

567501

https://github.com/ssangyeon/software_2

Requirement list

- 1.회원가입: 비회원은 id, 비밀번호, 전화번호를 입력하여 회원가입한다.
- 2.로그인: 관리자와 회원은 id와 비밀번호를 입력하여 로그인한다.
- 3.로그아웃: 관리자와 회원은 로그아웃 할수 있고, 시스템 접속이 종료된다.
- 4.자전거등록: 회원은 특정 자전거를 대여할 수 있다.
- 5.자전거 대여 정보 조회: 회원의 현재 대여중인 자전거 리스트가 출력되고, 각 항목에는 자전거 ID, 자전거제품명을 보여준다.

회원 가입 기능:

1. 비회원은 id, 비밀번호, 전화번호를 입력한다.
2. 회원가입 성공 메시지를 출력하고 로그인 페이지로 이동한다.

로그인 기능:

1. 로그인 페이지 출력
2. 멤버(회원, 관리자)가 ID, 비밀번호 입력
3. 로그인 성공 메시지 출력 후 메인화면 이동

로그아웃:

1. 멤버(회원, 관리자)가 로그아웃 버튼 클릭
2. 로그아웃 완료 메시지 출력 후 메인 화면 이동

자전거 등록:

1. 자전거 정보 등록 화면 출력
2. 관리자가 자전거 정보(자전거 id, 자전거 제품명) 입력 후 등록 버튼 클릭
3. “자전거가 등록되었습니다” 메시지 출력

자전거 대여 기능:

1. 자전거 대여화면 출력
2. 회원이 특정 자전거대여 선택
3. “자전거 대여 완료” 메시지 출력

자전거 대여 정보 조회:

1. 현재 회원이 대여 중인 자전거 리스트 출력. 각 항목에는 자전거 ID, 자전거 제품명을 보여준다.

회원가입.

Actor Action	System Response
1.비회원은 id, 비밀번호, 전화번호를 입력한다.	2. 회원가입 성공 메시지를 출력한다.

로그인.

Actor Action	System Response
2.멤버(회원,관리자)가 ID,비밀번호 입력한다.	1.로그인 페이지 출력한다.
	3.로그인 성공 메시지를 출력한다.

로그아웃.

Actor Action	System Response
1. 멤버(회원, 관리자)가 로그아웃 버튼을 클릭한다.	2. 로그아웃 완료 메시지를 출력한다.

자전거 등록.

Actor Action	System Response
	1. 자전거 정보 등록 화면을 출력한다.
2. 관리자가 자전거 정보(자전거 id, 자전거 제품명) 입력 후 등록 버튼을 클릭한다.	3. “자전거가 등록되었습니다” 메시지를 출력한다.

자전거 대여.

Actor Action	System Response
	1. 자전거 대여화면을 출력한다.
2. 회원이 특정 자전거대여를 선택한다.	3. “자전거 대여 완료” 메시지를 출력한다.

자전거 대여정보 조회.

Actor Action	System Response
	1. 현재 회원이 대여 중인 자전거 리스트 출력. 각 항목에는 자전거 ID, 자전거 제품명을 보여준다.

회원가입..

:Guest → :JoinUI: 회원가입 요청 (id, pwd, phone)

:JoinUI → :JoinController: 입력 정보 전달

:JoinController → :Member: 회원 객체 생성

:JoinController → :MemberDB: 회원 정보 저장 요청

:JoinController → :JoinUI: 회원가입 성공 메시지 전달

<<boundary>>JoinUI

<<control>>JoinController

<<entity>>Member, MemberDB

Member

id: String

password: String

phone: String

paymentMethod: String

preferredBikeType: String

JoinController— "1" ↔ "0..1" MemberDB

MemberDB— "1" ↔ "0..*" Member

로그인..

:Member → :LoginUI: 로그인 시도 (id, pwd)

:LoginUI → :LoginController: 로그인 요청 전달

:LoginController → :MemberDB: 회원 조회 요청

:MemberDB → :LoginController: 조회 결과 반환

:LoginController → :LoginUI: 로그인 성공/실패 메시지

<<boundary>>LoginUI

<<control>>LoginController

<<entity>>Member, MemberDB

LoginControllerMemberDB(association: "1" : "0..1")

MemberDBMember(association: "1" : "0..*")

로그아웃..

:Member → :MainUI: 로그아웃 버튼 클릭

:MainUI → :LoginController: 세션 종료 요청

:LoginController → :MainUI: 로그아웃 완료 메시지

LoginController— "1" ↔ "1" MainUI

자전거 등록..

:Admin → :BikeRegisterUI: 자전거 등록 화면 요청

:BikeRegisterUI → :BikeRegisterController: 등록 요청 (bikeId, model)

:BikeRegisterController → :Bike: 자전거 객체 생성

:BikeRegisterController → :BikeDB: 자전거 정보 저장

:BikeRegisterController → :BikeRegisterUI: 등록 성공 메시지

<<boundary>>BikeRegisterUI

<<control>>BikeRegisterController

<<entity>>Bike, BikeDB

Bike

bikeId: String

model: String

type: String (일반/전기)

status: String (사용 가능/수리 중)

station: String

BikeRegisterController- "1" ↔ "1" BikeDB

BikeDB- "1" ↔ "0..*" Bike

자전거 대여..

:Member → :RentalUI: 자전거 대여 요청 (bikeId)

:RentalUI → :RentalController: 대여 처리 요청

:RentalController → :BikeDB: 자전거 사용 가능 여부 확인

:RentalController → :RentalInfo: 대여 정보 생성

:RentalController → :Bike: 상태 변경 (대여 중)

:RentalController → :RentalUI: 대여 완료 메시지 반환

<<boundary>>RentalUI

<<control>>RentalController

<<entity>>Bike, RentalInfo, BikeDB

RentalInfo

rentalId: String

member: Member

bike: Bike

rentalTime: DateTime

Member- "1" ↔ "0..*" RentalInfo

RentalInfo- "1" ↔ "1" Bike

RentalControllerBikeDB, RentalInfo

자전거 대여 정보 조회..

:Member → :RentalUI: 대여 정보 조회 요청

:RentalUI → :RentalController: 현재 회원 ID 전달

:RentalController → :RentalInfo: 대여 목록 조회 (by memberId)

:RentalController → :RentalUI: 자전거 ID/모델 리스트 전달

<<boundary>>RentalUI

<<control>>RentalController

<<entity>>RentalInfo, Bike, Member

RentalInfo- "1" ↔ "1" Bike

RentalInfo- "1" ↔ "1" Member

RentalControllerRentalInfo