

# SE 2020 봄학기 팀프로젝트

## 헌책 장터 시스템 (Used-book Marketplace)

**팀규모:** 개인과제이므로 각자 프로젝트를 독립적으로 진행해야 함 (1인 1팀)

**제출마감:** 6월 25일 목요일 오후 10시

**제출방법:** eclass 과제 및 평가 탭의 과제 "Project"에 업로드

**소프트웨어 구현 언어:** Java 혹은 C++ (C++를 선택한 경우 CPP Unit 등 JUnit과 유사한 단위테스트 프레임워크를 사용해야 함)

**제출물:** "성명\_학번.zip" 형식의 화일명을 갖는 zip 화일 1개를 제출하며, 해당 zip 화일은 다음을 포함해야 함

- README.txt 화일: 제출된 모든 산출물에 대한 목록 및 요약.
- 모든 소스코드 화일, 실행 화일 (Java의 경우 class 화일, C++의 경우 exe화일), 테스트 코드, 데이터 등
- 프로젝트 문서 (문서에 포함되어야 하는 내용은 하기 설명 참고)
- 비디오 클립1: 제출된 프로그램을 실행하며 주요 기능들이 잘 동작하는 것을 시연하는 비디오. 프로그램을 구동하는 장면부터 각 기능들이 잘 동작한다는 것을 효과적으로 보여주는 시연을 할 것. JUnit이 수행된 화면도 포함할 것. (20분 내외 정도의 분량)
- 비디오 클립2: 프로젝트 문서의 내용을 설명하는 비디오. 요구 명세 및 분석, 설계 및 구현, 테스트 산출물 등에 대한 설명을 포함하며 특히 주요한 설계 결정들에 대해 언급하고 그에 대한 근거를 제시할 것. 또한, 테스트 케이스를 만든 목적 등에 대한 설명이 포함되어야 함. 만일 프로그램이 완성되지 못하였다면 어떤 부분들이 부족한지 설명을 할 것 (20분 내외 정도의 분량)

(주의) 각 비디오 클립의 시작 부분에는 신분증 제시와 함께 본인의 얼굴을 보여주고 이름과 학번을 말하는 영상과 음성이 포함되어 있어야 함. 이후로는 화면과 음성만 있어도 관계 없음.

### Project Description:

본 과목의 수강을 통해 그간 배운 OOAD 기법을 사용해서 소프트웨어 개발을 하고 단위 테스트를 수행한다. 본 프로젝트는 사용자들이 가진 책을 다른 사용자들에게 팔 수 있는 헌책 장터 시스템을 개발하는 것이 목표이다. 사용자들은 시스템에 등록된 책을 찾아보고 다음과 같은 기능을 이용할 수 있다. 본 시스템은 실제 거래를 지원하지는 않으며, 거래를 원하는 구매자와 판매자가 오프라인에서 직접 만나서 거래하는 것으로 가정한다. 책을 판매한 사용자는 거래가 끝난 후에 삭제 기능을 이용하여 해당 책을 없앤다. (하기에서 충분히 설명되지 않은 조건 및 세부 명세 등에 대해서는 각자 적절한 가정을 세우고 프로젝트 문서에서 해당 가정을 명시할 것)

### 일반 사용자와 관리자를 위한 기능

- 등록된 id와 암호를 통해 시스템에 로그인함.
- 제목, ISBN 번호, 저자, 판매자 id별로 시스템에 등록된 책을 검색할 수 있음. 검색 결과에는 책의 제목, 출판사, 저자 정보, 출판년도를 비롯하여, 등록된 가격과 상태(Excellent, Good, Fair)가

포함 되어야함. 이때 동일한 제목을 가진 책이 여러 권 있을 수 있다고 가정함.

### 일반 사용자를 위한 기능

- 각 사용자는 id, 암호, 이름, 전화번호, 이메일을 제공하여 시스템에 등록함.
- 검색된 책을 구입 (구입 의도를 나타내는 이메일을 구매자와 소유자에게 발송. 이메일은 외부의 서비스를 사용하는 것으로 가정하고 실제 구현은 화면에 구매자와 소유자의 이메일을 표시하고 그 메일 주소로 메일이 발송되었다고만 출력)
- 판매를 위해 자신의 책을 등록. 이때 사용자는 책의 제목, 출판사, 저자 정보, 출판년도, 가격 및 상태를 입력함. 이때 책의 제목을 제외한 나머지 책의 정보는 생략될 수 있음.
- 판매를 위해 자신이 등록한 책들을 리스트하고 그들 중 하나를 선택하여 해당 책의 정보를 고치거나 책을 삭제할 수 있음

### 관리자를 위한 기능

- 관리자의 id는 admin이고, 암호는 nayana임
- 검색된 특정한 책을 삭제. 이 기능을 이용해서 분쟁 등이 발생할 경우 문제의 책을 시스템에서 삭제할 수 있으며, 해당 책의 판매자 계정에서도 동일하게 없어져야 함
- 등록된 사용자를 리스트하여 전체 사용자의 정보를 보여줌. 부가적으로 사용자를 선택하여 activated 혹은 deactivated 상태로 전환할 수 있으며, deactivated 상태의 사용자는 시스템에 로그인할 수 없음. 등록된 사용자의 디폴트(default) 상태는 activated 임. 또한 deactivated 상태의 사용자를 시스템에서 삭제할 수 있으며, 이 경우 해당 사용자의 계정 정보와 함께 등록한 책들도 함께 삭제함.

### 유저 인터페이스(User Interface, 이하 UI) 관련 사항

- GUI(Graphical User Interface) 혹은 콘솔 (키보드와 텍스트 형식) 인터페이스를 통해 동작함
- GUI 혹은 콘솔 방식을 취하는 지에 따라 평가에서 가산/감점 하지 않음 (Tip: GUI 프로그래밍에 사전 지식과 경험이 없다면 콘솔 방식으로 구현하는 것이 간편할 것임)
- 평가에서 중요한 것은 UI와 응용 로직(Logic)을 분리하여 추후 UI가 바뀌더라도 로직 이하의 계층은 변경없이 재사용할 수 있도록 설계 및 구현하는 것임. 이와 같은 설계 내용이 프로젝트 문서에도 명확히 나타나야 함

### 프로젝트 문서에 포함되어야 할 내용

#### [요구 정의 및 분석 산출물]

- 유스케이스 다이어그램
- 유스케이스 명세 (사용자용 2개, 관리자용 2개 선정. 하기 SSD를 작성할 수 있을 정도로 자세한 수준으로 작성할 것)
- 도메인 모델
- SSD (System Sequence Diagram): 작성된 유스케이스 명세를 기반으로 SSD 기술할 것
- Operation Contract 최소 2개

[설계 산출물]

- Design Class Diagram
- Sequence Diagram: SSD에 나타난 대표적인 System operation 3개 이상을 선정하여 작성

[구현 및 테스트 산출물]

- 소스코드
- JUnit (혹은 CppUnit)을 이용한 테스트 케이스 및 테스트 결과