1조: 회의록(1일차)

> 💡 **프로젝트 공지사항**

1. **진행기간**
    - 평균적으로 4~5일 가량 진행
2. **주제:** 자판기(Vending Machine)
    - 어떤 품목을 다루는 자판기인지?
    - 어떤 기능을 탑재한 자판기인지?
        - 팀별로 회의하여 선정하되, 게임X 선정성X 장난X
        - ex) 매출 총액, 칵테일 제조 품목, 재고관리
    - 자바 기초 과목에서 공부한 내용을 모두 적용시킬 수 있도록 구성
    - 반드시 각 팀 자판기의 핵심 기능이 뭔지 파악/공유할 것
3. **제출 항목**
    - **기획안** (⇒ 권순모, 정가연)
        - Flowchart (⇒ 김진희, 정효진, 최수지)
        - 예상구현화면 (⇒ 최수지)
    - 회의록 (⇒ 정미화)
    - 결과물
    - 발표자료 / 보고자료(각 팀원들 후기 포함)

주제 회의

- 도서관
    - 책(대출/반납, 책 정보 기재) : **핵심 기능**
    - 자료검색 : **핵심기능**

- 도서관 좌석 지정 프로그램
- 5분짜리 읽을 거리 출력
- 로그인 → 본인인증 기능
    - 공공도서관: 회원가입, 로그인, 비회원 선택
      학교도서관: 학생 인증

- 카페 키오스크(엑스트라, 매출관리)
- 꽃 자판기(문구 넣어서 엽서 뽑기, 포장)

메인 테마 결정 >>도서관<<

도서관 관련 구현 목록

1. 메인 선택메뉴
    a. 이용자 : 회원가입/ 로그인
    b. 관리자 : 로그인

2. 이용자 선택메뉴
    a. 신착도서 확인
    b. 대출/반납: 이미 대출한 자료는 대출 불가
    c. 자료검색: 장르별 열람 가능
    d. 희망도서 신청
    e. 마이페이지: 대출중인 자료 확인, 연체 및 이용중지 현황 확인
    f. 추가 구현 항목
        i. 도서예약: 도서대출현황 확인
        ii. 열람실 좌석 현황: 좌석 선택 → 입퇴실

3. 관리자 메뉴 선택
    a. 도서 추가
    b. 도서 현황
    c. 희망도서 현황
    d. 이용자 현황
    e. 추가 구현 항목
        i. 열람실 관리

기능 구동

[메인 화면] - 이용자 모드

- 회원가입 → 아이디 설정 → 중복 확인
    - 중복 X: 설정완료→ 비밀번호 설정 → 설정완료 → 회원가입 완료
    - 중복 O: 다시 입력
- 로그인 → 아이디 입력 → 아이디 확인 → 비밀번호 입력 → 비밀번호 확인 → 로그인 성공
- 로그인 후 아이디에 해당하는 정보 끌어오기

[이용 메뉴] - 이용자 모드

1. 도서 대출
    - 자료검색
        - → 존재 → 대출신청 → 회원 확인(잔여 대출 가능 권수, 연체된 도서 존재 여부 등) → 대출 성공
        - → 없음
            - → 메뉴로 돌아가기
            - → 도서 입고 신청
2. 도서 반납 → 도서 ISBN 번호 입력 → 반납 완료
3. 마이페이지

- 기본 정보 표시(아이디, 대출 가능 권수 등)
- 대출 중인 도서 → 목록, 각 책의 반납 기한 표기
- 내가 신청한 도서 → 신청완료/ 입고완료/입고불가(절판 등) 등

[메인 화면] - 관리자 모드(관리자는 아이디 1개 가능)

- 관리자 아이디, 비밀번호 입력 → 확인 → 접근 성공

[이용 메뉴] - 관리자 모드

1. 도서 대출 현황
   - 자료 검색 → 검색한 도서의 대출 현황(대출 여부, 반납예정일)
2. 도서 관리
   - 도서 추가 → 제목, 저자 등 입력 → 신착도서 목록에 추가 → 완료
   - 신착도서목록(책이 추가되면 등록일자를 활용하여 자동으로 갱신)
   - 회원신청도서목록(책이 추가 등록되면 자동으로 삭제)

변수, 컬렉션, 클래스, 메소드

```
변수
final int 대여가능한 책의 권수
final int 모든 사람들의 대출기한(예: 14일)
final int 1일 벌금(예: 300원)
final int 누적 연체료

컬렉션
Hashtable<String(id), 회원> 회원목록
HashSet 도서목록
ArrayList 대출반납현황                // 날짜 순서로 정렬

VO클래스
class 회원
{
  String id
  String pw
  String 이름
  String 주민등록번호      // 유효성 검사();
  int 대출 중인 권 수
  boolean 연체 여부
```

```
  int 이용정지 날 수
}

class 책
{
  String 제목
  String 저자
  boolean 대출현황
  Calendar 입고일자       // 신간도서 출력();
}

class 대출 정보
{
  Calendar 대출일자
  Calendar 반납일자       // 연체일수 계산();
  책
  회원
}

//class 좌석
//{
//  좌석번호
//  boolean 입실현황
//}

=====================================================================

[0] 메뉴선택
void printMenu
{
  1. 이용자
  2. 관리자
}

void 이용자선택메뉴
{
  ...;
}

void 관리자선택메뉴
{
  ...;
}


[1] 이용자: 회원가입/로그인
void 회원가입
{
  아이디 입력 → 중복 확인 → 설정 완료
  비밀번호 입력 → 설정 완료
  이름 입력
  주민등록번호 입력 → 유효성검사 → 설정 완료
}

void 로그인
{
  ...;
}
```

```
void 자료 검색
{
  ...;
}

void 자료 대출
{
  대출가능여부확인();
  자료검색();
}

void 자료 반납
{
  자료검색(); => 대출현황 변경
  대출잔여일수(); => 대출정지여부...
}

int 대출잔여일수
{
  ...;
}
```

1조: 회의록(2일차)

기획서 1차 피드백 내용

1. **프로그램 실행 시**
    - 일단 관리자 로그인 후, 이용자 모드로 선택 가능
      프로그램 종료도 관리자모드만 가능
      이용자모드 시 관리자모드로 돌아가기 기능 추가(재로그인 필요)

2. **비용(연체료) 관련 사항**
    - 화폐권종별관리(수입/지출)
    - 총연체료 확인
    - 잔돈관리
    - ⇒ 1일 300원

3. **회원정보관리**
    - 각 회원의 1) 연체 현황, 2) 대여현황 등에 대한 내용 필요
    - 회원정보관리 ⇒ "회원정보 조회 및 삭제" 로 변경
    - 조회항목: 아이디, 비밀번호, 이름, 주민번호, 대출도서정보, 연체정보, 누적연체료

4. **필요 없는 항목 제외**

5. **주요 클래스 및 메소드 정리**

    ■ Library 추상 클래스
        - 로그인
        - 서비스선택(Admin/User 에 따라 재정의)
        - 신착 도서 목록 조회

    ■ User 클래스 extends Library 추상클래스

- static Hashtable<String(id), 회원VO클래스> 회원list;
- 로그인 메소드 재정의
- 서비스선택 메소드 재정의
- 신간 도서 목록 조회(출력) 메소드
- 반납 메소드
- 대출 메소드
- 희망도서 신청 메소드
- 마이페이지
    - 내 정보 조회(출력) 메소드
    - 내 대출 현황 조회(출력) 메소드
    - 희망도서 신청 현황 조회(출력) 메소드

■ Admin 클래스 extends Library 추상클래스

- static final 필드 → 관리자의 아이디 비밀번호
- 로그인 메소드 재정의
- 서비스선택 메소드 재정의
- 신간 도서 목록 조회(출력) 메소드
- 대출 현황 조회(출력) 메소드
- 회원신청 도서 목록 조회(출력)메소드
- 도서 추가 메소드

■ books 클래스 extends 추상클래스

- static Hashtable<String(책제목), 책VO클래스> 도서목록;
- static ArrayList<VO대출정보> 대출반납현황;

■ books 클래스 extends 추상클래스

- static Hashtable<String(책제목), 책VO클래스> 도서목록;
- static ArrayList<VO대출정보> 대출반납현황;

■ 도서 대출 시스템 클래스(main 클래스)

기획안 2차 피드백 내용

1. 대출 가능 여부 확인 세분화
2. 책 정보 세분화
    - 제목 / 저자 / 출판사 / 출판년도 / 분류번호 / 카테고리(소설, 동화, 잡지 등)
    - 검색 유형 세분화 : 책 정보를 기반으로
    - ⇒검색 결과 세분화 : 책 정보 / 대출 여부 / (대출 불가능할 시) 반납예정일

⇒ 회원 속성 컨트롤 세분화

- 연체중인 회원 목록 확인

기획안 통과 후, 조 나눠서 회의

- 회의 주제: 프로그램에 어떤 기능을 넣을 것인지
- 조 활동
    - 관리자조: 정가연, 권순모, 정미화
    - 이용자조: 김진희, 정효진, 최수지

1조: 회의록(3일차)

조 회의 내용 공유 및 논의 진행

- **관리자 조**
  - **연체료 관리 부분**
    - 자판기 내 거스름돈의 권종별 관리를 위해 VO클래스 추가
    - 권종의 표기방법 통일 ⇒ 10,000원 (O) , 1만원 (X)
    - 거스름돈 메뉴 통합 ⇒ 거스름돈 조회 및 입출금 (O), 거스름돈 조회, 거스름돈 입출금(X)
    - 총 수익금 확인을 위해 '누적 연체료' 변수 추가

- **이용자 조**
  - **희망도서 신청 부분**
    - 희망도서 신청 시 받을 정보 값 ⇒ 도서명, 저자명으로 한정
    - 희망도서 신청 목록을 관리할 자료구조 혹은 배열 필요
      - 희망도서 VO클래스 구성 : 도서명, 저자명, 신청일자, 신청자, 신청상태
      - 희망도서 자료구조(ArrayList) 구성
    - 관리자 모드에서 희망도서를 컨트롤(신청 거부)할 수 있는 항목 추가

  - **도서검색 세분화**
    - 도서명, 저자명, 출판사 등 세부 사항을 골라 검색 가능

프로젝트 발표 관련 추가 공지사항

- **발표 일자: 화요일**
  - 단, 개인 코딩은 일요일까지 마칠 수 있도록
  - 월요일은 팀원별 코드 내용 공유 및 수정, 발표 준비

- **발표 방식**
  - PPT
    - 주제, 설계 방식, 접근 방식, 진행 방식 등...
    - 코드: 필요하다면 추가하고, 필요하지 않다면 제외해도 됨
  - 프로그램 시연
    - 아마도 대부분의 발표 시간을 할애할 것임
  - 발표자
    - 인원도 방식도 자유! 자율적으로 정할 것.

주요 클래스 및 메소드 논의

- 논의 자료
- 주요 논의 사항
  - 메인 메소드가 포함될 클래스

```
class Main
{
  final int RENT_BOOKS = 5;   // 대여 가능한 책의 권수: 5권
  final int RENT_DAYS = 14;   // 대출 기간: 14일
  final int LATE_FEE = 300;   // 1일당 연체료: 300원

  public static void main(String[] args)
  {
    Hashtable<String, Members> memList = new Hashtable<String, Members>();
      // String id 를 key 값으로 받는 Hashtable
    Hashtable<String, Books> bookList = new Hashtable<String, Books>();
      // String title 을 key 값으로 받는 Hashtable
    ArrayList<RentalInfo> rentList = new ArrayList<RentalInfo>();
    ArrayList<Wish> wishList = new ArrayList<Wish>();
  }
}
```

  - 거스름돈 클래스 : 100원, 500원, 1,000원, 5,000원, 10,000원, 총 수익

```
class Money
{
  int m_man;      // 10,000원 갯수
  int m_5cheon;   // 5,000원 갯수
  int m_cheon;    // 1,000원 갯수
  int m_5baek;    // 500원 갯수
  int m_baek;     // 100원 갯수
  int m_tot;      // 총 수익(누적합)
}
```

- VO 클래스

```
■ 책 VO클래스 : 도서명, 저자명, 출판사, 출판년도, 분류번호(예: 가-15-20),
                카테고리(예: 문학, 에세이...), 도서관 입고일자, 현재 대출 여부

class Books         // 책 VO 클래스
{
  String title;           // 도서명
  String author;          // 저자명
  String publisher;       // 출판사
  String pubYear;         // 출판년도
  String codeNumber;      // 분류번호
  String category;        // 카테고리
  Calendar storedDate;    // 도서관 입고일자
  boolean rental=true;    // 현재 대출 여부 (true : 대출 가능  / false : 대출 중)
}


■ 회원 VO클래스 : id, pw, 이름, 주민등록번호, 대출 중인 권 수, 연체여부

class Members     // 회원 VO 클래스
{
  String id;              // 아이디
  String pw;              // 비밀번호
  String name;            // 이름
  String ssn;             // 주민등록번호
  int rentalBook=0;       // 대출 중인 책 수
  boolean overdue=false;  // 연체 여부 (true : 연체  / false : 연체X)
}


■ 대출 정보 VO클래스 : 대출일자, 반납일자, 대출도서, 대출회원, 연체료
class RentalInfo
{
  Calendar rentalDate;        // 대출한 일자
  Calendar returnDate;        // 반납된 일자
  Book rBook;                 // 대출도서
  Member rMem;                // 대출회원
  int lateFee=0;              // 연체료
}

■ 희망도서 VO클래스 : 도서명, 저자명, 신청일자, 신청상태
```

```
class Wish
{
  String wTitle;        //도서명
  String wAuthor;       //저자명
  Calendar reqDate;     //신청일자
  int request=1;        //신청상태(1:신청완료, 2:입고완료, 3:신청거부)
}
```

- 인터페이스
  - 로그인 / 서비스선택(Admin/User 에 따라 재정의) / 신규 입고 도서 목록 조회

```
Interface LibInterface
{
  public void login();        // 로그인
  public void menuDisp();     // 메뉴 화면
  public void menuSelect();   // 메뉴 선택
  public void menuRun();      // 메뉴 실행
}
```

- 공통 클래스: 관리자, 유저에서 둘 다 사용할 메소드

```
abstract class LibCommon implements LibInterface
{
  boolean checkBook(String title) {}
        // 도서존재여부 - 도서 추가, 도서 검색, 희망도서 신청
  boolean checkMem(String id) {}
        // 회원 존재 여부 - 대출 현황 검색(이용자ID), 회원 정보 검색, 회원 로그인
  void newBookListPrint() {}                  // 신간도서목록 출력
  int calLateFee(대출정보)                     // 연체료 계산
  int calLateDays(대출정보) {}                 // 연체일수 계산
  void inputMoney() {int, int, int, int, int}
        // 돈 입금(관리자 충전, 이용자 연체료)
}
```

- 관리자 클래스

```
class AdCommon extends LibCommon
{
  login() / menuDisp() / menuSelect() / menuRun(int sel)
}

class AdminSystem
{
  1(도서대출현황), 2(도서관리), 3(회원정보조회및삭제),
```

```
  4(연체료관리), 5(이용자모드), 6(프로그램종료)
}

class AdminSystem extends AdCommom
{
  @Override
  public void login() {}

  @Override
  public void menuDisp() {}

  @Override
  public void menuSelect() {}

  @Override
  void menuRun(int sel)
  {
    1: selectInfoMenu();    // 도서 대출 현황
    2: selectAdminMenu();   // 도서 관리
    3: selectMemCon();      // 회원정보 조회 및 삭제
    4: selectLateFee();     // 연체료 관리
    5: userOn();            // 이용자모드
    6: exit();              // 프로그램 종료
  }

  // 도서 대출 현황.
  void selectInfoMenu()
  {
    1 : selectSearchMenu();   // 대출현황검색(제목/id)
    2 : overdueStatus();      // 연체 대출 현황
    3 : rentalStatus();       // 전체 대출 현황
  }
  void overdueStatus() {}     // 연체 대출 현황
  void rentalStatus() {}      // 전체 대출 현황

  // 도서 대출 검색 유형 선택
  void selectSearchMenu()
  {
    1 : searchBook();         // 도서 제목 검색
    2 : searchId();           // 이용자ID 검색
  }
  void searchBook() {}        // 도서 제목 검색
  void searchId() {}          // 이용자ID 검색

  // 도서관리
  void selectAdminMenu() {}
  {
    1: addBook();       // 도서추가
    2: allBookList();   // 전체도서목록
    3: wishBookCon();   // 희망도서관리
    4: newBookList();   // 신규입고도서목록 (공통)
  }
  void addBook() {}         // 도서추가
  void allBookList() {}     // 전체도서목록
  void wishBookCon() {}     // 희망도서관리
  void newBookList() {}     // 신규입고도서목록

  //회원정보
```

```
  void selectMemCon()
  {
    1: allMemList();    // 전체 회원 조회
    2: searchMem();   // 회원 검색
    3: deleteMem();   // 회원 삭제
  }
  void allMemList() {}      // 전체 회원 조회
  void searchMem() {}     // 회원 검색 조회
  void deleteMem() {}     // 회원 삭제

  //연체료 관리
  void selectLateFee()
  {
    1: totalLateFee();    // 총 연체료 조회
    2: memOfLateFee();    // 회원별 연체료 이력 조회
    3: bookOfLateFee();   // 도서별 연체료 이력 조회
    4: changeInOut();     // 거스름돈 조회 및 입출금
  }
  void totalLateFee() {}
  void memOfLateFee() {}
  void bookOfLateFee() {}
  void changeInOut() {}

  //이용자모드on
  void userOn() {}

  //프로그램 종료
  void exit()
  {
      System.out.println("\t프로그램 종료~!!!");
      System.exit(-1);
  }
}
```

- 유저 클래스

```
class UserCommon extends LibCommon
{
  login() / menuDisp() / menuSelect() / menuRun(int sel)
}

class User
{
  1(검색), 2(대출), 3(반납), 4(희망도서신청),
  5(마이페이지), 7(로그아웃), 8(회원가입)
}

class UserSystem extends UserCommon
{
  // 회원여부
  static void memCheck()
  {
    Y: 로그인 정보 입력
```

```
  N: 회원가입 여부 물어보기
  A: 관리자 로그인
}

// 회원가입
static void join(){}

@Override
public void login(){}
@Override
public void menuSelect() {}
@Override
public void menuDisp() {}

@Override
void menuRun()
{
  1: selectSearchMenu();  // 도서 검색
  2: rentalBook();        // 도서 대출
  3: returnBook();        // 도서 반납
  4: requestBook();       // 희망 도서 신청
  5: selectMypage();      // 마이페이지
  6: newBookList();       // 신규 입고 도서 목록(공통)
  7: logout();            // 로그아웃
}

//도서검색
void selectSearchMenu()
{
  1: void searchTitle();      // 도서명 검색
  2: void searchAuthor();     // 저자명 검색
  3: void searchPublisher();  // 출판사명 검색
  4: void searchCategory();   // 카테고리명 검색
}
void searchTitle() {}       // 도서명 검색
void searchAuthor() {}      // 저자명 검색
void searchPublisher() {}   // 출판사명 검색
void searchCategory() {}    // 카테고리명 검색

//도서대출
void rentalBook() {}

//도서반납
void returnBook() {}

//희망도서신청
void requestBook() {}

//마이페이지(1.내정보, 2.나의대출현황, 3.나의희망도서신청현황)
void selectMypage()
{
  1: myInfo();            // 내 정보
  2: myRentalStatus();    // 나의 대출 현황
  3: myWishBookStatus();  // 나의 희망 도서 신청 현황
}
void myInfo() {}
void myRentalStatus() {}
void myWishBookStatus() {}
```

```
  //신규입고도서
  void newBookList() {}

  //로그아웃
  void logout() {}
}
```

내일 논의해야 하는 것

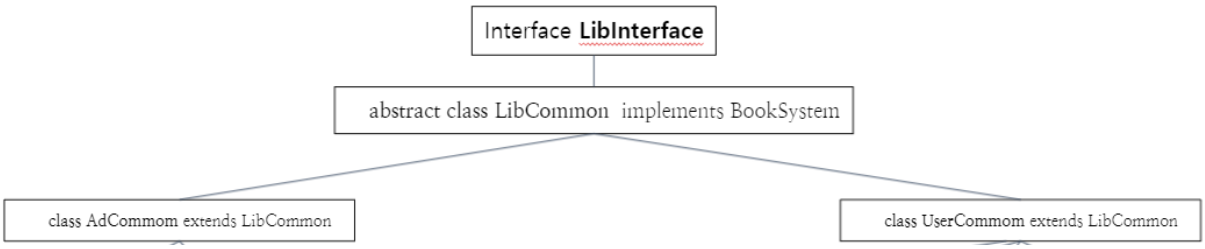
1. main 에서 AdimSystem, UserSystem class에 대한 각각의 인스턴스를 생성할 것인지 여부
2. 공통 메소드 결정
    a. 신규입고도서 메소드(newBookList();) 등
3. LibInterface 인터페이스에서 선언할 추상메소드 결정
4. 코드 업무 분장

1조: 회의록(4일차)

문제되는 부분 논의

1. main 메소드에서 만든 자료구조 호환 문제
    - 자료구조를 static 으로 생성하여,
      각 클래스에서 main 메소드가 포함된 클래스에 접근하는 방식으로 해결
2. 각각 클래스의 메소드 활용 문제
    - 각 클래스의 메소드는 static 으로 생성하는 방식으로 해결

공통메소드 결정



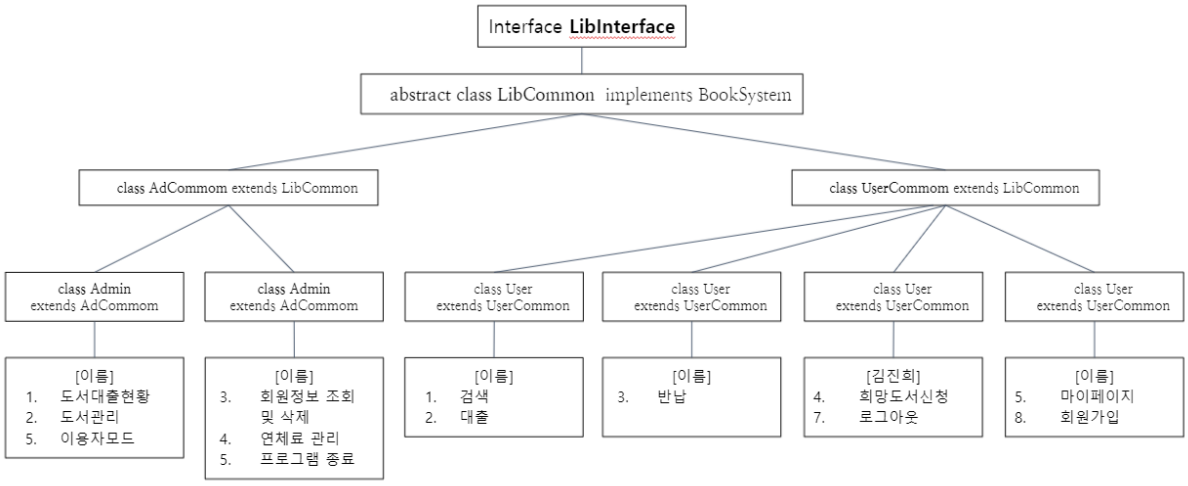

```
abstract class LibCommon implements LibInterface
{
    public boolean checkBook(String title)    // 도서존재여부
    public boolean checkMem(String i)         // 회원 존재 여부
    public void newBookListPrint()            // 신간도서목록 출력
    public int calLateFee(RentalInfo r)       // 연체료 계산
    public int calLateDays(RentalInfo r)      // 연체일수 계산
    public void inputMoney(int man, int 5cheon, int cheon, int 5baek, int baek)
                                  // 돈 넣기 : 관리자 충전 시, 이용자 연체료 입금 시
}
```

공통메소드 구현 소모임 활동

- Lib : 권순모, 정효진

    - 테스트 용도로 사용할 기본 자료구조값 설정
- User : 정가연, 정미화, 최수지
    - String con 을 각 클래스 변수로 선언
    - String nowId 를 User 클래스의 클래스 변수로 선언
    - members와 books 의 사용자 정의 생성자 필요성
    - members와 books 의 key 값에 들어가는 id 와 title 은 제외하기로
    - members와 books 의 Calendar 값에 대해 논의
- 전체 논의
    - Hashtable 전체 출력 방법 찾기

역할 분배



1조: 회의록(5~6일차)

5일차

- 수시로 소통하며 배분된 부분을 각자 코딩

6일차 회의 오후 4시 : 회의 ①

- 주요 사항 공유 및 의논
  - calLateFee() 메소드 오버로딩
  - Exception 처리 관련 사항 논의
  - LibCommon 클래스의 newBookListPrint() 메소드 미완성 사항 확인

- 개인별 진행사항 확인
  - 미화, 효진 : 미완성(70~80% 완성)
  - 나머지 : 완성

- 역할 재배분
  - 미화, 효진 : 맡은 부분 개발
  - 가연: newBookListPrint() 메소드 개발
  - 수지: 완료된 부분 통합
  - 순모: 서포트

- 다음 회의 일정
  - 오후 7시

6일차 회의 오후 8시 30분 : 회의 ②

- 주요 사항 공유 및 의논
  - 남은 업무
    1. 기존 결과물 통합, 검토 및 수정
    2. 보고/발표용 자료 준비 및 역할 분담
    3. 추가적인 내용 개발
       a. try~catch
       b. 자료구조 저장

- 개인별 진행상황 확인
  - 미화, 효진 : 미완성(80~95% 완성)
  - 가연: newBookListPrint() 메소드 완료
  - 수지, 진희: UserSystem 통합 미테스트 완료
  - 순모: 서포트 진행 특이사항 없음

- 역할 재배분
  - 미화, 효진 : 맡은 부분 개발
  - 순모, 진희 : 테스트 데이터 셋팅
  - 가연, 수지 : 서포트(myRentalStatus() 메소드)

- 다음 회의 일정
  - 12시(자정)

6일차 자정 : 회의 ③

- 주요 사항 공유 및 의논
  - 테스트 데이터 관련 질의응답
    - 대출했다가 반납한 이력의 데이터 추가 요청

        - 해당 데이터에 대한 사항 조율
    - calLateFee() 메소드 관련 사항 전달
        - calLateFee(String id) 메소드 정정
        - calLateFee(String id, boolean f) 메소드 추가 공유(오버로딩)

- 개인별 진행상황 확인
    - 미화 : 미완성(95% 완성)
    - 효진 : 완성
    - 순모, 진희 : 추가 요청 받은 테스트 데이터 셋팅 진행 중
    - 가연, 수지 : 서포트 완료, 완료된 작업물 통합

- 역할 재배분
    - 미화 : 맡은 부분 개발
    - 순모 : 테스트 데이터 셋팅
    - 그 외: 내일 분배 예정
        - 남은 업무
            1. (★) 맡은 부분(미화), 테스트 데이터 셋팅(순모)
            2. 기존 결과물 통합, 검토 및 수정
            3. 보고/발표용 자료 준비 및 역할 분담
                a. (★) 발표자
                b. (★) 발표자료 준비
                c. (★) 보고자료 준비(각 팀원들 후기 포함)
                d. 회의록 정리
                e. 기타 작업물 정리(기획안, 예상구현화면, 플로우 차트, 공유문서 등...)
            3. 추가적인 내용 개발
                a. try~catch
                b. 자료구조 저장

1조: 회의록(7일차)

팀장회의 내용 전달

- 각 팀별 진행상황 확인
- 발표 및 보고자료 질의
    - 발표: 시간 및 형태는 자유
    - 보고서: 후기 필수, 그 외 형태 및 양식은 자유

회의 ① : 9시

- 의논사항
    - 테스트 자료 관련 질의
    - 본인이 짠 코드에 대한 숙지할 것

- 최종 업무 예상 흐름도
    1. 개발 완료
        - 미화, 순모 미완성 부분 마무리
    2. 최종 발표 및 보고 자료 준비
        - 보고자료(순모, 효진), 발표자료(가연, 진희) 조가 함께 기본 보고 자료 준비
        - 기본 논의사항을 바탕으로, 각자 보고서 최종 준비
    3. 발표 준비
        - 대표 발표자(미화, 수지)
        - 발표자가 아니더라도, 질의응답에 답변할 수 있도록 본인 코드 숙지할 것

- 발표 및 보고 역할 분담
    - PPT 발표: 미화
    - 시연 발표: 수지

    - 보고자료: 순모, 효진
    - 발표자료: 가연, 진희

- 다음 회의까지 역할 분담
    - 미화, 순모: 미완성 부분 마무리
    - 수지: 취합 및 발표 준비
    - 가연, 효진, 진희: 보고자료 준비

- 다음 회의 일정
    - 11시

회의 ② : 11시

- 의논 및 전달 사항
    - 메인메뉴 → 하위메뉴 → 작업1 실행 후, 바로 메인메뉴로 돌아가지 않고 하위메뉴를 반복하는 형태로 수정작업 진행 필요 ⇒ 수지가 해결 중

- 개인별 진행사항 확인
    - 미화: 미완성 부분 완료
    - 순모: 테스트 자료 완료 했으나 수정사항 발생, 수정 중
    - 수지: 프로그램 통합 및 테스트, 세부 사항 수정, 발표 준비 진행 중
    - 가연, 효진, 진희: 보고자료 준비 중(플로우 차트 정리)
        - 관리자, 대출 이력 부분 : 코딩 담당자(미화, 순모)가 체크할 필요 있음

- 역할 분담
    - 순모: 테스트 자료 수정, 플로우 차트 확인
    - 수지: 프로그램 통합 및 테스트, 세부 사항 수정, 발표 준비 진행 중

    - 미화: 회의록 최종 정리
    - 가연, 효진, 진희: 플로우 차트 보완 ⇒ 보고서 1차 작성

- 진행 예정
    - 프로그램 통합 및 테스트 완료 ⇒ 예외처리, 자료구조 저장
    - 플로우 차트 보완 완료 ⇒ 보고자료 1차 작성 ⇒ 보고자료, PPT 작업

- 다음 회의 일정
    - 2시

회의 ③ : 오후 2시

- 의논 및 전달 사항
    - 특이사항 없음

- 개인별 진행사항 확인
    - 미화: 회의록 정리 완료
    - 수지
        - 프로그램 테스트 진행 중, 세부사항 수정
        - 도서 추가 메소드 오류 해결 완료
        - 대출 메소드 오류 해결 중
        - 오류 해결 서포트 필요(⇒ 미화가 하기로 결정)
    - 순모: 보고서 1차 작성 서포트 ⇒ 테스트 데이터 오류 해결 중
    - 가연, 효진, 진희: 플로우 차트 완성, 보고서 개요 작성 완료

- 역할 분담
    - 수지/미화: 프로그램 테스트 및 오류 해결

    - 순모: 테스트 자료 수정, 플로우 차트 확인
    - 가연, 효진, 진희: 플로우 차트 보완 ⇒ 보고서 1차 작성

- 다음 회의 일정
    - 4시

회의 ④ : 오후 5시

- 의논 및 논의 사항
    - 프로그램 통합하였으니 변경사항 있을 경우 바로 공유할 수 있도록

- 개인별 진행사항
    - 오류 해결팀 : 수지, 미화, 순모
    - 보고서팀
        - 1차 작성 가능한 부분까지 완료
        - 주요 클래스 및 변수 작성 진행 중
            1. 가연 : 관리자, LibCommon
            2. 효진 : 이용자
            3. 진희 : VO클래스
        - 보고서 작성 완료 후 PPT 진행 예정

- 다음 회의 일정
    - 9시

회의 ⑤ : 오후 9시

- 의논 및 논의 사항

    - 발표 형식 관련
        - 내용: 분석, 설계 과정, 시연
        - 완전히 발표와 시연을 분리하지 않고, 시연을 발표 중간에 배정
        - 모든 메소드를 담기는 어려울 것 같으니, 중요하고 어려운 기능 위주로 설명
        - 기능 설명 시에도 "시연에서 추가 설명 드리겠습니다."라고 언급하기
        - 화면 공유는 한 명만 하기
        - 소스코드 삽입 여부: PPT(X), 보고서(O)

- 개인별 진행사항
    - 특이사항 없음

- 역할 분담
    - 미화: PPT 발표 준비
    - 수지: 시연 발표 준비
    - 순모: 에러 처리 후, PPT 준비 예정
    - 가연: PPT 준비 후, 에러 처리 예정
    - 효진, 진희: PPT 준비

- 다음 회의 일정
    - 11시

회의 ⑥ : 오후 11시 30분

- 의논 및 논의 사항
    - PPT 발표 형식 관련 논의

- 개인별 진행사항

    - 특이사항 없음

- 역할 분담
    - 미화: PPT 발표 준비
    - 수지: 시연 발표 준비
    - 가연, 효진, 진희: PPT 준비
    - 순모: 에러 처리 후

- 다음 회의 일정
    - 1시

회의 ⑦ : 오전 1시

- 의논 및 논의사항
    - PPT 발표 예시, 시연 발표
    - PPT 방향성 검토
    - 보고서 준비

- 역할분담
    - 미화, 수지: 발표 준비
    - 가연, 효진, 진희, 순모: 보고서 작성