

최종 보고서

1조(한순규,송찬환,임준규,이연정)

1. 프로젝트 주제 및 목적

주제 : 도서관 관리 시스템

목적 : 관리자 권한으로 책을 추가, 삭제, 검색 회원 관리를 하고 회원 권한으로 책 대출, 반납을 한다.

2. 팀원 구성 및 역할

한순규 : 파일 입출력 구현 및 Model의 도서 검색 메소드 구현

송찬환 : View 구현 및 전체적 원활한 구동을 위한 디자인, 추상클래스 구현

임준규 : Controller 구현 및 버그 픽스

이연정 : 도서 대출 및 반납 등 Model의 메소드 구현

3. 진행 사항 보고

관리자의 계정으로 접속하면 관리자가 관리할 수 있는 페이지로 갈 수 있다. 최초의 관리자 페이지에는 사용자 관리와 책 관리를 선택할 수 있도록 이루어져 있으며, 각 메뉴를 선택하여, 추가와 삭제를 수행할 수 있다.

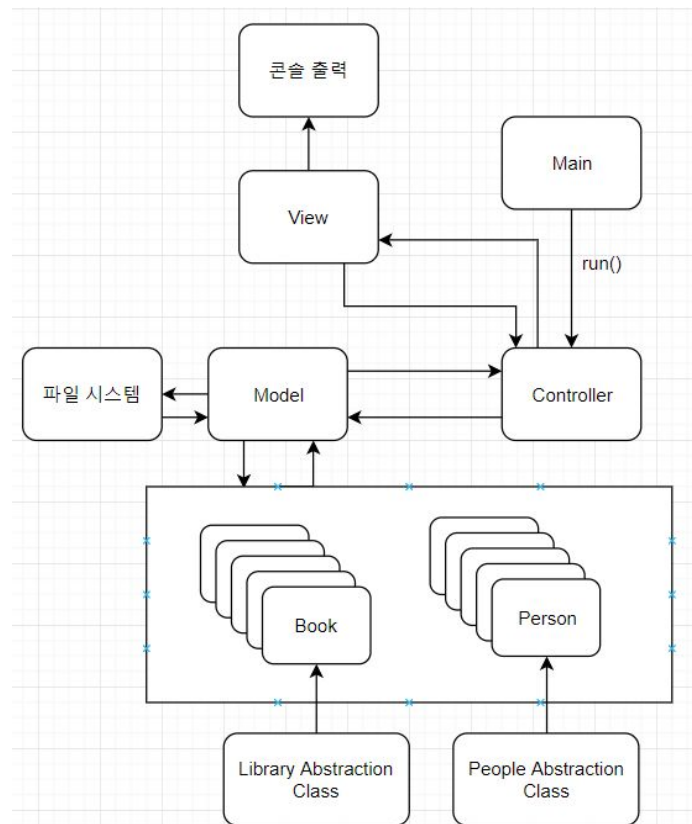
관리자 계정을 통하여 추가하거나 제거한 책과 사용자는 txt파일을 데이터 베이스처럼 이용하여 저장된다. 이 데이터들은 최초 실행할 때 불러와서 파일 내의 list에 저장되고, 어떠한 작업을 할 때마다 파일 입출력을 통하여 최신화된다.

위와 같이 추가된 책과 사용자 계정을 이용하여 user페이지에 접속할 수 있는데, 사용자는 관리자가 추가해준 ID와 PassWord를 이용하여 접속할 수 있다. 최초의 사용자 페이지는 책의 검색/대출과 반납으로 이루어져 있다. 검색 후 검색한 고유번호를 바탕으로 대출을 진행할 수 있다. 대출 후 반납으로 가면 페이지에 대출한 책들의 고유번호들이 뜨고, 그 고유번호 중 선택하여 입력하면 반납처리가 완료된다.

위와 같이 최초에 보고한 기능들은 구현을 완료하였으며, txt파일을 데이터베이스로 이용하여 다시 구현하여도 이전의 Data를 유지할 수 있도록 하였다.

4. 프로그램 설계

- 구조



MVC 구조로, view객체는 콘솔출력을 담당하고, model객체는 데이터의 저장, 수정, 검색, 삭제 등의 일을 담당하면서, 파일 시스템에 데이터를 저장하고, 읽어온다.

Controller 객체는 view와 model사이를 제어하는 역할을 하고,

Book 객체는 책에 대한 데이터가 저장되는 부분으로 model에서 linked list로 만들어 관리한다. person객체는 회원에 대한 데이터가 저장되는 객체로 마찬가지로 model에서 person에 대한 linked list로 만들어서 관리한다.

- 주요 기능

- 로그인
- 파일 입출력
- 대출
- 반납
- 검색
- 도서, 회원 추가

- 각 클래스 별 명세

* Model Class

- bool loginCheck(std::string id, std::string passwd)
 - 로그인을 확인하는 함수. id와 passwd를 받아서 회원정보의 id와 passwd가 일치하면 true를 리턴하고 일치하지 않으면 false를 리턴한다.
- bool bookInsert(std::string& addbookTitle, std::string& addauthor, std::string& addpublisher, int& addbookNumber)

- 책 추가하는 함수. 추가하고 싶은 책 이름, 작가, 출판사, 색인을 받아서 새로운 책 객체를 만들어서 그 객체를 booklist에 넣어준다. 색인이 같은 책이 있다면 false를 반환한다.
- bool addPerson(std::string& name, std::string& id, std::string& passwd)
 - 회원 추가 하는 함수. 회원의 이름, 아이디, 패스워드를 받아서 person 객체를 만들어서 그 객체를 personlist에 넣어준다. 만약 입력 받은 id가 이미 있다면 false를 리턴해준다.
- person idToPerson(std::string id)
 - id로 person객체 찾는 함수. 입력 받은 아이디와 같은 person객체를 반환 해준다.
- std::list<Book> bookSearchByNumber(int givenBookNumber)
 - 색인으로 책 찾는 함수.
- std::list<Book> bookSearchByTitle(std::string title)
 - 책 제목으로 책을 찾는 함수.
- std::list<person> personSearch(std::string id)
 - id로 사람 찾는 함수.
- bool bookDelete(int bookNumber)
 - 색인으로 책 삭제하는 함수. 입력 받은 색인과 같은 책이 있다면 booklist에서 지워준다.
- bool bookLoan(std::string id, int bookNumber)
 - 색인과 아이디로 책을 대출하는 함수. 로그인 한 아이디와 책의 색인을 입력 받아서 그 책이 있으면 받은 아이디에 책의 색인을 저장해준다. 저장에 성공하면 true를 리턴해준다. 대출 하지 못하면 false를 반환한다.
- bool bookReturn(std::string id, int bookNumber)
 - 색인과 아이디로 책을 반납하는 함수. 로그인 한 아이디와 책의 색인을 입력 받아서 그 책이 있는지 확인하고 그 아이디에 입력 받은 아이디의 빌린 목록을 확인해서 있으면 반납을 해준다. 반납에 성공하면 true를 반환해주고 성공하지 못하면 false를 반환한다.
- bool personDelete(std::string id)
 - 회원 삭제하는 함수. 입력 받은 아이디와 같은 아이디가 있다면 그 아이디를 personlist에서 지워준다.
- void fileWrite()
 - 파일에 쓰는 함수
- void fileRead()
 - 파일을 읽는 함수
- * **Controller Class**
 - run()
 - 프로그램을 시작하게 해 주는 메소드로 프로그램의 흐름을 제어한다.

* View Class

- void initScreen(std::string& Id, std::string& passwd)
 - 초기 화면을 띄워주는 메소드
- void userPage(int& input)
 - 유저로 로그인 했을 때 user에게 띄워주는 초기 화면
- void searchBook(std::string& bookTitle)
 - 유저가 도서검색을 선택했을 때 보여주는 화면
- void loanBook(std::list<Book> searchBooklist, int& bookNumber)
 - 유저가 도서검색 후 대출을 할 수 있도록 띄워주는 페이지
- void returnBook(person Person, int& bookNumber)
 - 유저가 반납을 선택했을 때 보여주는 화면
- void managePage(int& input)
 - 관리자로 접속했을 때 보여주는 화면

- void manageBook(int& input)
 - 관리자가 도서관리를 선택했을 때 보여주는 화면
- void managePerson(int& input)
 - 관리자가 회원관리를 선택했을 때 보여주는 화면
- void addBook(std::string& bookTitle, std::string& author, std::string& publisher, int& bookN)
 - 관리자가 도서추가 선택시 도서에 대한 정보를 입력받는 화면
- void addPerson(std::string& name, std::string& id, std::string& pwd)
 - 관리자가 회원추가 선택시 회원에 대한 정보를 입력받는 화면
- void deletePerson(std::string& deleteId)
 - 관리자가 회원삭제를 했을 때 삭제에 대한 정보를 입력받는 화면
- void deleteBook(int& bookNumber)
 - 관리자가 도서삭제를 했을 때 삭제에 대한 정보를 입력받는 화면
- void faultPage(), void fail(), void success()
 - 정보입력이나 저장에서 처리가 잘 되거나 잘 되지 않았을 때 실패, 성공 등의 메시지를 화면에 출력해 주는 메소드

* Library abstraction Class

- 도서관에 있는 것들에 대한 추상클래스

* book Class :Library

- 도서에 대한 정보가 저장되는 클래스
- 필드 :
 - std::string bookTitle;
 - std::string author;
 - std::string publisher;
 - int bookNumber;
 - bool loan;
- 메소드 :
 - 각 필드에 대한 getter()와 setter()

*People abstraction Class

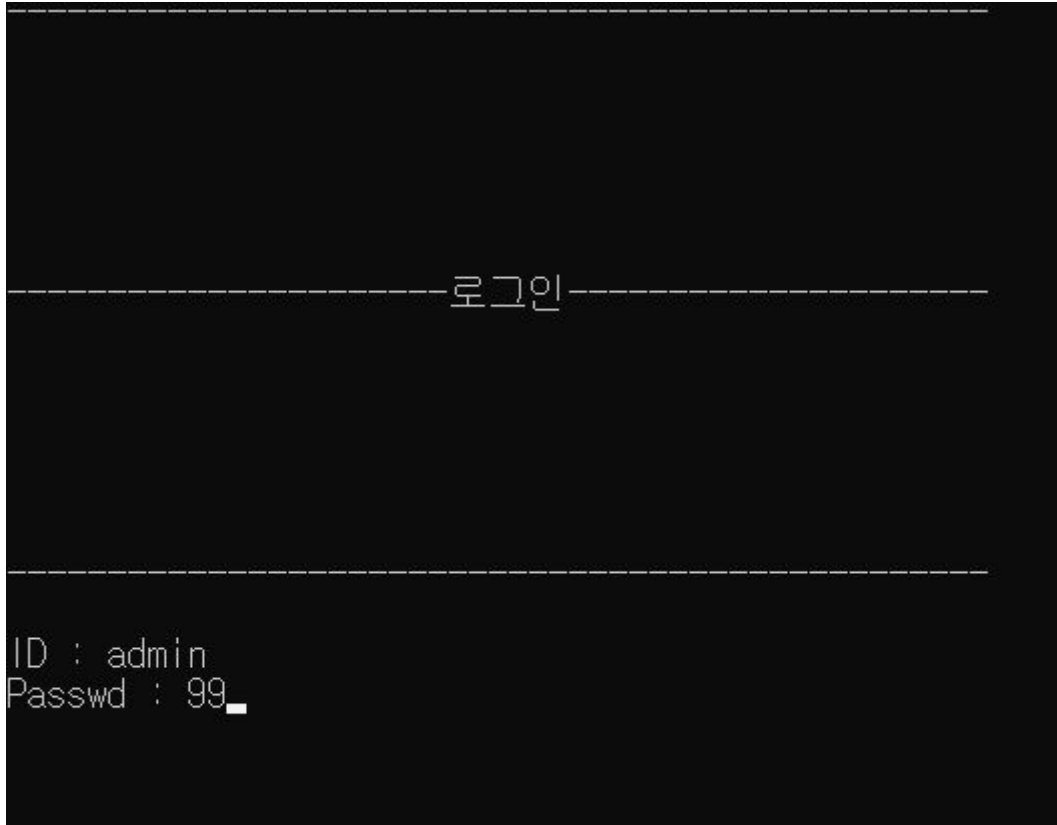
- 사람에 대한 추상클래스

* person Class :People

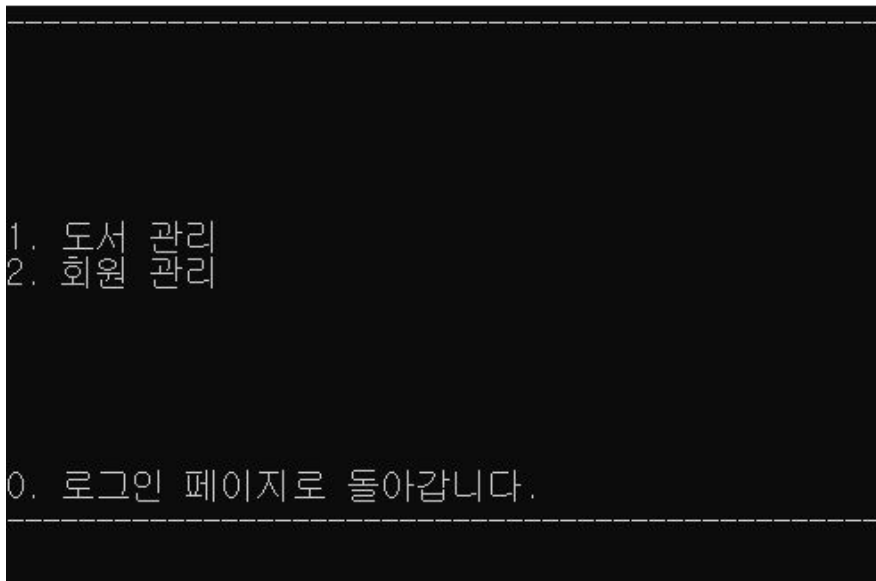
- 회원에 대한 정보가 저장되는 클래스
- 필드 :
 - std::string name;
 - std::string id;
 - std::string passwd;
 - int loanNumber;
 - std::string loanBook1;
 - std::string loanBook2;
 - std::string loanBook3;
- 메소드 :
 - 각 필드에 대한 getter()와 setter() 메소드

5. 실험

- 프로그램 실행하여 기능에 대한 검증
- 중요 인풋과 프로그램 동작에 대한 스크린 샷 등 자유 구성



시작 페이지에서 관리자로 로그인



입력: 1

1. 도서 추가
2. 도서 삭제

0. 이전 페이지로 돌아갑니다.

도서 관리페이지



booklist - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말

보노보노|보노보노|보노보노|898|1

123|123|123|4412|0

현재 booklist.txt 내용

입력: 2

2. 도서 삭제를 선택하셨습니다.

삭제할 책의 고유번호를 입력해주세요 : 4412

성공했습니다.



booklist - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말

|보노보노|보노보노|보노보노|898|1

삭제후 booklist.txt 내용.

2. 도서 삭제를 선택하셨습니다.

삭제할 책의 고유번호를 입력해주세요 : 4412

실패했습니다.

없는 책 삭제 예외처리


입력: 1

1. 도서 추가를 선택하셨습니다.

추가할 도서의 제목을 입력해 주세요 : 새로운책
추가할 도서의 작가를 입력해 주세요 : 신인작가
추가할 도서의 출판사를 입력해 주세요 : 신생출판사
추가할 도서의 책번호를 입력해 주세요 : 4

성공했습니다.

도서 추가 성공

 booklist - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말

보노보노|보노보노|보노보노|898|1

새로운책|신인작가|신생출판사|4|0

현재 booklist.txt 내용

입력: 0

입력: 2

- 1. 회원 추가
- 2. 삭제

0. 이전 페이지로 돌아갑니다.

회원 관리페이지

입력: 1

1. 회원 추가를 선택하셨습니다.

추가할 회원의 이름을 입력해 주세요 : 홍길동
추가할 회원의 ID를 입력해 주세요 : hong
추가할 회원의 비밀번호를 입력해 주세요 : 44323

성공했습니다.



personlist - Windows 메모장

파일(F) 편집(E) 서식(O) 보

이이이이||

김정은|kim|99|이||

홍길동|hong|44323|이||

현재 personlist.txt 내용

입력: 2

2. 회원 삭제를 선택하셨습니다.

삭제할 회원의 id를 입력해 주세요 :hong

성공했습니다.

personlist - Windows 메모장

파일(F) 편집(E) 서식(O) 보기

이이이이||

김정은|kim|99|0||

삭제명령 이후의 personlist.txt 내용

입력: 0

입력: 0

```
-----로그인-----
ID : 0
Passwd : 0
```

id가 0인 회원으로 로그인

- 1. 도서 검색 / 대출
- 2. 도서 반납

0. 로그인 페이지로 돌아가기

1

로그인된 페이지

입력: 1

1. 도서 검색 / 대출을 선택하셨습니다.

검색할 책을 입력해 주세요 : 나온

고유 번호 : 4
제목 : 새로운책
저자 : 신인작가
출판사 : 신생출판사
대출 여부 : 대출 가능

대출할 책의 고유번호를 입력해 주세요 : 4

성공했습니다.



personlist - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V)

이이이이||

김정은|kim|99|0||

홍길동|hong|44323|1|4||

id가 hong인 사람의 대출권수가 1 증가하고 대출된 책의 고유번호(4)가 저장됨.



booklist - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말

|보노보노|보노보노|보노보노|898|0

새로나온책|신인작가|신생출판사|4|1

대출된 책의 대출상태가 0에서 1로 변경됨.


입력: 2

4

반납할 책의 고유번호를 입력해주세요 : 4

성공했습니다.

대출중인 책의 고유번호가 표시됨. 4 입력하여 반납처리.

 personlist - Windows 메모장


파일(F) 편집(E) 서식(O) 보기

이이이이||

김정은|kim|99|0||

홍길동|hong|44323|0||

id가 hong인 사람의 대출권수가 0으로, 대출중인 책은 없어짐.

 booklist - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말

|보노보노|보노보노|보노보노|898|0

새로나온책|신인작가|신생출판사|4|0

booklist 에서도 대출상태가 0으로 바뀜.

6. 결과

- 팀 프로젝트 소감 및 미래 계획

-- 어려웠던 점

보다 큰 프로젝트에서 여러 인원들이 많은 메소드들을 구현하여 프로젝트를 하며 메소드들의 사용에 어려움을 느꼈으며, 개발자들 간의 의견이나 구현 방식에서 차이가 있을 수 있음을 느꼈습니다.

-- 개선할 점 및 앞으로 추가 구현하면 좋을 부분

현재로서는 대출이 가능한 Library클래스를 상속받는 객체가 Book하나 밖에 없지만, 최근의 도서관은 빌릴 object들이 그냥 Book만이 아닌 오디오나 영화같은 색다른 객체들도 존재할 수 있으리라 생각한다. 이후에 이러한 클래스들이 Library를 상속받아 대출할 수 있는 또하나의 객체로 손쉽게 추가시켜서 사용하면 더욱 프로그램의 활용의 범위가 늘어나리라 생각한다. 또한, 사용한 people 객체 또한 추상클래스를 이용하여 나눠두었는데, 후에 관리자의 계정을 추가하여, 특별한 작업을 수행할 수 있으며, 유저의 성질도 또 다르게 지정할 수 있도록 한다. 예를 들어 연체고객의 패널티를 부여할 수 있고, 우수고객에게 이점을 제공할 수도 있을 것이다.

현재로서는 콘솔을 이용하여 디자인 하는 방식으로만 구현을 하였으나, 후에 GUI를 이용하여 더 직관적으로 디자인하여 사용자들의 이용에 더욱 도움이 되도록 구현이 필요할 수 있음을 느꼈다.