리자ID로 로그인 했을 경우 회원정보 및 앨범을 삭제, 음원의 추가 등을 할 수 있다.

모든 프로그램 내부의 모든 음원은 생성되는 순서대로 primary key를 배정 받아서 출력 시 게시 순서대로 출력하게 된다.

각 클래스의 ADT는 다음과 같다.

|  |
| --- |
| **#include “**Songmanager.h”  #include “AlbumManager.h”  #include “UserManager.h”  **class Application**  {  private:  Songmanager sm;  AlbumMabager am;  UserManager um;  public:  Application();  ~Application();  //===============================================================  // Funct:어플리케이션 구동  // pre :  // post :어플리케이션 구동  //===============================================================  void Run();  //===============================================================  // Funct:메인메뉴의 커맨드를 받아옴  // pre :  // post: 메인메뉴의 커맨드를 리턴  //===============================================================  int GetMainCommand();  //===============================================================  // Funct:유저 로그인 했을 때 메뉴  // pre :  // post :유저 메뉴 커맨드 리턴  //===============================================================  int GetUserCommand();  //===============================================================  // Funct:관리자 모드 로그인 했을 때 메뉴  // pre :  // post :관리자 메뉴 커맨드 리턴  //===============================================================  int GetManagerCommand();  //===============================================================  // Funct:회원가입  // pre :  // post :회원가입  //===============================================================  void Add\_U();  //===============================================================  // Funct:회원탈퇴  // pre :  // post :회원탈퇴  //===============================================================  void Delete\_U();  //===============================================================  // Funct:회원 검색  // pre :  // post :회원 검색  //===============================================================  void Search\_U();  //===============================================================  // Funct:로그인  // pre :  // post :멤버또는 관리자 로그인후 확인후 맞는 메뉴실행  //===============================================================  void LogIn();  //===============================================================  // Funct:로그아웃  // pre :  // post :로그아웃  //===============================================================  void LogOut();  //===============================================================  Void Add\_A ()//앨범 추가  Void Delete\_A()//앨범 삭제  Void Search\_A()//앨범 검색  Void Print\_A();//앨범 출력  Void Search\_S();//음원 검색  Void Add\_S()://음원을 플레이어에(SongManager) 추가  Void Add\_S\_At\_A();//음원을 유저이름과 앨범이름이 일치하는 앨범에 추가  Void Delete\_S\_From\_A();//유저이름과 앨범이름이 일치하는 앨범에서 음원을 삭제  }; |
| **Class Song**  **{**  **Private:**  **String S\_Name;** //음원 이름  **String S\_Singer**  **Int S\_Number;** //음원 번호  **Int S\_Play\_cnt;** //음원 재생횟수  **Int S\_Length;** //음원길이  **String S\_Genre** //음원 장르  **Song\* Song\_Next;**  **Public:**  //=====================================================================  // Funct: Song안의 S\_Play\_cnt = 0으로 초기화함  // pre :  // post :  //=====================================================================  Song()  //=====================================================================  // Funct: Song안의 S\_Name을 얻어옴  // pre : Song 객체 안에 S\_name에 대한 정보가 저장되어있어야 함.  // post : S\_name을 반환  //=====================================================================  String GetName();  //=====================================================================  // Funct: Song안의 S\_Number을 얻어옴  // pre : Song 객체 안에 S\_Number에 대한 정보가 저장되어있어야 함.  // post : S\_Number을 반환  //=====================================================================  Int GetNumber();  //=====================================================================  // Funct: Song안의 S\_Length를 얻어옴  // pre : Song 객체 안에 S\_Length에 대한 정보가 저장되어있어야 함.  // post : S\_Length을 반환  //=====================================================================  Int GetLength();  //=====================================================================  // Funct: Song안의 S\_Play\_cnt를 얻어옴  // pre : Song 객체 안에 S\_Play\_cnt에 대한 정보가 저장되어있어야 함.  // post : S\_Play\_cnt을 반환  //=====================================================================  INt GetPlaycnt();  //=====================================================================  // Funct: Song안의 S\_Gerne을 얻어옴  // pre : Song 객체 안에 S\_Gerne에 대한 정보가 저장되어있어야 함.  // post : S\_Gerne을 반환  //=====================================================================  String GetGerne();  //=====================================================================  // Funct: Song안의 S\_Singer을 얻어옴  // pre : Song 객체 안에 S\_Singer에 대한 정보가 저장되어있어야 함.  // post : S\_Singer을 반환  //=====================================================================  String GetSinger ();  //=====================================================================  // Funct: Song안의 S\_name을 키보드입력을 통해 설정  // pre :  // post : //=====================================================================  Void SetNameFromKB();  //=====================================================================  // Funct: Song안의 S\_Singer을 키보드입력을 통해 설정  // pre :  // post :  //=====================================================================  Void SetSingerFromKB();  //=====================================================================  // Funct:Song안의 S\_Gerne를 키보드입력을 통해 설정  // pre :  // post :  //=====================================================================  Void SetGerneFromKB();  //=====================================================================  // Funct:Song안의 S\_Length를 키보드입력을 통해 설정  // pre :  // post : //=====================================================================  Void SetLengthFromKB();  //=====================================================================  }; |
| #include “Song.h”  **Class Album**  **{**  **Private:**  **List<Song> song;**  **List<Song>::iterator S\_iter;**  //앨범내의 Song을 관리할 Song리스트  **String Album\_name;**  **String User\_name;** //앨범내의User\_name과 User class안의 U\_Name을 비교하여 앨범 주인을 파악함  **Album\* Album\_Next;**  **Public:**  //=====================================================================  // Funct:Album안의 User\_name을 키보드입력을 통해 설정  // pre :  // post :  //=====================================================================  **Void Set\_User\_NameKB()**  //=====================================================================  // Funct: Album안의 User\_name을 반환  // pre : User\_name이 초기화 및 저장되어있어야 함  // post :User\_name반환  //=====================================================================  **String Get\_User\_Name();**  //=====================================================================  // Funct: Album 안의 Song list인 Al에 노래를 추가한다.  // pre :  // post : data와 동일한 name,singer,gerne등을 가지고 있는 음원을 Al에 추가  //=====================================================================  **Void Add\_Song(Song& data);**  //=====================================================================  // Funct: Album 안의 Song list인 Al에 노래를 삭제한다.(Album에서만 지워지는 것)  // pre : 음원이 존재해야한다.  // post : data와 동일한 name,singer,gerne등을 가지고 있는 음원을 Al에서 삭제  //=====================================================================  **Void Delete\_Song(Song& data);**  //=====================================================================  // Funct: Album 안의 Song list인 Al에서 노래를 검색한후 출력한다.  // pre : 음원이 존재해야한다.  // post : data와 동일한 name,singer,gerne등을 가지고 있는 음원을 출력  //=====================================================================  **Void Search\_Song(Song& data);**  //=====================================================================  // Funct: Album안의 멤버변수인 Album\_name을 반환한다.  // pre : Album\_Name이 존재해야한다.  // post : Album\_name반환  //=====================================================================  String Get\_Album\_name();  //=====================================================================  // Funct: 키보드로부터 입력을 받아 Album 멤버변수인 Album\_name을 초기화한다.  // pre :  // post :  //=====================================================================  Void Set\_Album\_nameFromKB();  //=====================================================================  **};** |
| **Class User**  **{**  **Private:**  **String U\_Name;** //회원 이름  **String U\_Id;** //회원 id  **String U\_Password;** //회원 비밀번호  **User\* User\_Next;**  **Public:**  //=====================================================================  // Funct: U\_Name을 키보드로부터 입력받아서 초기화  // pre :  // post : //=====================================================================  Void SetU\_NameFromKB()  //=====================================================================  // Funct:Song U\_Id를 키보드로부터 입력받아서 초기화  // pre :  // post : //=====================================================================  Void SetU\_IdFromKB()  //=====================================================================  // Funct: U\_PassWord를 키보드로부터 입력받아서 초기화  // pre :  // post : //=====================================================================  Void SetU\_PasswordFromKB()  //=====================================================================  // Funct: User안의 U\_name을 얻어옴  // pre : User 객체 안에 U\_name에 대한 정보가 저장되어있어야 함.  // post : U\_name을 반환  //=====================================================================  **String GetU\_name();**  //=====================================================================  // Funct: User안의 U\_Id을 얻어옴  // pre : User 객체 안에 U\_Id에 대한 정보가 저장되어있어야 함.  // post : U\_Id을 반환  //=====================================================================  **String GetU\_Id();**  //=====================================================================  // Funct: User안의 U\_PassWord를 얻어옴  // pre : User 객체 안에 U\_ PassWord 에 대한 정보가 저장되어있어야 함.  // post : U\_ PassWord를 반환  //=====================================================================  **String GetU\_PassWord();**  //=====================================================================  **};** |
| **Class SongManager** //전체 음원을 관리하는 class  {  Private:  Song\* SongHead;  Song\* SongTail;  Song\* SongCur;      Public:  //노래추가,삭제,검색,정렬  //=====================================================================  // Funct:Song list인 sm에 Song을 추가  // pre :  // post : //=====================================================================  Void Add\_Song()  //=====================================================================  // Funct:Song list인 sm에 있는 Song을 삭제  // pre :  // post : Delete Song //=====================================================================  Void Delete\_Song()  //=====================================================================  // Funct:Song list인 sm에 있는 Song을 검색(한 곡)  // pre : 한 곡 이상의 Song이 sm에 저장되어 있어야 함  // post :Search Song //=====================================================================  Void Search\_Song()  //=====================================================================  // Funct:Song list인 sm에 있는 Song들 중 같은 S\_Singer를 가진 Song 검색(여러곡)  // pre : 한 곡 이상의 Song이 sm에 저장되어 있어야 함  // post : //=====================================================================  Void Search\_SongBySinger()  //=====================================================================    }; |
| **#include “Album.h”**  **Class AlbumManager**  **{**  Private:  Album\* AlbumHead;  Album\* AlbumTail;  Album\* AlbumCur;  Public:    Void Add\_Album(string name) //al에 앨범을 하나 추가하고 Album\_name을 name으로 초기화  Void Delete\_Album(string a\_name) //al에서 a\_name에 해당하는 앨범을 삭제  Void Search\_Album(string a\_name) //al에서 a\_name에 해당하는 앨범목록 출력  Void Search Album(string name) //al에 존재하는 Album들 중에서 User\_name이 name인 목록들을 출력  **};** |
| **Class UserManager**  **{**  Private:  User\* UserHead;  User\* UserTail;  User\* UserCur;  Public:    Void Add\_User() //us에 User를 하나 추가하고 키보드로부터 정보들을 입력받아 초기화  Void Delete\_User(string name) //us에서 name과 동일한 U\_Name을 가진 User를 삭제  Void Search\_User(string name) //us에서 name에 해당하는 User를 출력  **};** |

**프로그램을 만들어 감에 있어서 추가로 필요한 함수들은 구현하겠습니다.**