

# 소프트웨어공학 프로젝트

## < 과제 1 : Requirement Capturing – 공유 자전거 대여 시스템 >

### 1. 개요

이 프로젝트에서는 사용자가 자전거를 대여하고 반납할 수 있는 공유 자전거 대여 시스템을 개발한다.

### 2. 기능

#### - 회원 가입 기능

시스템을 이용하려면 사용자는 회원 가입을 해야 한다. 회원의 필수 입력 정보는 ID, 비밀번호, 전화번호, 결제 수단, 선호 자전거 유형(일반/전기) 등이다. 관리자는 주어진 ID(admin)와 비밀번호(admin)로 로그인한다.

#### - 회원 탈퇴 기능

회원은 언제든지 탈퇴할 수 있으며, 탈퇴 시 모든 이용 권한과 데이터가 삭제된다.

#### - 로그인/로그아웃 기능

관리자와 회원은 ID와 비밀번호로 로그인하며, 로그아웃 시 시스템 접속이 종료된다.

#### - 대여소 등록/조회/삭제 기능

관리자는 대여소 정보를 등록, 조회 및 삭제할 수 있다. 등록 시 입력 정보는 대여소 이름, 위치(도시, 주소), 자전거 보관 가능 수량, 운영 시간 등이다. 등록된 대여소 리스트를 조회할 수 있다. 이때, 대여소 리스트 조회 화면에서 원하는 대여소 항목을 선택하면 등록시 입력한 상세내용을 볼 수 있다. 또한, 대여소 리스트 조회 화면에서 특정 대여소 항목을 선택해서 삭제할 수 있다.

#### - 자전거 등록/조회/삭제 기능

관리자는 자전거 정보를 등록, 조회 및 삭제할 수 있다. 등록 시 입력 정보는 자전거 ID, 자전거 제품명, 유형(일반/전기), 소속 대여소, 상태(사용 가능/수리 중) 등이다. 등록된 자전거 리스트를 조회할 수 있으며 원하는 자전거 항목을 선택해서 상세내용을 볼 수 있다. 또한, 등록된 자전거 리스트 조회 화면에서 특정 자전거 항목을 삭제할 수 있다.

#### - 대여소 검색 기능

회원은 조건에 맞는 대여소를 검색하고 자전거를 대여할 수 있다. 검색을 위해 대여소 이름을 입력한다. 검색 결과로 조건에 맞는 대여소 리스트가 출력된다.

**- 상세정보 조회 및 자전거 즉시대여/예약대기 기능**

회원은 검색된 대여소 리스트 화면에서 특정 대여소를 선택하면 상세정보화면을 볼 수 있다. 이 화면에서는 대여소 이름, 대여소 위치, 사용 가능 자전거 목록 등이 출력된다. 이때 현재 자전거가 남아 있는 경우 즉시 대여할 수 있고, 자전거가 없는 경우 예약대기를 신청할 수 있다. 두 경우 모두 문자 알리를 통해 해당 내용을 전송받는다.

**- 자전거 대여 정보 조회**

회원이 현재 대여 중인 자전거를 조회하면 해당 리스트가 출력되고 각 항목에는 대여소 이름, 대여소 위치, 자전거 ID, 자전거 제품명, 자전거 유형을 보여준다.

**- 자전거 예약대기 정보 조회/취소**

회원이 자전거 예약대기 정보(대여소 이름, 대여소 위치, 자전거 ID, 자전거 제품명, 자전거 유형)를 조회하면 해당 리스트가 출력된다. 또한, 각 예약대기에 대해 취소할 수 있다.

**- 자전거 반납 및 식당 예약 서비스 연계 기능**

회원은 자전거 대여 정보 조회 화면에서 특정 자전거를 지정된 대여소에 반납할 수 있다. 자전거 반납 후에는 원하는 경우 사용자 위치 정보를 기반으로 근처 식당을 추천받아서 예약할 수 있는 외부 서비스와 연결된다. 해당 자전거에 대기 예약한 회원이 있는 경우 대기 1순위 회원에게 예약되었다는 이메일을 보낸다.

**- 결제 및 요금 조회 기능**

회원은 반납 시 사용 시간에 따라 요금이 자동 결제된다. 요금 조회 화면에서는 대여 시간 및 요금을 볼 수 있다.

**- 이용 내역 조회/삭제 기능**

회원은 과거 대여 기록을 조회할 수 있다. 디폴트는 날짜별로 출력되며 원하는 경우 대여소별로 정렬해서 조회할 수 있다. 과거 기록 중 특정 항목을 선택해서 삭제할 수 있다.

**- 통계 기능**

관리자는 자전거 대여 정보를 반납 시간 기준 최근순으로 조회할 수 있다. 원하면 지역별 기준으로 정렬해서 조회할 수 있다. 또한, 최근 1주일, 1개월, 1년 단위로 대여 금액 및 대여 횟수를 조회할 수 있다.

※ <<include>>와 use case 간의 generalization은 사용하지 않는다.

### 3. 과제 제출물

#### (1) 보고서 파일 3개

이번 과제에서는 Requirement Capturing 단계를 수행함. 교재에 있는 Case Study A2 참조하여 다음 documents를 각각 별개의 파일로 작성함.

([https://books.google.co.kr/books?id=IMovEAAAQBAJ&pg=PA169&hl=ko&source=gbs\\_toc\\_r&cad=3#v=onepage&q&f=false](https://books.google.co.kr/books?id=IMovEAAAQBAJ&pg=PA169&hl=ko&source=gbs_toc_r&cad=3#v=onepage&q&f=false))

1) requirement list (functional requirement만 작성함)

2) use case diagrams (UML tool 형식을 pdf 파일로 변환해서 제출함)

3) use case descriptions (step by step breakdown)

※ 위의 1)번 파일 맨 첫 부분에 채점내용 확인용 팀 코드 추가해야 함.

-> 첫 숫자는 0 사용금지 (예: 01234, 00987 등)

-> 동일한 숫자 3개 이상 사용금지 (예: 11123, 19299 등)

-> 연속적으로 증가 혹은 감소하는 숫자 사용금지(예: 12345, 98765 등)

그 아래에는 팀 내 역할 분담 및 각 팀원이 실제 수행한 내용을 상세히 명시하기 바람.

-> 협업용 메신저 사용 권장(e.g. <http://slack.com>)

※ UML tool 정보

- [https://en.wikipedia.org/wiki/List\\_of\\_Unified\\_Modeling\\_Language\\_tools](https://en.wikipedia.org/wiki/List_of_Unified_Modeling_Language_tools)

-> Tool 선정 요건: UML 2.x 지원, C++ code generation 제공

#### (2) Git commit history 파일 1개

\$ git log --pretty="%an" %cd %s %h" > log\_output.txt

(<https://git-scm.com/book/en/v2/Git-Basics-Viewing-the-Commit-History>에서

git log 명령어의 --pretty 옵션에서 사용되는 서식문자를 참고하기 바람)

※ 위의 파일 첫 부분에 GitHub 주소를 명시하기 바람.

#### ▶ 압축 파일 이름 형식

✓ 보고서 파일 3개와 git commit history 파일 1개를 압축하고 압축 파일 이름만 아래 형식에 맞게 작성해서 제출하기 바람.

- 팀번호\_팀원이름 리스트.zip (예: 23\_홍길동-김홍익-김대학.zip)

### 4. 제출 마감 시간 및 방법

✓ 제출 마감 시간 : 5월 9일 (금요일) 오후 2시

✓ 방법 : 클래스룸 '과제1'에 업로드 (팀당 1명만 제출함)

## 5. 유의 사항

### (1) 감점 사항

- ✓ 제출 기한 이후에는 매일 25점씩 감점함
- ✓ 부정행위 발견 시 관련 학생 모두 F 학점 처리함

### (2) 질문은 클래스룸 QnA 게시판을 이용하기 바람.

## 6. 채점 기준표

### 1) 요구사항/기능을 모두 반영했는가 ? (60점)

	체크 항목	감점
1	requirement list에서 빠진 기능	개당 -3점
2	작성한 functional requirement list 항목 중에서 use case로 미표현, description 작성 오류 혹은 미작성 (1번과 중복 감점하지 않음.)	개당 -5점
3	use case description을 step by step으로 표현하지 않음	개당 -3점 최대 -12점

### 2) Use case diagram의 개념을 적절히 적용하고 있는가? (15점)

	체크 항목	감점
1	use case description에서 시스템 내부 기능을 기술한 경우 (예: 내부A 모듈에 B 처리를 요청함, DB 연산을 요청함 등)	1개 이상 -9점 (단일점수 감점)
2	불필요한 actor 혹은 시스템 내부 actor 항목 추가	1개 이상 -3점 (단일점수 감점)
3	diagram notation 오류(<<extend>> 화살표 방향 등)	1개 이상 -3점 (단일점수 감점)

### 3) Tool 사용 방법을 잘 알고 있는가? (5점)

	체크 항목	감점
1	use case diagram을 수기로 작성한 경우	-5점

### 4) 팀원 간에 역할 분담 및 수행 정도가 적절한가? (20점)

	체크 항목	감점
1	log_output.txt를 참고해서 commit 한번도 안한 팀원이 있는 경우	-10점
2	use case 단위로 분배하지 않았거나 고르게 분배하지 않은 경우	-10점