Software Requirements Specification

1 소개(Introduction)

1.1 목적(Purpose)

다수의 킥보드 공유서비스의 이용을 한 눈에 보고 가깝고 더 경제적인 가격의 킥보드 서비스의 이용을 돕는 웹 서비스이다.

1.2 문서 규칙(Document Convention)

긴 용어의 경우 ‘이하’ 표시를 통해 언급하고 이후엔 축약된 용어로 나타낸다.

가급적 같은 대상을 지칭하는 단어는 하나로 통일하여 쓴다.

1.3 독자 대상과 읽는 방법(Intend Audience and Reading Suggestion)

본 SRS는 프로젝트에 참여하는 기획자와 개발자들을 대상으로 한다.

1.4 프로젝트 범위(Project Scope)

유저 클라이언트와 서버에 대해 다룬다. 클라이언트는 사용자의 입력과 시각적 데이터를 제공하기 위한 목적을 지닌다.

서버는 사용자가 필요로 하는 정보들을 통합하여 제공하는 목적을 지닌다.

1.5 참조(Reference)

2 전체 설명(Overall Description)

2.1 제품조망(Product Perspective)

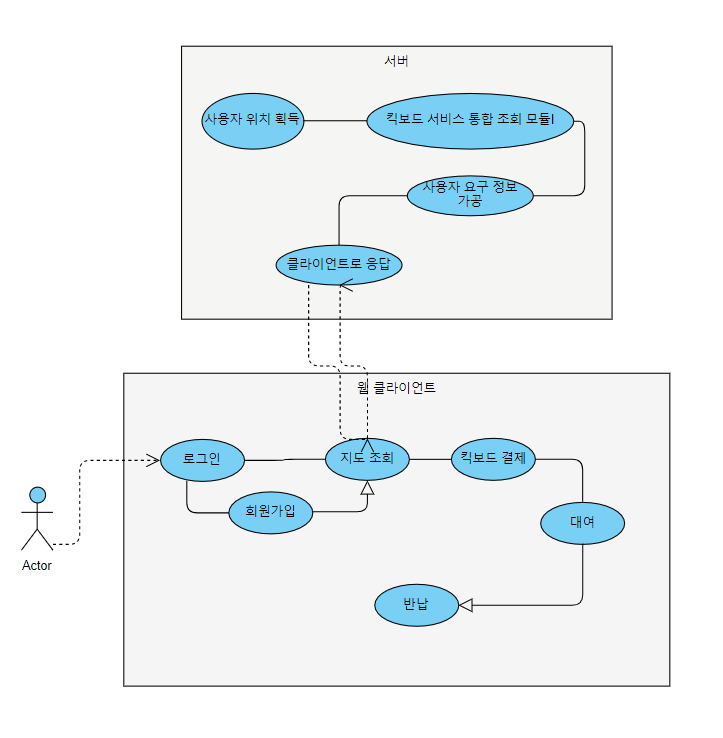
본 제품은 새로운 프로젝트로 출시되며

클라이언트와 서버간의 REST API 통신을 통해 연결된다.

2.2 제품 기능(Product Feature)

제품의 주요 기능은 킥보드 상태와 가격, 사용자 위치기반 킥보드 대여가능 위치 정보를 제공합니다.

제품의 플로우는 아래 다이어그램과 같이 구성됩니다.



2.3 사용자 계층과 특징(User Classes and Characteristic)

이 제품은 킥보드 공유 사업이 활발하게 진행되고 있는 강남권과 판교 인근 주민 및 회사 근로자가 사용자 계층이 된다.

회사 근로자의 경우, 출퇴근시간과 점심시간이 주 사용시간대가 될 것으로 보인다.

또한, 주변 버스, 지하철등 대중교통과 연계하여 사용할 것으로 파악된다.

2.4 운영 환경(Operation Environment)

운영환경은 예상 불가한 사용량에유연한 대비를 위해 Azure virtual machine을 사용한다.

서버의 운영체제로는 Ubuntu 18.04으로

데이터베이스와 프론트엔드 웹서버는 내부에서 서버와 가상네트워크망으로 연결하여 사용자에게 제공된다.

한국 지역을 타겟으로 서비스하기에 빠른 응답을 위해 Asia region으로 지역을 선택한다.

2.5 설계 및 구현 제약사항(Design and Implementation constraint)

웹브라우저는 Safari, Chrome, Edge, Firefox의 2017년 이후 모든 버전에서 정상 호환한다.

반응형 디자인을 통해 데스크탑, 모바일, 태블릿의 해상도를 모두 지원한다.

사용자 정보 데이터베이스는 MongoDB를 사용하며, 로그데이터 수집은 Elastic Search를 이용한다.

서버파트의 개발은 Python의 PEP8 규약을 지켜 작성한다

클라이언트와 서버간의 통신은 SSL을 적용한 HTTPS, application 타입은 JSON으로 REST-API 아키텍쳐를 지닌다.

2.6 사용자 문서(User Documentation)

사용자가 회원가입 후 최초 이용 시, 튜토리얼 형식을 통한 사용자 매뉴얼을 제공한다.

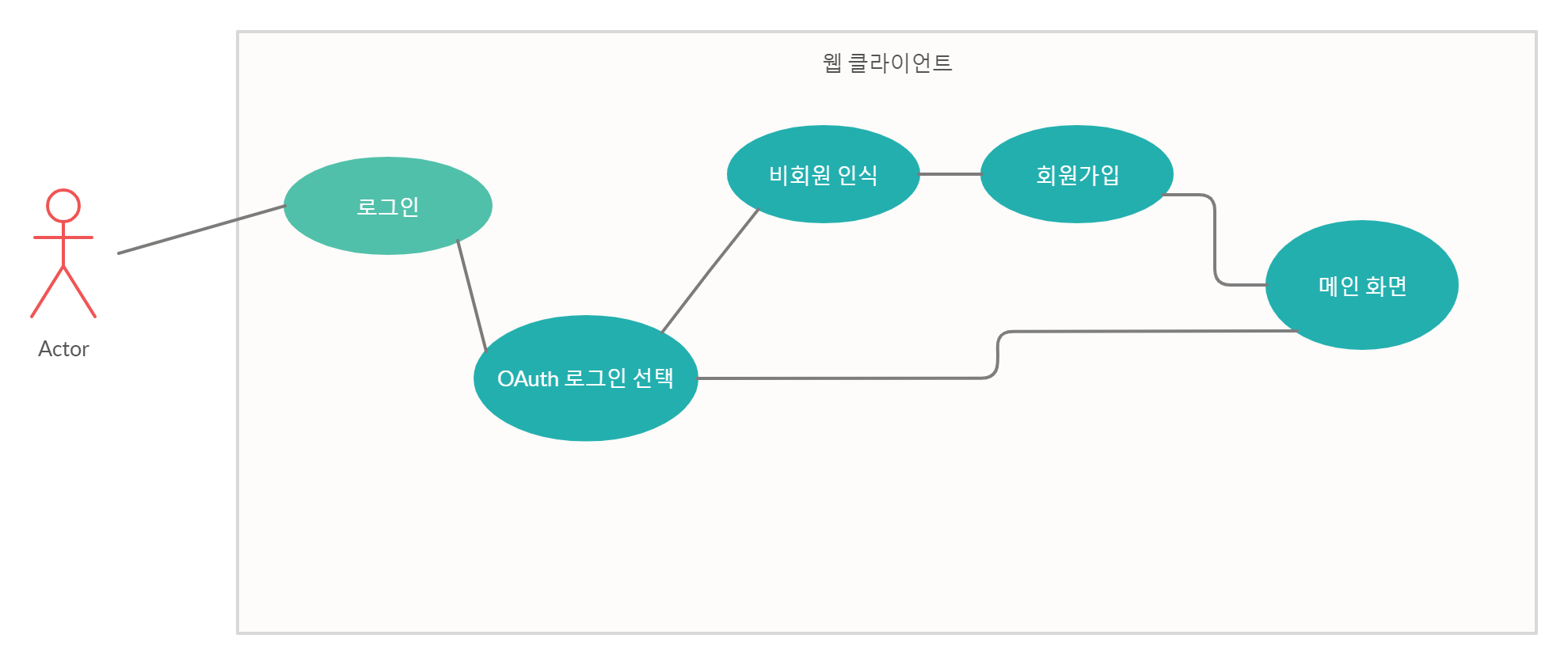
2.7 가정과 종속관계(Assumptions and Dependencies)

서비스 전, 정보보호 및 개인정보보호 관리체계 인증인 ISMS-P을 발급받는다.

3 유스 케이스(Use case)

***3.1 로그인 /회원가입***

가. 요약 시나리오



나. 이벤트 흐름

1. 기본흐름

로그인버튼을 눌러 Oauth 로그인 서비스를 선택하고 로그인 인증이 완료되면 메인화면으로 이동한다.

1. 예외흐름

로그인 버튼을 눌러 Oauth2 로그인 서비스 선택후 로그인 시, 회원가입한 정보가 없는 경우, 비회원으로 인식하여 회원가입 절차로 넘어가 수행한 뒤, 작성완료와 함께 제출하면 메인화면으로 이동한다.

다. 사전조건

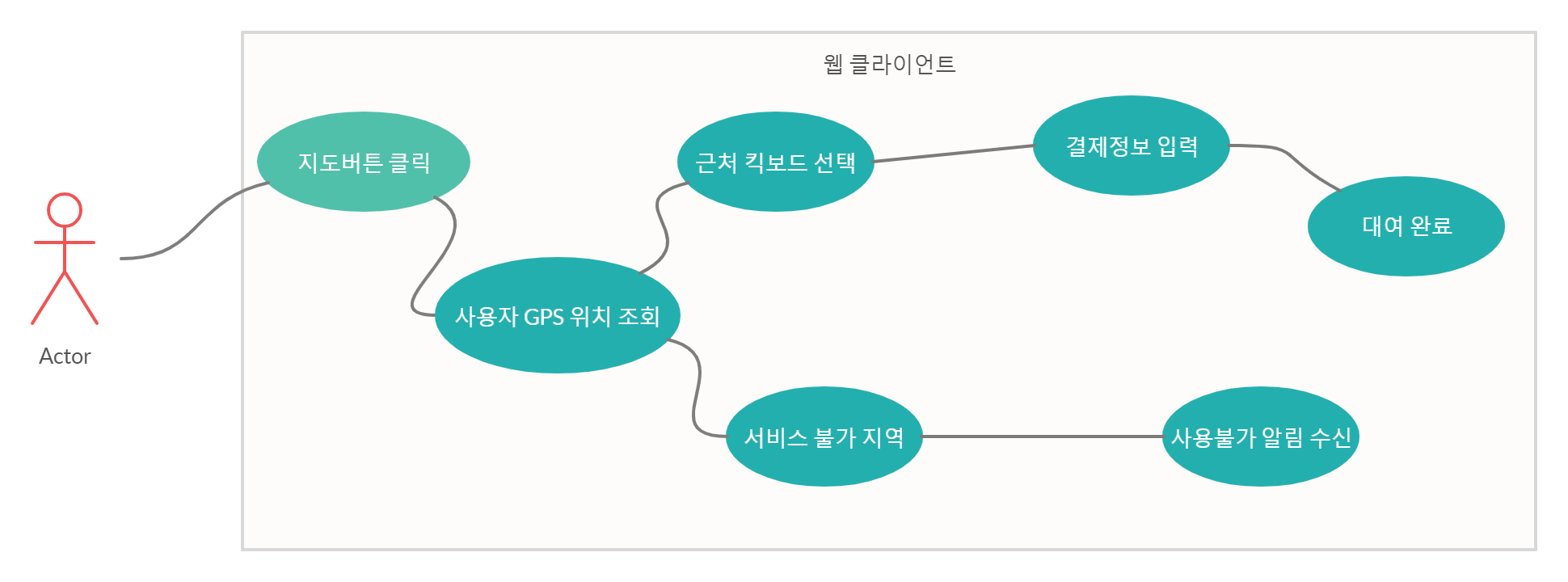
웹브라우저가 로그인 데이터 정보를 가지고 있지않은 상태에서 로그인 표시가 뜬다.

라. 사후조건

메인화면으로 넘어간 뒤 로그아웃시 까지 지속적으로 로그인 유지상태를 가진다.

***3.2 결제***

가. 요약 시나리오



나. 이벤트 흐름

1. 기본흐름

지도버튼을 클릭한 뒤, 사용자 근처 킥보드를 조회하기 위해 GPS를 켜서 자신의 위치를 조회한다.

근처 킥보드를 선택하고, 다음 결제창에서 결제정보를 입력한 뒤 결제를 눌러 대여를 완료한다.

1. 예외흐름

지도버튼을 클릭한 뒤, 사용자가 근처 킥보드 조회위해 GPS를 켜서 자신의 위치를 조회한다.

서비스 불가 지역인 경우, 알림 메세지를 수신받고 이용이 불가능해진다.

다. 사전조건

사용자가 로그인 권한을 가지고 있어야 한다.

라. 사후조건

해당 킥보드의 소유권한을 얻는다.

4 시스템 특징(System Feature)

4.1 시스템 특징 X

가 설명과 우선순위(Description and Priority)

지도조회 기능은 사용자가 대여할 수 있는 근처 킥보드를 보여주고 결제를 위한 이전 단계의 역할을 한다.

결제 기능은 지도에서 조회된 킥보드를 대여하기 위한 절차이다.

로그인 기능은 사용자의 이용내역, 부여 권한 식별을 위해 둔다.

로그인 사용자 정보변경은 사용자의 개인정보가 변경되거나 패스워드 유출로 인해 변경이 필요할때 사용된다

지도 조회, 결제 기능, 로그인 사용자 정보변경 순으로 우선순위를 둔다.

나 자극/응답 순서(Stimulus/Response Sequence)

기능 모두 사용자의 화면 터치 혹은 클릭을 통해 동작한다.

* 로그인
  + 입력 자극 : 입력 후 로그인하기 버튼 클릭
  + 반응 : 조회 후 권한 취득
* 지도 내 킥보드 조회
  + 입력 자극 : 지도 화면 움직이기
  + 반응 : 지도 상 킥보드 위치 표시
* 결제
  + 입력 자극 : 결제 동의 후 결제하기 버튼 클릭
  + 반응 : 연결된 결제수단 과금 후, 킥보드 사용권한 소유

다 기능요구사항(Functional requirement)

R-014

킥보드 이용상세정보 조회가 정상적으로 되지 않는 경우

원활하지 않다는 토스트 메세지를 사용자에게 표시한다

R-016

결제 과정에서 정상 처리되지 않는 오류가 발생하는 경우

결제실패 메세지와 함께 “재시도”, “지도로 이동하기” 버튼을 사용자에게 제공

5 외부 인터페이스 요구사항(External InterfaceRequirement)

5.1 사용자 인터페이스

5.2 소프트웨어 인터페이스(Software Interface)

이에 대한 자세한 내용은 ‘요구사항 정의서’ 문서 참조  
(<https://docs.google.com/spreadsheets/d/1OQcsZIbqa8ORymtCHGat8Q0KXyzzrdlw/edit#gid=711649815>)

5.3 통신 인터페이스(Communications Interface)

HTTP프로토콜과 전송 application type은 JSON으로 지정한다.

6 기능 이외의 다른 요구사항(Other Nonfunctional Requirement)

6.1 성능 요구사항(Performance Requirement)

클라이언트에 전송되는 모든 사용자 로드타임은 2.0초를 넘지 않도록 처리한다.

특정 시간대의 과부하를 방지하고자 동시처리가 가능하게 처리한다.

6.2 안전 요구사항(Safety Requirement)

ISMS-P 개인정보 인증체계를 지킨다.

6.3 보안 요구사항(Security Requirement)

클라이언트와의 통신은 이용 정보에 대해 SHA256를 통해 암호화하여 전송한다.

입력에는 XSS 공격방지와 SQL인젝션 방지를 위한 처리를 한다.

내부 서버간의 처리는 가상네트워크망을 위한 Proxy를 이용해 처리하여 외부접속이 방지되도록 처리한다.

사용자의 패스워드 변경 요청 이후 최초 로그인 시 자동할당된 암호를 즉시 변경해야 한다.

6.4 소프트웨어 품질 특성(Software Quality Attribute)

모든 클라이언트-서버 간 통신은 사용자 로드타임 당 2.0초 이내로 수행되게 한다.

7 다른 요구사항(Other Requirement)\_\_

동일한 응답&연산의 전송은 cache 메모리를 사용하여 자원소모를 줄이며 응답시간을 단축시킨다.