**비디오 대여 프로그램 설계**



**소프트웨어공학**

**권세진 교수**

**Ai소프트웨어학과 2학년**

**201920276**

**이봉규**

**목차**

**1.요구분석**

**-포함해야 할 기능**

**2.설계**

**-각 기능별 설계 flow chart**

**3.구현**

**-주요함수 설명**

**-각 기능별 구현**

**4.테스트 결과**

**-명령어 삽입 테스트 스크린샷**

**-결과분석**

**5.유지보수**

**-예외처리문**

**1. 요구분석**

**-가입자 / 비디오 추가**

가게에 새로운 비디오가 들어오거나 새로운 고객이 회원가입 하려고 할 때, 쉽고 간편하게 등록할 수 있어야 한다. 입력한 비디오나 가입자의 정보는 데이터에 저장되어 프로그램을 종료하여도 남아있으며, 후에 데이터를 불러와 활용할 수 있다.

**-가입자 / 비디오 검색**

가입한 고객의 정보를 찾거나, 가게에 등록되어 있는 비디오를 찾아야 할 때 사용자가 검색어를 입력하여 쉽고 간편하게 원하는 결과물을 찾을 수 있어야 한다.

가입자로 검색하면 가입자가 대여한 비디오의 목록도 알 수 있다.

**-비디오 대여**

가게에 있는 비디오를 가입자가 빌릴 수 있다. 데이터에 등록한 가입자여야 하며, 데이터에 등록되어 있는 가입자인지 확인한 후 비디오를 빌려갈 수 있다. 한 가입자가 비디오를 빌려가면 빌려간 가입자가 반납할 때까지 그 비디오는 다른 가입자가 빌릴 수 없다.

**-비디오 반납**

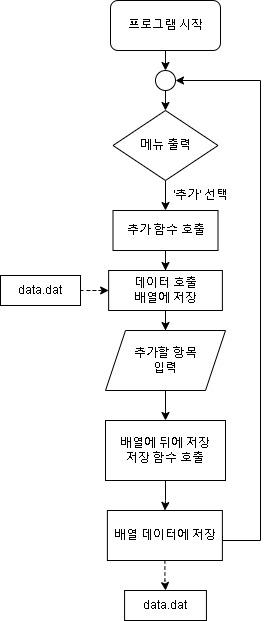
빌려간 비디오를 반납할 수 있다. 누군가가 대출한 비디오는 반납할 때 까지는 다른 사람이 대출 할 수 없으므로 반납이 필요하다. 대출할 때는 가입자 확인이 필요하지만, 반납할 때는 비디오만 있다면 손쉽게 반납할 수 있다.

**-비디오 / 가입자 목록 출력**

현재 가게에 어떤 비디오가 등록되어 있는지, 가입되어 있는 가입자는 누가 있으며 얼마나 있는지 목록으로 한번에 확인할 수 있다. 데이터에 저장되어 있는 정보를 활용함으로써 프로그램을 종료해도 데이터가 날아가지 않으며 수정이나 추가, 삭제한 그 즉시 메뉴의 목록 출력을 통하여 그때그때 바로 리스트를 확인할 수 있다.

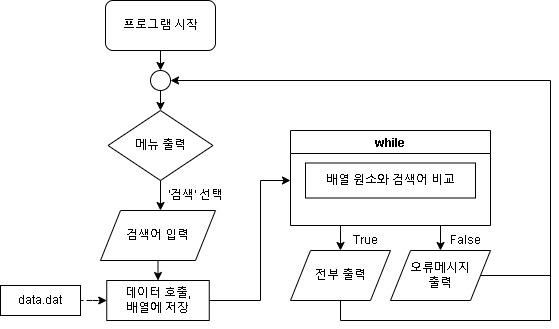
**-비디오 / 가입자 삭제**

데이터에 있는 비디오, 가입자 추가와 함께 삭제하는 기능이다. 필요 없는 비디오나 탈퇴하는 가입자를 데이터에서 삭제할 수 있다.

**2.설계**

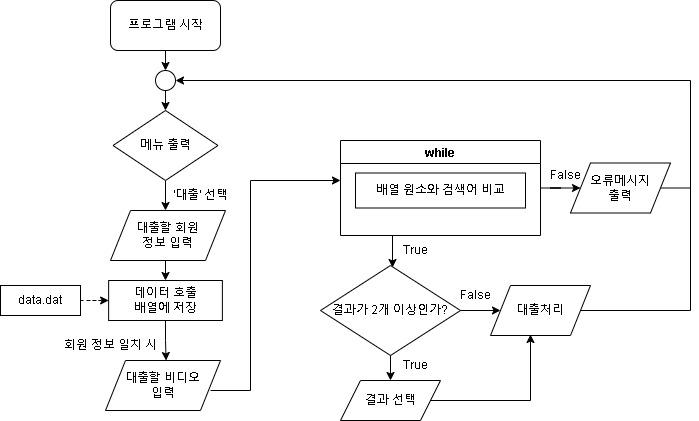
-**비디오 / 가입자 추가**

사용자가 메뉴에서 ‘추가’를 선택하면 사용자는 비디오를 추가할 것인지 가입자를 추가할 것인지 선택할 수 있다. 함수는 사용자에게 추가할 항목의 종류와, 그에 맞는 제목을 받아온 후, 데이터에게서 문자열을 받아 string형 벡터에 저장한다. 사용자에게 입력 받은 내용을 push\_back 한 이후에 다시 저장 함수에게 벡터를 전달하여 데이터에 저장하는 과정을 거치게 된다.

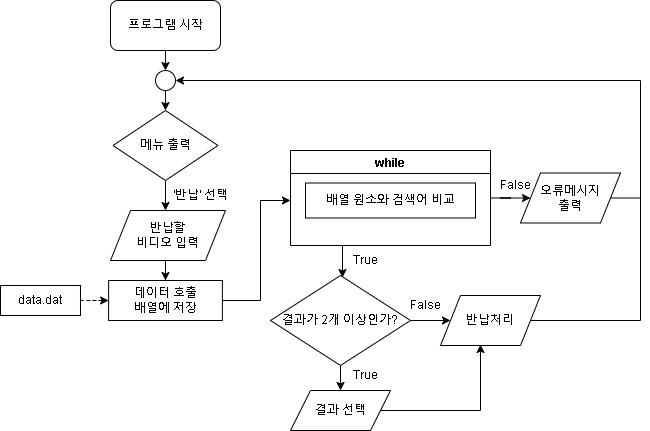
**-비디오 / 가입자 검색**

사용자가 메뉴에서 ‘검색’을 선택하면 함수는 사용자에게 검색할 항목과, 항목의 이름을 받는다.

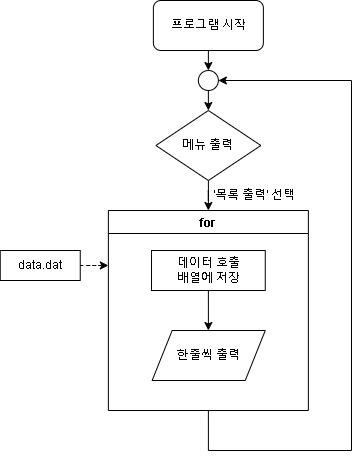
이후에 함수는 데이터에서 비디오 목록을 가져와 벡터에 저장한 후, 반복문을 통하여 벡터에 저장된 데이터와 사용자에게 받아온 검색어를 비교해가며 일치하는 항목을 찾는다. 만약 일치하는 항목이 있다면 리스트에 저장 후, 전부 출력한다. 일치하는 검색어가 없다면 결과를 찾을 수 없다는 오류메시지를 출력한 후, 검색 기능을 종료하게 된다.

**-비디오 대여**

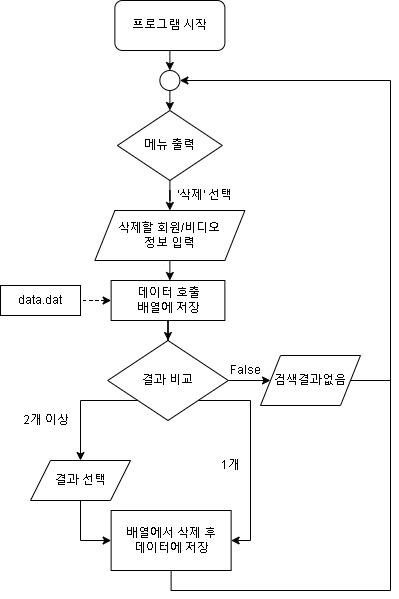
사용자가 메뉴에서 ‘대출’을 선택하면 함수는 사용자에게 회원의 정보를 입력 받는다. 검색 기능과 동일하게 함수는 회원 데이터를 받아온 후, 입력 받은 회원정보와 데이터에 저장 되어있는 회원정보가 일치하는지 검색한다. 일치하는 데이터가 확인된 후 사용자에게 대출할 비디오의 정보를 입력 받고 검색기능으로 데이터를 받아와 찾고 대출처리 시킨다.

**-비디오 반납**

사용자가 메뉴에서 ‘반납’을 선택하면 대출 기능과는 다르게 함수는 회원 정보를 받는 대신에 비디오의 정보만 받아온다. 이후 비디오 데이터를 벡터로 받아와 저장한다. 사용자에게 검색 결과를 출력하고, 선택한 비디오가 누군가가 대출한 비디오가 맞다면 자동으로 반납처리가 된다.

**-비디오 / 가입자 출력**

목록 출력을 입력 받으면 함수는 데이터에서 getline으로 한줄씩 벡터에 저장한다. 데이터의 줄마다 있는 대출 구분기호를 삭제하고 각각 번호를 붙여서 사용자에게 출력하게 된다.

**-비디오 / 가입자 삭제**

사용자에게 삭제 기능을 입력 받으면 함수는 사용자에게 검색할 항목과, 항목의 이름을 받는다.

이후에 함수는 데이터에서 비디오 목록을 가져와 벡터에 저장한 후, 반복문을 통하여 벡터에 저장된 데이터와 사용자에게 받아온 검색어를 비교해가며 일치하는 항목을 찾는다. 항목이 1개 이상이면 사용자에게 항목을 출력하고, 선택하여 삭제할 수 있다.

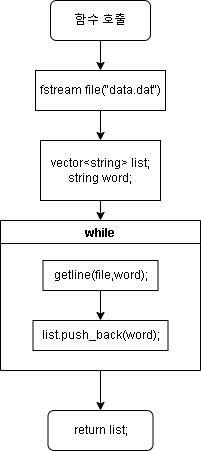
**3.구현**

**1. 중요함수 설명**

|  |
| --- |
| **Add\_info()**: 데이터에 새로운 비디오와 새로운 회원을 추가하는 함수. 사용자에게 입력 받아 데이터에 저장함.  **Seacrh\_guest(), Search\_video()**: 저장 되어있는 비디오, 회원 데이터에서 원하는 키워드로 검색하는 함수. 데이터에서 문자열을 벡터로 받아온 후 str.find()기능으로 찾는 데이터가 있는지 검색한다.  **rent\_video():** 가게에 있는 비디오를 빌릴 수 있는 함수. 손님의 회원 데이터가 있는지 확인한 후, 가게에 있는 비디오를 대출 처리한다.  **Print\_user(), Print\_video():** 데이터에 저장 되어있는 비디오와 회원의 정보를 출력해주는 함수. 데이터에서 문자열을 벡터로 받아온 후 사용자에게 한줄씩 출력한다.  **return\_video()**: 빌린 비디오를 다시 반납처리해주는 함수. 대출과는 다르게 회원의 정보가 없이 비디오의 정보만으로 반납처리를 할 수 있다.  **Del\_info()**: 데이터에 저장되어 있는 회원의 정보나 비디오의 정보를 삭제해주는 함수. 데이터에서 문자열을 벡터로 받아온 후 str.find()기능으로 찾는 데이터가 있는지 검색한다. 그 후에 해당 데이터를 삭제하고 저장한다.  **Return\_guestlist(), Return\_videolist():** 데이터에 직접적으로 접근하여 벡터에 비디오나 회원의 리스트를 벡터에 넣고 돌려주는 함수. 비디오 리스트는 자동적으로 비디오가 대출 중인지 구분하는 문자를 삭제하고 돌려주게 됨.  **Return\_rawvideolist()**: 기본적으로는 위 함수와 기능이 같으나, 이 함수에서는 대출 구분 문자를 삭제하지 않고 원형을 돌려주게 됨.  **save\_guestdata(), save\_videodata()**: 수정이 완료된 벡터형 문자열을 받아오고 그것을 다시 데이터에 저장하는 함수.  **Search\_guesttool(), Search\_videotool():** Search\_\_함수에서 이용하는 함수이며 문자열 벡터와 찾는 검색어를 받으면 벡터 안을 반복문으로 검색어가 포함된 원소를 찾아주는 함수이다. 필요시에 대출 구분문자 부분도 지워서 리턴 할 수 있다. |

**2.함수 구현부분 설명**

**-데이터 -> 문자열**

**Return\_-list()**

1.함수가 호출되면 저장되어 있는 데이터를 불러온다.

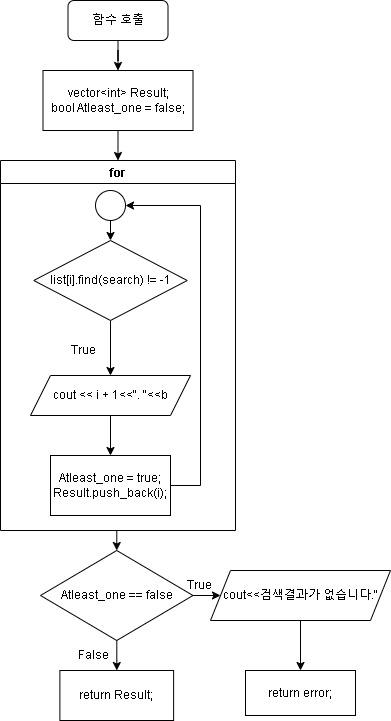
2.문자열을 담을 벡터와 string형 변수를 선언한다.

3.데이터에서 한 줄씩 불러오며 벡터에 저장한다.

4.데이터의 마지막에 도달하면 반복문을 종료하고 벡터를 리턴 한다.

**-검색툴**

**vector<int> Search\_-tool(vector<string>, string)**

1.함수가 호출되면 저장되어 있는 데이터를 불러온다.

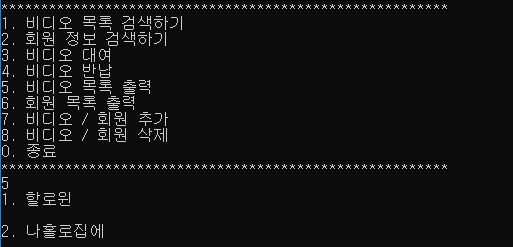
2.인덱스를 담을 벡터와 한개 이상의 검색결과가 있는지 검사하는 bool형 변수를 선언한다.

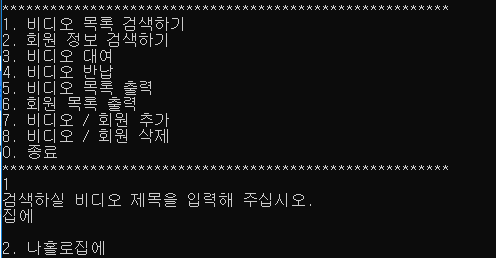
3.인자로 들어온 문자열 벡터를 반복문으로 돌며 검색어가 포함되어 있는 원소를 찾고, 찾을 시에는 해당 인덱스를 결과 벡터에 넣는다.

4.검색 결과가 하나 이상 있을 경우 인덱스가 들어간 벡터를 반환하며 결과가 없을 경우 지정기호가 들어간 벡터를 반환한다.

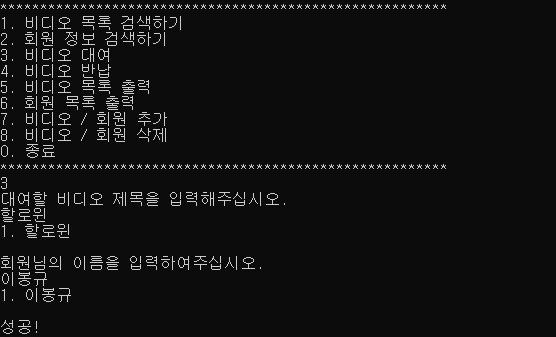
**4.테스트 결과**

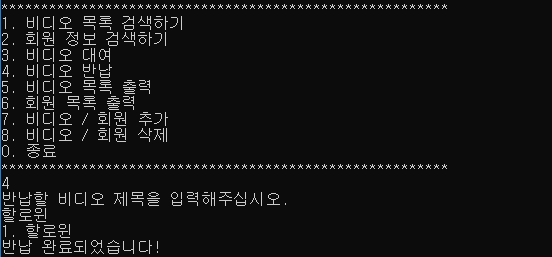
**-명령어 삽입 테스트 스크린샷**

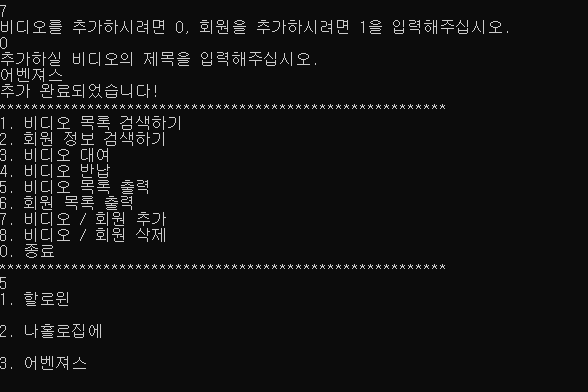
-목록출력



-비디오 검색

-비디오 대여

-비디오 반납

-비디오 추가

**2.결과분석**

-설계 때 넣고 싶었던 기능인 사용자를 검색하면 그 사용자가 대출한 비디오를 보여주는 기능을 추가하지 못하여 아쉬웠다.

-간헐적으로 데이터에 빈 공간이 들어가는 일이 있는데 그것을 고치지 못해 아쉬웠다. 빈 공간이 있으면 그 줄은 띄어서 출력하거나, 데이터를 저장 할때 그 줄은 빼고 저장하는 등의 기초적인 방법은 마련했지만 근본적으로 해결은 하지 못했다.

-반납 처리시에 비디오데이터 마지막줄에 알 수 없는 오류 문자가 들어 가는 게 확인되었으나 해결 방안을 찾지 못했다. 대안으로 오류 문자를 무시하고 쓰고 읽을 수 있게 처리했으나 완전히 해결하지는 못하였다.

**5.유지 보수**

**-예외 처리문**

-**if(!file)**

데이터에 직접적으로 접근하여 받아오는 함수에 들어가 있으며 만약 어떠한 오류로 인하여 저장되어 있는 데이터가 열리지 않을 경우 에러코드를 반환시키고 해당 기능을 중지시킨다.

**-if(array[i].size > 5)**

데이터에 수정된 문자열을 저장할 때, 아무것도 쓰여 있지 않은 빈 줄을 저장하는 것을 막는다. 또한 알 수 없는 이유로 생기는 오류 문자열을 중복 저장하는 것을 막는다.

**-for (int i = 0; i < return\_choice.size(); i++){**

**if (video\_choice == return\_choice[i]) break;**

**if (i == return\_choice.size()) }**

검색이나 기타 기능으로 검색 결과가 2개 이상 출력 되었을때, 사용자가 주어진 검색 결과 이외의 인덱스를 입력 하였을시에 검사하는 구문이다. 검색툴 함수가 검색 결과의 인덱스가 들어간 벡터를 반환하게 되고, 사용자가 입력한 인덱스를 비교하여 입력 받은 인덱스가 실제로 검색 결과로 반환된 인덱스에 포함되어 있는지 검사하게 되는 반복문이다.