**시나리오**

전제 조건 : 비회원으로 사용하는 것에 대해서 구현함

1. 사용자가 자전거 위치를 검색한다.
2. 사용자가 자전거 Rental를 신청한다.
3. 사용자가 자전거의QR 코드를 찍어 Rental할 자전거를 등록한다.
4. 자전거가 사용 가능 상태일 때 자전거 잠김이 해제되고 Rental가 시작된다.
5. 카톡으로 자전거 Rental 시작 메시지가 발송된다. (자전거 ID, 시작 시간)
6. 사용자가 사용 후 자전거를 반납한다.
7. 자전거가 잠김 상태가 되고 위치와 배터리 잔량이 저장된다.
8. 카톡으로 자전거 반납 완료 메시지가 발송된다. (자전거 ID, 반납 시간, 사용 요금)
9. 사용자는 사용 요금을 결제한다.
10. 카톡으로 결제 완료 메시지가 발송된다.

비기능적 요구사항

1. 트랜잭션
   * 자전거가 사용 가능 상태 일 때만 잠김이 해제되고 Rental가 시작된다. (sync 호출)
2. 장애격리
   * 결제 서비스가 수행되지 않더라도 365일 24시간 자전거 Rental/반납이 가능해야 한다. (Async, event-driven, Eventual consistency)
   * Rental시스템이 과중되면 Rental를 잠시 동안 받지 않고 Rental를 잠시 후에 하도록 유도한다 Circuit breaker, fallback
3. 성능
   * 고객이 자전거 위치를 Rental시스템 (프론트엔드)에서 확인할 수 있어야 한다 CQRS
   * Rental 시작과 반납 시 카톡으로 알림을 줄 수 있어야 한다 Event driven

- service : Rental 서비스, Bike 관리 서비스, 결재 서비스, Message 서비스

**# 시나리오2 (류주현 수정)**

- [Core Domain] Rental 서비스, Bike 관리 서비스, [Supporting Domain] 회원관리 [General Domain] 결제

- 전제 조건 : 회원/비회원 모두 사용 가능, 결제는 반납 후 진행

- 체크 포인 트 중 서포팅 서비스를 제거 하여도 기존 서비스에 영향이 없도록 설계가 되어야함. 그래서 Rental 시 회원관리 서비스와는 Sync 호출 요소 없이 사용 실적 저장하는 것으로 시나리오를 수정해 봄

1. 회원 가입을 한다.
2. 사용자가 자전거 위치를 검색한다.
3. 사용자가 자전거 Rental을 신청한다.
4. 사용자가 자전거의QR 코드를 찍어 Rental할 자전거를 등록한다.
5. 자전거가 사용 가능 상태일 때 자전거 잠김이 해제되고 자전거 Rental이 시작된다.
6. 사용자가 사용 후 자전거를 반납한다.
7. 자전거가 반납되면 잠김 상태가 되고 위치와 배터리 잔량이 저장된다.
8. 사용자는 사용 요금을 결제한다.
9. 사용자가 회원인 경우 회워 정보에 사용 이력이 저장된다. (phone number로 회원 여부 판단, 사용 일자, 자전거 시작 위치, 반납 위치, 사용 시간 저장)
10. 자전거 Rental을 종료 한다.

비기능적 요구사항

1. 트랜잭션
   * 자전거가 사용 가능 상태 일 때만 잠김이 해제되고 Rental이 시작된다. (sync 호출)
2. 장애격리
   * 회원 서비스가 수행되지 않더라도 365일 24시간 자전거 Rental/반납이 가능해야 한다.
   * 결제 서비스가 수행되지 않더라도 365일 24시간 자전거 Rental/반납이 가능해야 한다. (Async, event-driven, Eventual consistency)
   * Rental시스템이 과중 되면 Rental를 잠시 동안 받지 않고 Rental를 잠시 후에 하도록 유도한다 Circuit breaker, fallback
3. 성능
   * 고객이 자전거 위치를 Rental시스템 (프론트엔드)에서 확인할 수 있어야 한다 CQRS