

1. 프로젝트 개요(OverView)

전국적으로 10여개의 전동킥보드 공유서비스가 존재한다.
사용자들이 이 서비스들을 이용할때 각각의 앱을 실행해야하는 불편한 점이 있다.
따라서 이들의 문제를 해결하고자, 우리는 이 공유서비스들의 정보를 한 곳에 모아 보여주는 서비스를 제안하고자 한다.
각 회사에서 운영되는 킥보드의 위치정보, 상태 및 가격과 함께 예약서비스를 사용자에게 하나로 제공하며,
더 경제적인 서비스를 이용할 수 있게 추천해준다.
이로부터 사용자에게 편리성을 제공하고, 기업들에겐 타사의 고객 유치 및 데이터 확보를 통해 서비스를 개선할 수 있을 것으로 기대된다.

1.1 목적(Purpose)

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

1.2 하드웨어와 소프트웨어(Hardware & Software)

1.2.1 개발환경 (Development Environment)

개발환경은 아래와 같다.

구분	TYPE	Spec
서버	OS	Ubuntu 18.04
	클라이언트	Flutter
	back-end	Django 2.1 Apache2 Jenkins 2
	데이터베이스	Amazon RDS

1.2.2 운영 환경(Operation Environment)

운영환경은 예상 불가한 사용량에 유연한 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.

1.2.3 유지 보수 환경(Maintenance Environment)

유지보수 환경은 개발환경과 동일하게 사용한다. 앱의 정상적인 동작을 확인하기 위하여
안드로이드의 경우 7.0이상의 버전을 사용, IOS의 경우 11이상의 버전을 이용한다.

1.3 소프트웨어의 주요기능(Software Functions)

소프트웨어의 주요 기능은 키펀드 상태와 가격, 사용자 위치기반 키펀드 대여가능 위치 정보를
제공합니다.

1.3.1 관리자 기능(Administrator Functions)

관리자의 경우 키펀드 업체의 추가 ,키펀드 추가, 사용자 정보 조회, 키펀드의 현재 위치 정보를
확인 할 수 있습니다.

1.3.2 사용자 기능(User Functions)

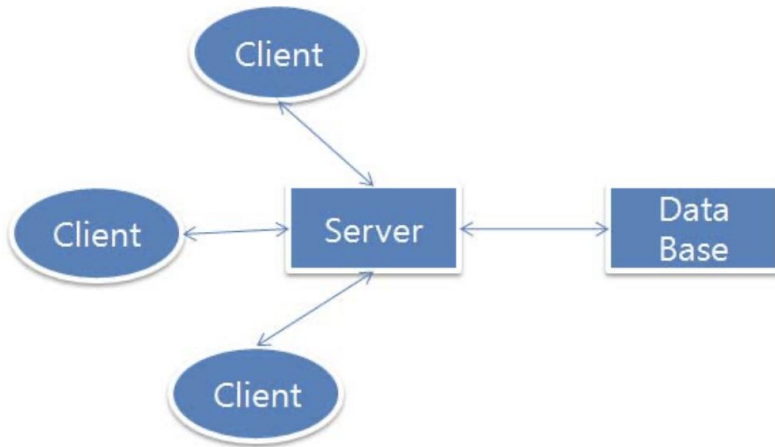
사용자는 사용자 주변에 있는 키펀드의 위치 및 배터리 및 주행가능 거리와 같은 상태를 확인 할 수
있고, 해당 키펀드를 대여, 결제를 할 수 있다. 또한, 키펀드 회사별 이용가격에 대하여 볼 수 있다.

1.4 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
아키텍처를 지닌다.

2. 시스템의 구조(System Structure)

2.1 시스템 구조 개요(System Structure Overview)

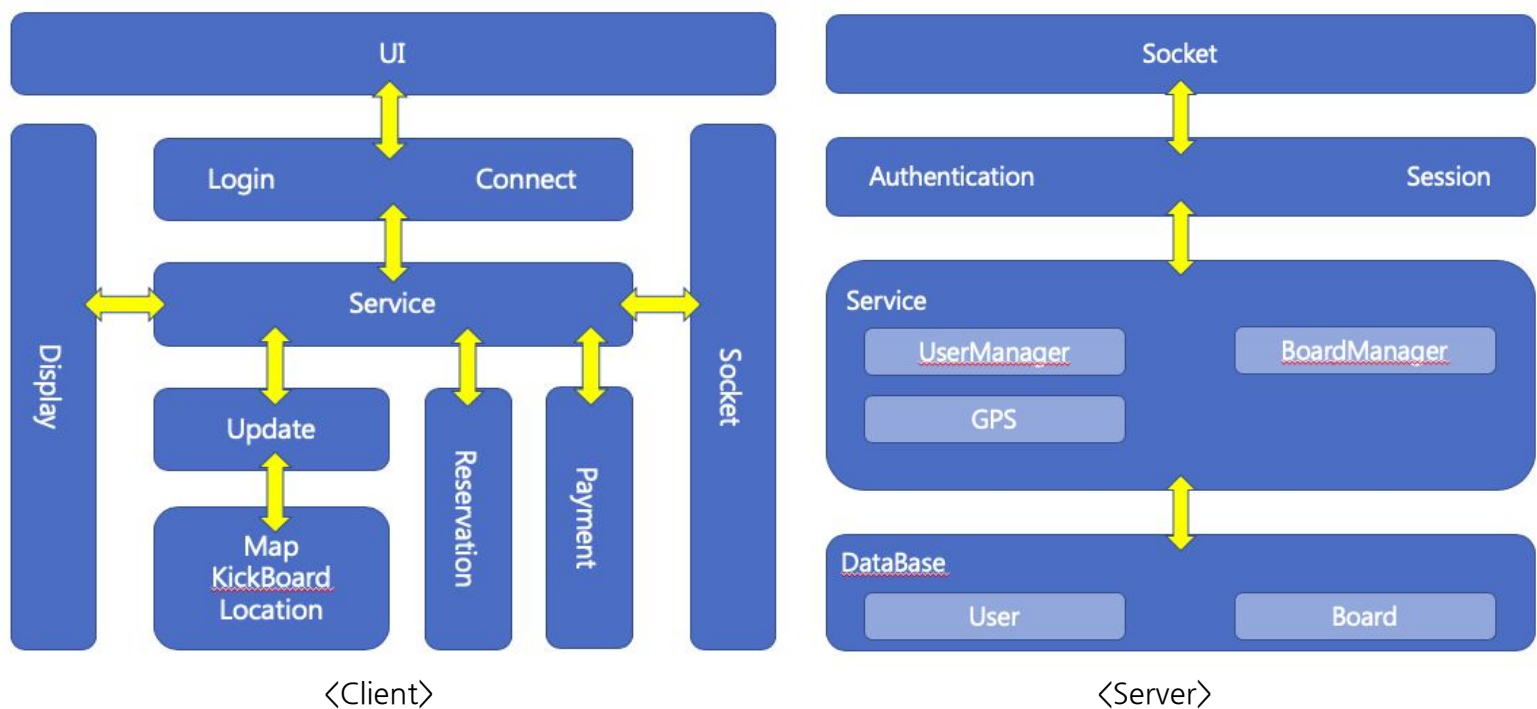


이 부분의 목적은 시스템의 요구사항을 만족하는 시스템 조직을 만들 수 있도록 구조를 설계하는 것이다.

이 시스템의 구조도는 시스템의 전체적인 흐름을 쉽게 알 수 있도록 간략히 그림으로 표현하였다.

2.2 시스템 구조도

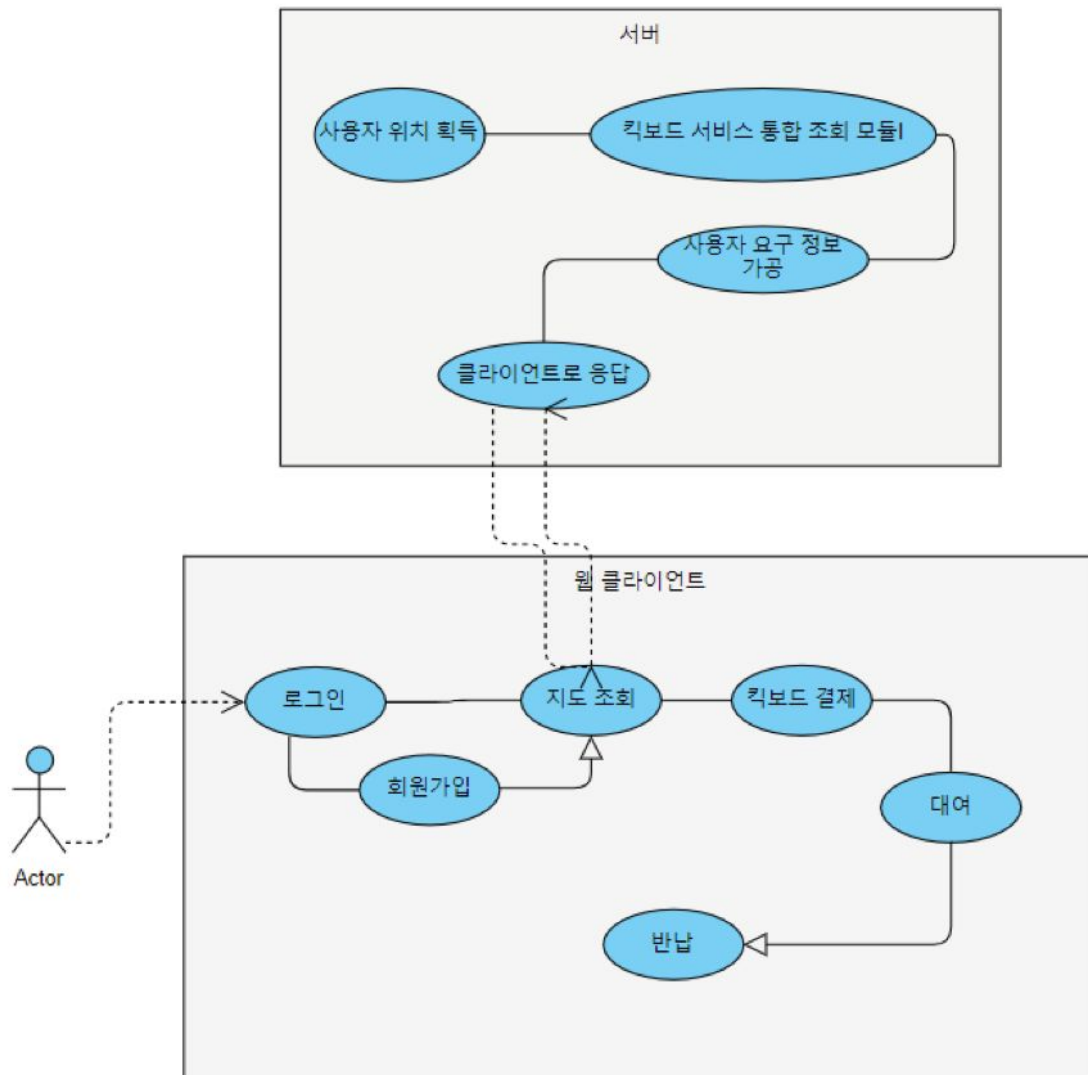
시스템은 크게 Server와 Client로 구성되어 있다. 기본 구조는 아래와 같다.



3. 모듈 설계(Module Design)

3.1 모듈 개요(Module Overview)

이 부분의 목적은 모듈의 구분, 표현을 나타내고 해당 모듈이 어떠한 동작을 하는지에 대하여 서술한다.



<동작 Flow>

3.1.1 모듈의 구분 및 표(Module Classification, Contents)

사용자 모듈, 키패드 모듈, 결제 모듈, 위치정보 모듈로 모듈을 나눈다.
User Module, KBoard Module, Pay Module, Location Module

3.2 모듈별 상세 설계(Module Detail Design)

3.2.1 모듈 목록(Module Type)

<User Class>

userID - 사용자 ID / 식별용
userKBoard - 사용중인 키패드 ID
payInfo - 사용자의 결제 정보
StartUseTime - 이용 시작 시간
EndUseTime - 이용 종료 시간

uLogin() - 사용자 로그인
uLogout() - 사용자 로그아웃
uSelectKBoard() - 사용자가 키패드 선택, 정보 불러오기

<KBoard Class>

kBoardID - 키패드 ID / 식별용
kCompany - 키패드 제공 회사
kLoc - 키패드 위치
kUserID - 키패드 사용중인 사용자 ID
kBattery - 남은 배터리 잔량
kAvailDist - 주행가능 거리

kLock(kBoardID, kUserID) - 키패드 대여시 더이상 지도에 표시x & 이용 시작
kRelease(kBoardID) - 키패드 반납시 지도에 다시 표시 & 이용 종료
kShow(kBoardID) - 사용자가 키패드 선택시 해당 키패드의 상태 정보 표시
setKLoc(kBoardID) - 키패드의 불러와서 위치 설정
getKLoc(kBoardID, kLoc) - 키패드의 위치 불러와서 화면에 표시, Location.getLoc() method
Overriding
getKStatus(kBoardID) - 키패드의 상태 불러오기

<Pay Class>

Charge - 결제 요금, CheckCharge로 부터 return 받음
PayNo - 결제 번호

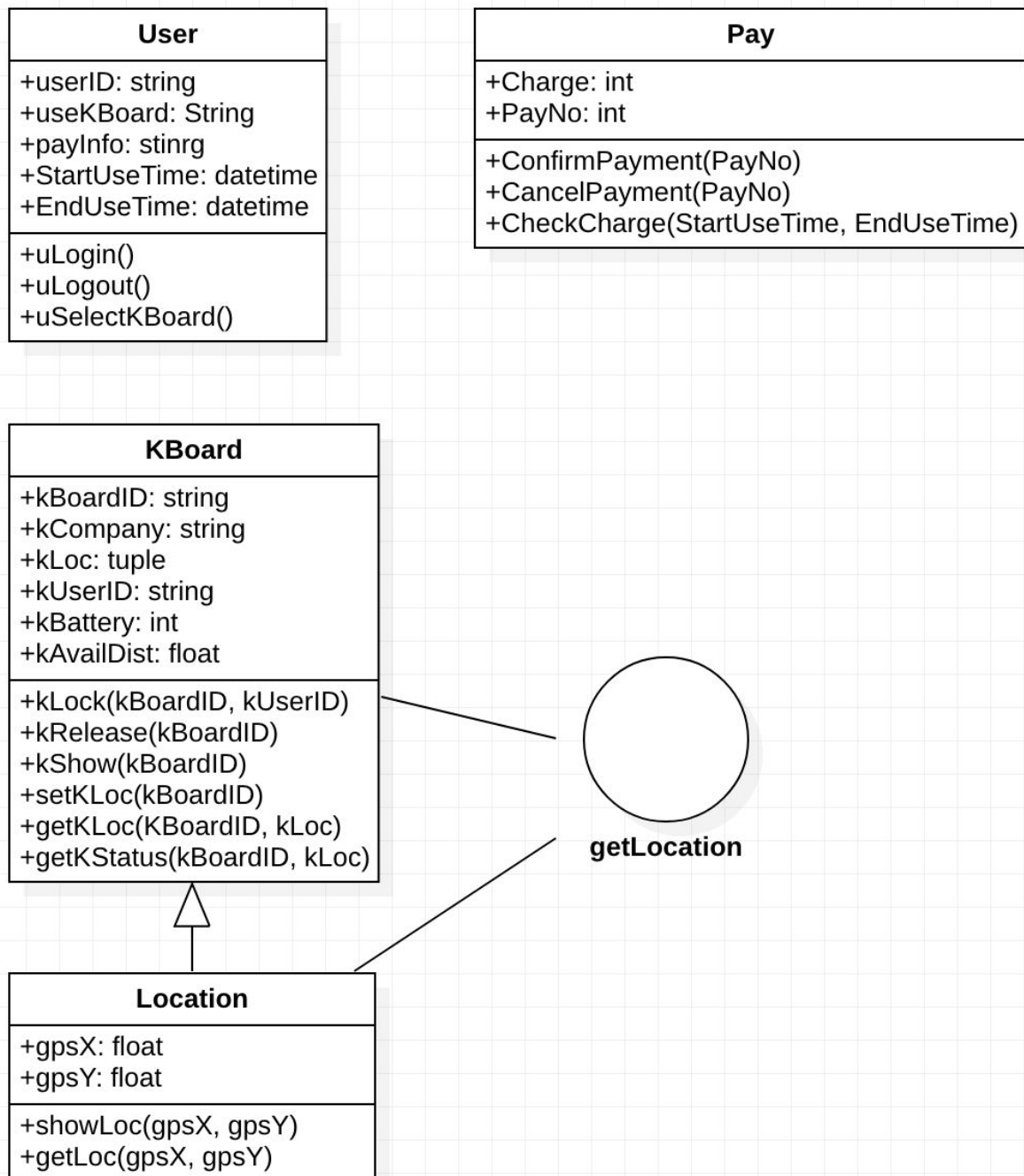
ConfirmPayment() - 결제하기
CancelPayment() - 결제 취소 요청
CheckCharge - (User.EndUseTime - User.StartUseTime)로 결제 금액 계산

<Location Class>

gpsX - gps의 위도 값
gpsY - gps의 경도 값

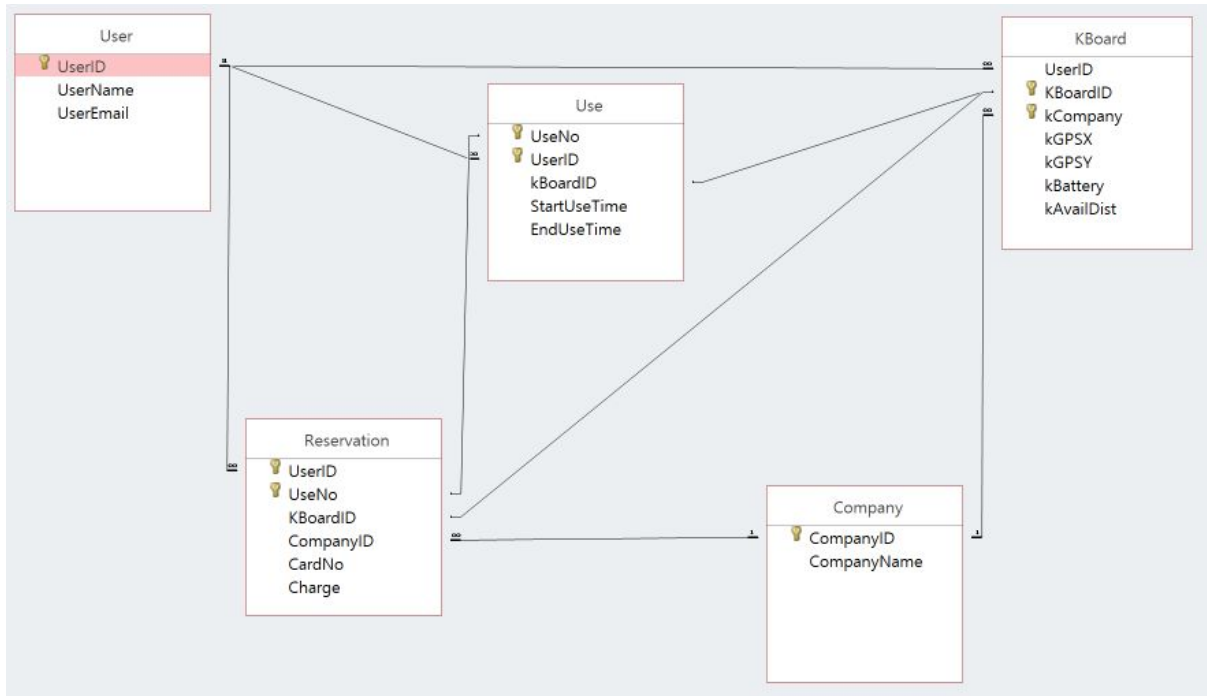
showLoc(gpsX, gpsY) - 위도 경도 계산해서 값 리턴
getLoc(gpsX, gpsY) - 키패드와 통신하여 위도, 경도 값 가져옴

3.2.2 상세 설계서(Detail N-S Chart)



4. 파일 구조 및 데이터베이스 설계(Data Structure Design)

4.1 자료의 논리적 구조(E-R Diagram)



4.2 자료의 물리적 구조(Physical Structure Table)

```
[mysql> desc User;
```

Field	Type	Null	Key	Default	Extra
UserID	int(11)	NO	PRI	NULL	
UserName	char(20)	NO		NULL	
UserEmail	char(50)	NO		NULL	

```
3 rows in set (0.00 sec)
```

```
[mysql> desc Company;
```

Field	Type	Null	Key	Default	Extra
CompanyID	int(11)	NO	PRI	NULL	
CompanyName	char(30)	NO		NULL	

```
2 rows in set (0.01 sec)
```

```
[mysql> desc User_Use;
```

Field	Type	Null	Key	Default	Extra
UseNo	int(11)	NO	PRI	NULL	
UserID	int(11)	NO	PRI	NULL	
kBoardID	char(50)	NO		NULL	
StartUseTime	timestamp	NO		NULL	
EndUseTime	timestamp	NO		NULL	

```
5 rows in set (0.00 sec)
```

```
[mysql> desc Reservation;
```

Field	Type	Null	Key	Default	Extra
UseNo	int(11)	NO	PRI	NULL	
UserID	int(11)	NO	PRI	NULL	
kBoardID	char(50)	NO	MUL	NULL	
CompanyID	int(11)	NO	MUL	NULL	
CardNO	char(16)	NO		NULL	
Charge	int(11)	NO		NULL	

```
6 rows in set (0.00 sec)
```

```
[mysql> desc KBoard;
```

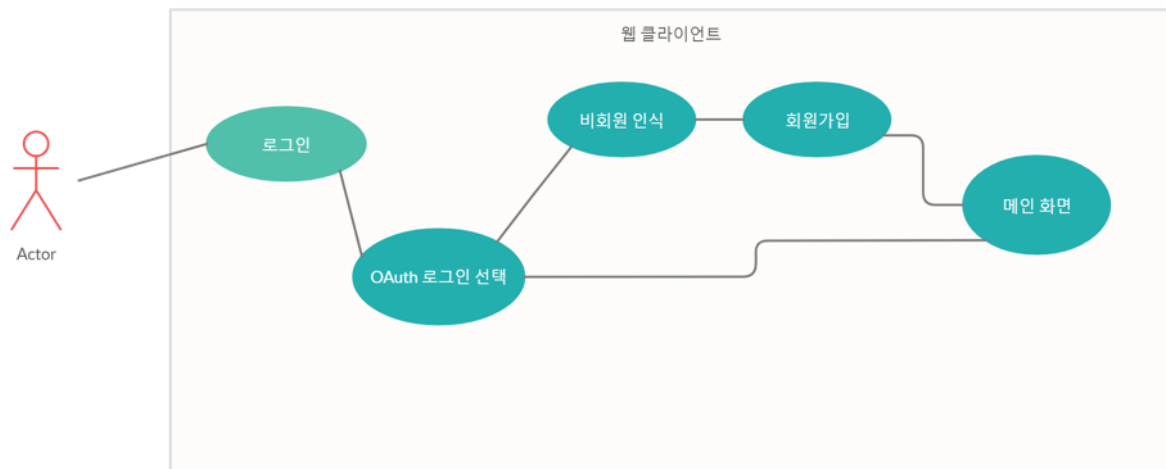
Field	Type	Null	Key	Default	Extra
UserID	int(11)	NO	MUL	NULL	
kBoardID	char(50)	NO	PRI	NULL	
kCompany	int(11)	NO	PRI	NULL	
kGPSX	float	YES		NULL	
kGPSY	float	YES		NULL	
kBattery	int(11)	YES		NULL	
kAvailDist	float	YES		NULL	

```
7 rows in set (0.01 sec)
```


5. 사용 사례와 예시(Use case)

5.1 로그인 / 회원가입 (Sign In / Sign Up)

5.1.1 요약 시나리오



5.1.2 이벤트 흐름

5.1.2.1 기본 흐름

로그인 버튼을 눌러 OAuth 로그인 서비스를 선택하고 로그인 인증이 완료되면 메인화면으로 이동한다.

5.1.2.2 예외 흐름

로그인 버튼을 눌러 OAuth 로그인 서비스를 선택 후 로그인 시, 회원가입한 정보가 없는 경우 비회원으로 인식, 회원가입 절차로 넘어가 가입을 수행한 뒤, 가입 완료 후 메인화면으로 이동한다.

5.1.2.3 사전 조건

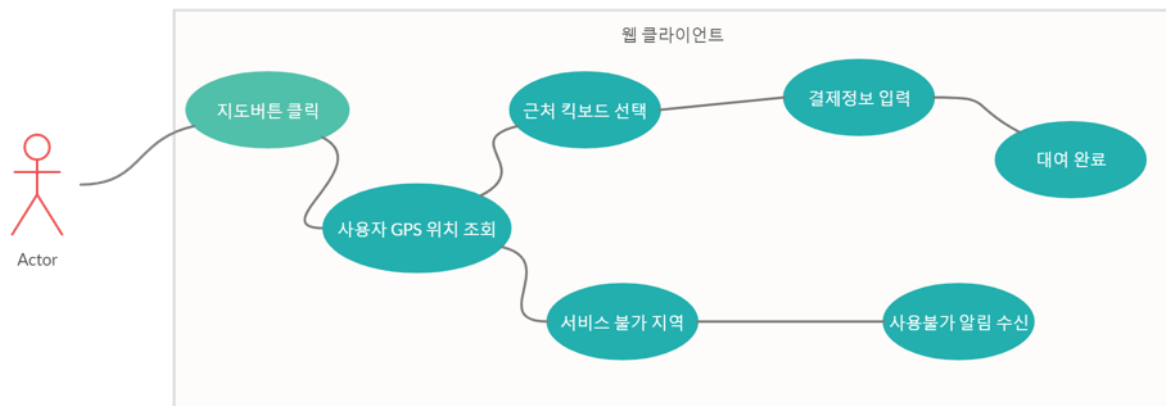
웹 브라우저가 로그인 데이터 정보를 가지고 있지 않은 상태에서 로그인 표시가 뜬다.

5.1.2.4 사후 조건

메인화면으로 넘어간 뒤 사용자가 앱을 종료 혹은 로그아웃을 할 때까지 로그인 상태를 유지한다.

5.2 선택 및 결제 (Select & Payment)

5.2.1 요약 시나리오



5.2.2 이벤트 흐름

5.2.2.1 기본 흐름

지도 버튼을 클릭한 뒤, 사용자 근처 킷보드 조회를 위해 GPS 권한을 요청하여 사용자의 위치를 조회한다.
이후 근처 킷보드를 선택하고, 결제 정보를 입력 후 대여를 완료한다.

5.2.2.2 예외 흐름

지도 버튼을 클릭한 뒤, 사용자 근처 킷보드 조회를 위해 GPS 권한을 요청하여 사용자의 위치를 조회한다.
이때, 사용자가 서비스 불가 지역에 있는 경우, 사용자에게 서비스 지역이 아니라는 알림 메시지를 띄운 후, 사용자가 서비스 지역으로 들어 갈 때까지 이용이 불가능해진다.

5.2.2.3 사전조건

사용자가 로그인 권한을 가지고 있어야하며, 사용자가 결제 정보를 입력해 두어야 한다.

5.2.2.4 사후조건

사용자가 이용 종료를 요청 할 때 까지 해당 킷보드의 이용 권한을 얻는다.

6. 외부 인터페이스 요구사항(External Interface Requirement)

6.1 소프트웨어 인터페이스(Software Interface)

이에 대한 자세한 내용은 '요구사항 정의서' 문서 참조

(<https://docs.google.com/spreadsheets/d/1OQcsZlbqa8ORymtCHGat8O0KXyzzrdlw/edit#gid=711649815>)

6.2 통신 인터페이스(Communications Interface)

HTTP프로토콜과 전송 application type은 JSON으로 지정한다.

7 기능 이외의 다른 요구사항(Other Nonfunctional Requirement)

7.1 성능 요구사항(Performance Requirement)

클라이언트에 전송되는 모든 사용자 로드타임은 2.0초를 넘지 않도록 처리한다.
특정 시간대의 과부하를 방지하고자 동시처리가 가능하게 처리한다.

7.2 안전 요구사항(Safety Requirement)

ISMS-P 개인정보 인증체계를 지킨다.

7.3 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.

7.4 소프트웨어 품질 특성(Software Quality Attribute)

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

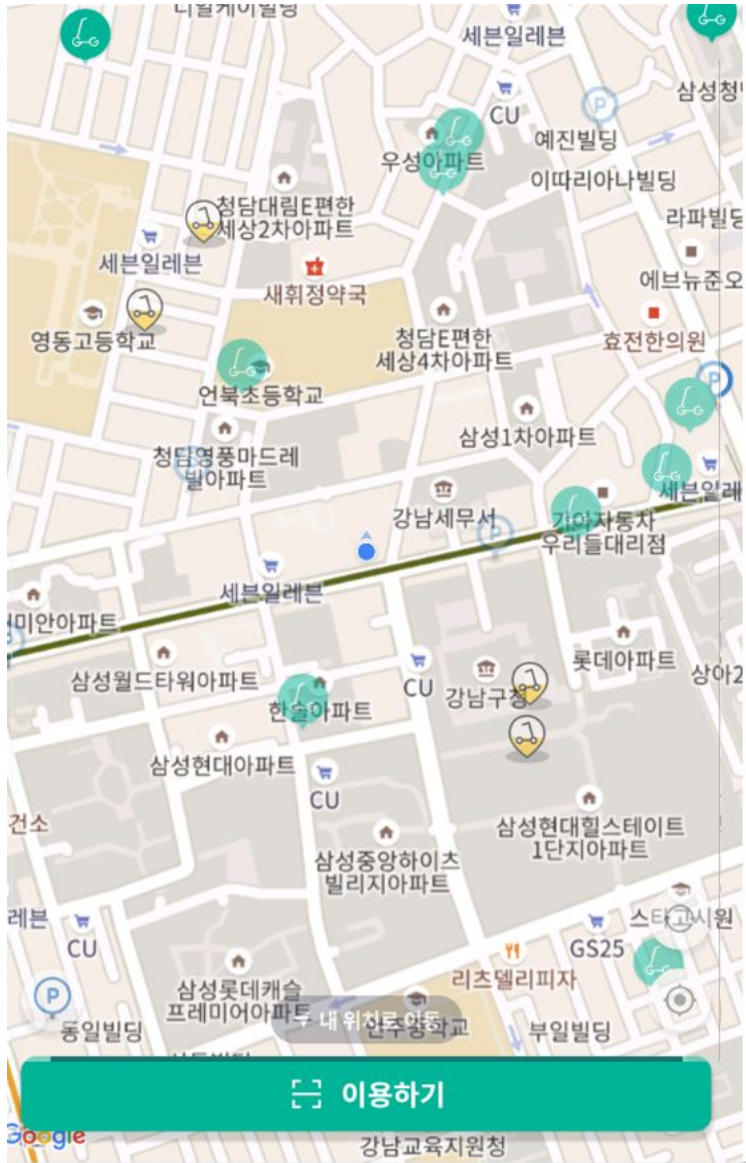
7.5 다른 요구사항(Other Requirement)

동일한 응답&연산의 전송은 Cache메모리를 이용하여 자원의 소모를 줄이며 응답시간을 단축시킨다.

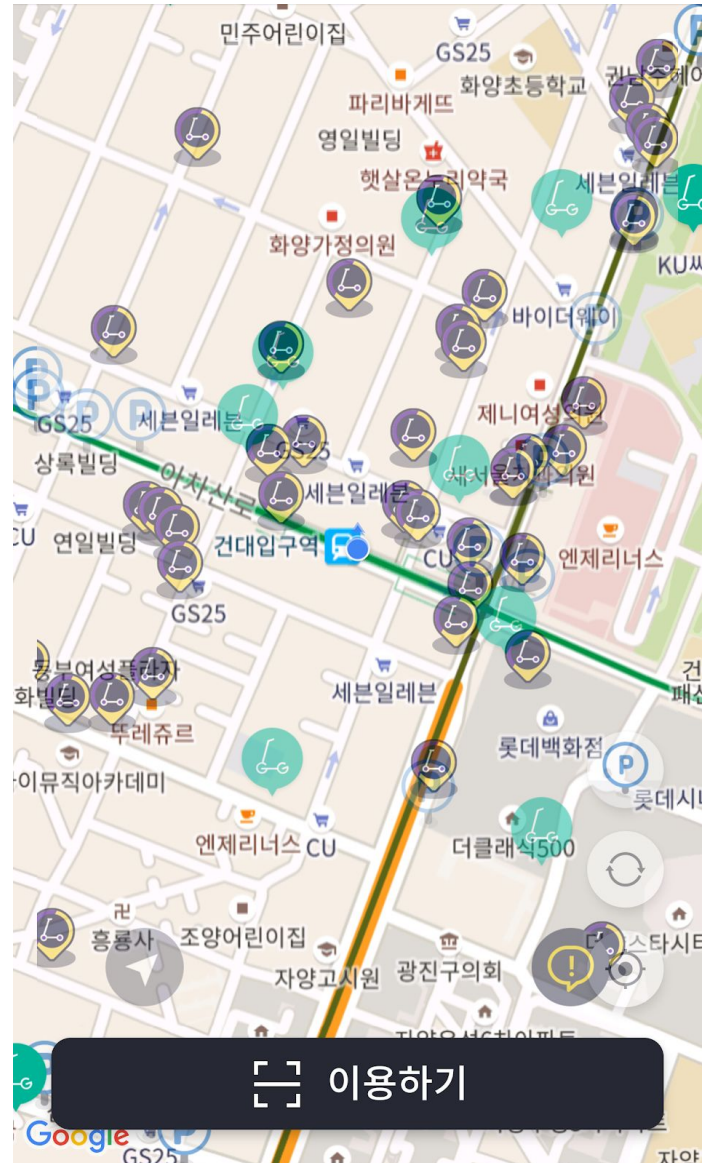
8. 주요화면 및 GUI 예시

8.1 앱 UI

아래의 사진은 해당 앱이 최종적으로 보여 주어야 할 메인 화면이다.



앱 예시1 - 강남구청



앱 예시2 - 건대입구

8.2 웹 UI

웹에서의 화면은 아래의 예시와 같이 유사하게 디자인 하여, 사용자가 한눈에 볼 수 있도록 단순하고 깔끔한 UI를 추구한다.

