

---

# 알고리즘 기반 신규 교통 서비스

## TANNAE Mobile Application

---



과 목 명 :	소프트웨어공학 종합프로젝트	
	Capstone Design	
담당교수 :	김경백	
지도교수 :	임형석	
학 과 :	소프트웨어공학전공	
팀 명 :	T7	
	171014	김동현
팀 원 :	175756	이승찬
	175768	최재원

지도교수 서명란

Chonnam National University

# 목 차

GitHub Link.....	<a href="#">codesver/Capstone: Capstone Design (github.com)</a>
<b>프로젝트 개발 계획서.....</b>	<b>3</b>
프로젝트 개요.....	4
개발 목표.....	6
요구사항 목록.....	7
개발 인원 및 일정.....	12
기타 사항.....	14
<b>프로젝트 분석.....</b>	<b>15</b>
요구사항 기술서.....	16
Use case diagram.....	20
Use case 기술서.....	21
순차 다이어그램 분석.....	36
클래스 다이어그램 분석.....	49
<b>프로젝트 설계.....</b>	<b>51</b>
클래스 다이어그램 설계.....	52
순차 다이어그램 설계.....	53
User Interface 설계.....	68
Database Table 명세서.....	89
<b>프로젝트 구현 .....</b>	<b>94</b>
구현.....	95
알고리즘 효율성.....	96
기능 테스트.....	97
<b>마무리 .....</b>	<b>99</b>
활동.....	100
질의 사항.....	103

# 프로젝트 개발 계획서

## 1. 프로젝트 개요

### 1.1 주제

#### 1.1.1 주제 설명

프로젝트 'TANNAE'는 기존의 존재하는 교통수단들의 장단점을 병합한 서비스를 제공하는 모바일 애플리케이션을 개발하는 프로젝트이다. 해당 서비스는 버스의 경제성과 택시의 편의성을 병합한 서비스로 비슷한 경로의 탑승객들이 유동적으로 차량에 탑승하고 하차하며 같이 탑승한 인원끼리 비용을 나누어서 낸다. 이에 따라 버스보다는 적은 이동시간으로 원하는 목적지에 도착하며 택시보다는 경제적인 가격으로 서비스를 이용할 수 있다.

#### 1.1.2 주제 선정 배경 및 목적

학교 근처에서 자취하지 않는 대학생들은 보통 본가에서 버스를 타고 학교로 등하교한다. 이런 학생들에게 이동시간은 생각보다 큰 스트레스로 다가온다. 그렇다고 택시를 매일 타기에는 비용이 부담된다. 복구에 있는 전남대학교와 광산구, 남구 사이의 이동시간은 약 40~60분 정도이며 시간을 절약하기 위해 택시를 타면 10,000원이라는 상대적으로 큰 비용이 발생한다. 이는 대학생뿐만이 아니라 대중교통을 자주 이용하는 많은 사람의 불편 사항이다. 이러한 불편함을 해소하기 위해 이번 프로젝트를 진행하게 되었다.

버스와 택시에서는 이동시간과 비용을 동시에 최소화하는 방법을 찾기가 어렵다. 이에 따라 기존의 서비스를 병합하여 좀 더 효율적인 교통 서비스를 제공하고자 한다. 소프트웨어 공학전공 학생으로서 이번 프로젝트는 서비스 자체가 아닌 보편화된 모바일 기기로 손쉽게 신규 교통 서비스를 이용할 수 있는 모바일 앱을 개발하는 것을 목적으로 한다.

### 1.2 프로젝트

#### 1.2.1 필요성

오랜 세월 동안 사람들은 버스와 택시를 이용해왔고 각각의 교통수단들을 서로의 단점들을 보완해주었다. 하지만 이동시간이 늘어나는 현대 사회에서 경제성과 효율성을 모두 충족시키는 대중교통이 이제는 필요한 상황이며 이번 프로젝트를 통해 생활 속에 새롭게 자리잡을 교통 서비스를 구상하고 이를 제공하는 모바일 애플리케이션을 개발하고자 한다.

#### 1.2.2 기대효과

e-나라 지표에 따르면 2011년부터 2020년까지의 차량 증가량은 약 600만대로 인구 증가량의 6배 정도가 된다. 또한 기술의 발달로 폐차 주기도 늘어나면서 차량 수가 늘어나고 있다. 이에 따라 교통량 증가와 환경 오염이 심해지고 있다. 이번 프로젝트는 대중교통을 활성화하여 이러한 문제들을 어렵지 않게 해결할 것으로 기대된다.

## 1.3 유사 서비스 및 개선사항

### 1.3.1 Kakao Taxi

가장 대표적인 교통 서비스 제공 애플리케이션이다. 일반적인 택시 서비스를 쉽게 이용할 수 있게 해주는 서비스이며 택시뿐만이 아니라 카카오택시, 대리운전, 퀵/택배, 렌터카 서비스도 제공한다. 현재 가장 활성화된 이동 수단 관련 서비스 앱이다.

손쉬운 방법으로 서비스를 제공하여 주지만 여전히 각 서비스의 비용은 반복적으로 사용하기에는 어려운 정도이다. 이번 프로젝트에서 제공하는 서비스는 이러한 비용적 측면을 개선하여 줄 것으로 예상된다.



### 1.3.2 반반 택시

프로젝트 구상과 가장 유사한 서비스이다. 같은 방향의 승객과 같이 타고 요금을 최대 50% 할인받는다. 반반 호출 서비스뿐만이 아니라 일반 호출 서비스 또한 제공하여 일반 택시 사용자들한테도 유용하다.

선호 운행 설정으로 내비게이션 경로 운행, 안전 운행, 불필요한 대화 금지 등 탑승 전에 택시 기사에게 요청 사항을 전할 수 있으며 포인트 제도를 통해 경제적인 효율성도 갖추는 부분은 이번 프로젝트에서 참고할만한 부분이다.

하지만 반반 호출 서비스는 출발지와 도착지가 비슷한 사람끼리 요금을 반으로 나누어 내지만 'TANNAE' 서비스는 출발지와 도착지가 상이하더라도 중간 경로가 비슷하면 중간에 동승을 하여 좀 더 유동적인 승하차가 가능하다.



### 1.3.3 우버

우버 택시는 기본적으로 탑승자와 기사의 구별이 없다는 것이 큰 특징이다. 개인이 자신의 차량을 우버 택시로 등록할 수 있으며 택시 면허가 없이 자신이 기사가 되어 우버 서비스를 제공할 수 있다.

하지만 우버 택시는 성공하지 못하였다. 이는 크게 택시 업체의 반발, 차별화 실패, 기존의 편한 운송 서비스 때문이다. 이번 프로젝트로 제공되는 새로운 서비스는 기사와 탑승자를 구별하고 기존의 택시 기사들을 기사로 우선 선발할 것이다. 또한 독자적인 배차 알고리즘 개발을 통해 기존 서비스를 개선하고 차별성을 둘 예정이다.



## 2. 개발 목표

### 2.1 개인적 측면

#### 2.1.1 탑승자의 경제적 이익

- 1) 문제 : 비용적인 측면 때문에 택시를 자주 이용하거나 혼자 이용하기에 부담
- 2) 목표 : 경로가 비슷한 탑승자들이 차량을 같이 이용하여 교통비를 나누어서 결제하여 비용에 대한 부담감을 완화

#### 2.1.2 운전자의 경제적 이익

- 1) 문제 : 탑승자가 있지 않은 상황에서 비용적으로 손실이 발생
- 2) 목표 : 탑승자가 있는 상황에서도 다음 탑승자를 받을 수 있어서 불필요한 이동이 발생하지 않아서 비용 절약

### 2.2 사회적 측면

#### 2.2.1 배기가스로 인한 환경 오염 감소

- 1) 문제 : 개인 차량의 증가로 인해 전체 차량 수가 증가하며 이에 따라 차량 배기가스의 배출량이 증가해 대기오염이 계속해서 발생
- 2) 목표 : 차량을 공유하여 서비스를 이용하고 대중교통을 활성화하여 전체 차량 수를 감소시켜 결과적으로 배기가스 배출량과 차량 제작 부수물 감소를 통해 환경 오염을 개선

#### 2.2.2 교통체증 완화

- 1) 문제 : 개인 차량을 포함한 여러 차량의 증가로 많은 차량이 운행되며 이에 따라 혼잡한 교통 상황이 반복적으로 발생
- 2) 목표 : 대중교통 활성화를 통해 개인 차량을 감소시켜 교통량 완화

### 3. 요구사항 목록

#### 3.1 계정 관련 기능

##### 3.1.1 회원 가입 ( REQ-F-01 )

- 1) 사용자 동작
  - (1) Sequence : 사용자 정보 입력 → 회원 가입 요청
  - (2) 사용자 입력 : ID/PW, 사용자 정보(계정 정보, 개인정보, 사용자 종류, 차량 정보)
- 2) 처리
  - (1) UI 출력 : 가입 화면 출력, 가입 완료 시 홈 화면으로 전환
  - (2) 내부 처리 : 회원 가입 완료 시 User DB update

##### 3.1.2 로그인 ( REQ-F-02 )

- 1) 사용자 동작
  - (1) Sequence : 사용자 계정 정보 입력 → 로그인 요청
  - (2) 사용자 입력 : 사용자 계정 정보 ID/PW
- 2) 처리
  - (1) UI 처리 : 로그인 화면 출력, 비정상 로그인 시 오류 메시지 출력
  - (2) 내부 처리  
User DB에서 사용자를 검색하여 로그인 요청 승인/오류 메시지 반환  
로그인 시 서비스 제공자/이용자/관리자를 판단하여 해당 기능 (비)활성화

##### 3.1.3 계정 정보 찾기 ( REQ-F-03 )

- 1) 사용자 동작
  - (1) Sequence  
사용자 정보 입력 → 회원 정보 검색 요청 → 전송된 PIN 입력 → 계정 정보 수신
  - (2) 사용자 입력 : 사용자 정보, 본인 확인용 PIN 번호
- 2) 처리
  - (1) UI 출력 : 사용자 정보 입력창 출력, PIN 번호 입력창 출력, 계정 정보 출력
  - (2) 내부 처리  
사용자 정보와 PIN 번호를 통해 본인 확인  
본인 확인이 완료되면 사용자 정보를 바탕으로 ID/PW 정보 검색

#### **3.1.4 계정 정보 수정 ( REQ-F-04 )**

1) 사용자 동작

(1) Sequence : 사용자 계정 정보 선택 → 정보변경 선택 → 신규 데이터 입력 → 변경

(2) 사용자 입력 : 수정 데이터

2) 처리

(1) UI 출력 : 계정 정보 화면 출력

(2) 내부 처리 : 사용자 정보를 변경하면 User DB 업데이트

#### **3.1.5 회원 탈퇴 ( REQ-F-05 )**

1) 사용자 동작

(1) Sequence : 계정 관리 화면 선택 → 회원 탈퇴 선택 → 재확인 알림 확인 → 탈퇴

2) 처리

(1) UI 출력 : 계정 관리 화면 출력, 회원 탈퇴 재확인 알림 출력

(2) 내부 처리 : 회원 탈퇴 시 User DB에서 삭제



## 3.2 운전자 서비스

### 3.2.1 서비스 요청 확인 ( REQ-F-06 )

- 1) 사용자 동작
  - (1) Sequence : 요청된 서비스 확인 및 경로 수정
- 2) 처리
  - (1) UI 출력 : 탑승자가 해당 차량 배차를 요청하면 배차 차량 화면에 관련 정보 출력
  - (2) 내부 처리 : 탑승자 정보를 바탕으로 Vehicle DB 업데이트

### 3.2.2 운행 여부 변경 ( REQ-F-07 )

- 1) 사용자 동작
  - (1) Sequence : 차량 정보 설정 선택 → 운행/비운행 상태 선택
  - (2) 사용자 입력 : 운행/비운행 여부
- 2) 처리
  - (1) UI 출력 : 차량 정보 화면 출력, 운행/비운행 toggle button
  - (2) 내부 처리 : Vehicle DB 업데이트

## 3.3 탑승자 서비스 기능

### 3.3.1 배차 요청 ( REQ-F-08 )

- 1) 사용자 동작
  - (1) Sequence  
배차 신청 → 경로 선택 → 동승 여부 선택 → 배차
  - (2) 사용자 입력 : 출발지, 목적지
- 2) 처리
  - (1) UI 출력  
사용자가 배차를 신청할 때 동승 여부 선택
  - (2) 내부 처리  
배차 알고리즘에 따라 최적합 차량 배차  
배차가 완료되면 Vehicle DB 업데이트

## 3.4 사용자 서비스 기능

### 3.4.1 분실물 찾기 ( REQ-F-09 )

- 1) 사용자 동작
  - (1) Sequence  
사용자 서비스 화면에서 '분실물 찾기' 선택 → 분실물 리스트에서 조회
- 2) 처리
  - (1) UI 출력  
분실물 찾기 화면(분실물 리스트) 출력
  - (2) 내부 처리 : Lost DB에 있는 개체들을 분실물 리스트에 출력

### 3.4.2 QnA ( REQ-F-10 )

- 1) 사용자 동작
  - (1) Sequence  
사용자 서비스 화면에서 'QnA' 선택 → QnA 리스트 조회 및 질문 작성  
관리자 계정으로 로그인하였을 경우 답변 작성
  - (2) 사용자 입력 : 질문사항 내용, 답변 내용
- 2) 처리
  - (1) UI 출력 : QnA 리스트 화면 출력, 질문 작성 화면 출력
  - (2) 내부 처리  
Content DB에서 리스트 출력  
새로운 QnA가 작성되었을 경우 Content DB 업데이트

### 3.4.3 FAQ ( REQ-F-11 )

- 1) 사용자 동작
  - (1) Sequence  
사용자 서비스 화면에서 'FAQ' 선택
- 2) 처리
  - (1) UI 출력 : FAQ 리스트 화면 출력
  - (2) 내부 처리 : Content DB에서 FAQ를 조회하여 반환

#### 3.4.4 이용 기록 조회 ( REQ-F-12 )

- 1) 사용자 동작
  - (1) Sequence  
사용자 서비스 화면에서 'History' 선택
- 2) 처리
  - (1) UI 출력 : History 화면 출력
  - (2) 내부 처리 : History DB에서 사용자의 이용 리스트를 검색

### 3.5 결제 서비스

#### 3.5.1 결제 및 평가 (운전자) ( REQ-F-13 )

- 1) 사용자 동작
  - (1) Sequence  
한 번의 운행이 종료되면 운행 종료 입력
- 2) 처리
  - (1) UI 출력 : 운행 종료 화면 출력
  - (2) 내부 처리  
Vehicle DB에서 운행 여부를 변경

#### 3.5.2 결제 및 평가 (탑승자) ( REQ-F-14 )

- 1) 사용자 동작
  - (1) Sequence : 결제 알림 수신 → 결제 정보 확인 → 서비스 평가
  - (2) 사용자 입력 : 운전자 평가 평점
- 2) 처리
  - (1) UI 출력 : 결제 영수증 출력, 운전자 평가 화면 출력
  - (2) 내부 처리 : History DB 업데이트 및 User DB의 운전자 평점 업데이트

#### 3.5.3 포인트 충전 ( REQ-F-15 )

- 1) 사용자 동작
  - (1) Sequence  
포인트 관리 화면 선택 → 포인트 충전 선택 → 포인트 충전량 입력 → 충전
  - (2) 사용자 입력 : 포인트 충전량
- 2) 처리
  - (1) UI 출력 : 포인트 관리 화면 출력, 포인트 충전 dialog 출력
  - (2) 내부 처리 : User DB에서 point 정보를 검색 및 업데이트

## 4. 개발 인원 및 일정

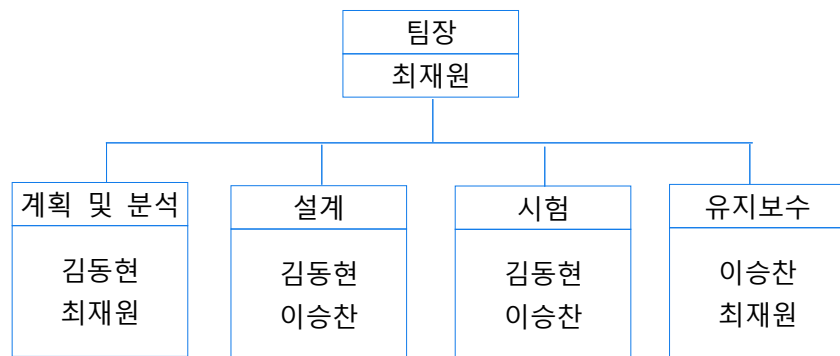
### 4.1 팀원 구성

성명	학과	학년	연락처	이메일
김동현	소프트웨어공학	4	010-9130-9942	qhfhdl12@naver.com
이승찬	소프트웨어공학	4	010-9352-3085	lsc4947@naver.com
최재원	소프트웨어공학	4	010-4790-2388	sumnerwon@naver.com

### 4.2 업무 분담

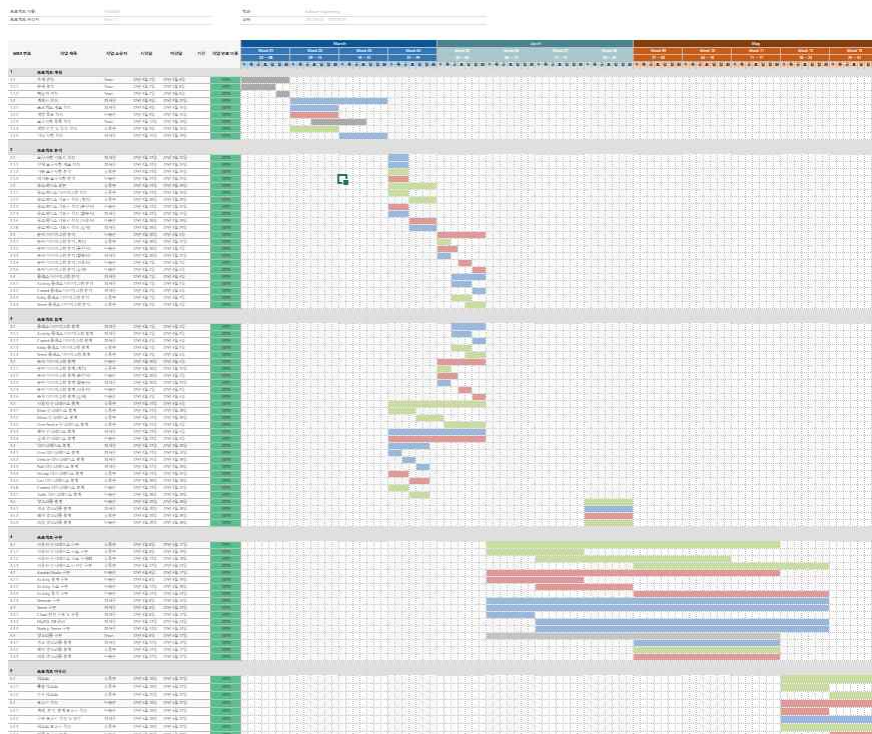
개발 계획의 업무인 계획 및 분석, 설계, 구현 및 시험, 유지보수 업무는 팀원 모두가 참여하도록 한다. 위의 업무 분담은 앞으로의 프로젝트 개발로서의 업무 분담으로 하도록 한다.

팀원 간의 의사소통이 원활할 수 있는 조직 체계를 가질 수 있는 구조를 채택하며, 팀장과 팀원 간의 인화 단결이 될 수 있는 구조가 될 수 있는 아래의 구조를 선택한다.



## 4.3 개발 일정

1 프로젝트 계획				
1.1	주제 선정	Team	22년 3월 2일	22년 3월 8일
1.2	계획서 작성	최재원	22년 3월 9일	22년 3월 22일
2 프로젝트 분석				
2.1	요구사항 기술서 작성	최재원	22년 3월 23일	22년 3월 25일
2.2	유스케이스 실현	김동현	22년 3월 23일	22년 3월 29일
2.3	순차 다이어그램 분석	이승찬	22년 3월 30일	22년 4월 5일
2.4	클래스 다이어그램 분석	최재원	22년 4월 1일	22년 4월 5일
3 프로젝트 설계				
3.1	클래스 다이어그램 설계	최재원	22년 4월 1일	22년 4월 5일
3.2	순차 다이어그램 설계	이승찬	22년 3월 30일	22년 4월 5일
3.3	사용자 인터페이스 설계	김동현	22년 3월 23일	22년 4월 5일
3.4	데이터베이스 설계	최재원	22년 3월 23일	22년 3월 28일
3.5	알고리즘 설계	이승찬	22년 4월 20일	22년 4월 26일
4 프로젝트 구현				
4.1	사용자 인터페이스 구현	김동현	22년 4월 6일	22년 5월 17일
4.2	Android Studio 구현	이승찬	22년 4월 6일	22년 5월 17일
4.3	Server 구현	최재원	22년 4월 6일	22년 5월 24일
4.4	알고리즘 구현	Team	22년 4월 6일	22년 5월 17일
5 프로젝트 마무리				
5.1	테스트	김동현	22년 5월 18일	22년 5월 31일
5.2	보고서 작성	이승찬	22년 5월 18일	22년 5월 31일



## 5. 기타 사항

### 5.1 프로그래밍 언어 및 도구

#### 5.1.1 IDE Tool

- 1) IDE : Android Studio
- 2) SDK Platform : Android API 31
- 3) Server IDE : Visual Studio Code

#### 5.1.2 Language

- 1) Programming Language : Java, Javascript
- 2) Markup Language : XML

#### 5.1.3 CASE Tool

- 1) draw.io
- 2) StarUML

#### 5.1.4 Database System

- 1) DBMS : MySQL
- 2) Connection : HTTP(Socket.io, Retrofit2)

#### 5.1.5 Server

- 1) Cloud computing service : AWS EC2 Ubuntu
- 2) Server : Node.js (Express.js)

# 프로젝트 분석

# 1. 요구사항 기술서

## 1.1 개요

### 1.1.1 정의

본 요구사항 기술서는 개발한 시스템을 정의한 시스템 정의서와 사용자의 애플리케이션 흐름을 파악한 것을 기반으로 작성되었다. 개발과정에서 구현해야 하는 요구 기능을 정의하기 위한 목적으로 작성되었으며, 추후 테스트 과정에서 진행하게 될 통합/인수 테스트에 대한 기초적인 자료로 활용될 것이다.

### 1.1.2 기술서

‘TANNAE’ 서비스를 사용하기 위해서는 이용자들은 자신의 계정을 하나 가지고 있어야 한다. 이용자는 ID/PW를 통해 로그인 함으로써 ‘TANNAE’ 서비스(계정, 운전자, 탑승자, 사용자, 결제 서비스)를 사용할 수 있다.

이용자는 최초 사용 시에 자신의 계정을 생성할 수 있어야 하며 로그인 시 ID/PW가 기억나지 않는다면 내부 기능을 통해 이를 찾을 수 있어야 한다. 자신의 개인 정보를 바탕으로 자신이 명시된 이용자임을 인증할 수 있으며 이를 통해 계정 정보를 확인할 수 있다. 또한 이용자는 인증을 거친 후에 자신의 계정 정보를 임의로 변경할 수 있으며 서비스를 계속해서 사용할 의사가 없을 때는 회원 탈퇴를 할 수 있다. 이용자의 개인 정보 보호를 위하여 개인 정보는 데이터베이스 상에서 즉시 삭제된다.

서비스를 제공할 때는 서비스 요청을 확인할 수 있으며 서비스를 제공하지 않는 시간대에는 운행 여부를 변경하여 잘못된 요청이 들어오지 않도록 제어할 수 있다.

일반적인 이용자는 탑승자로 서비스를 이용한다. 탑승자는 출발지/목적지/요청사항 등을 입력하여 서비스 이용할 수 있다. 결제는 포인트로 이루어지며 자동 결제를 기본으로 한다.

원활한 서비스 이용을 위하여 이용자는 기타 서비스를 사용할 수 있다. 차량에 물건을 두고 내려서 분실물을 찾고자 하는 기능을 사용할 수 있으며 문의 사항을 통해 서비스에 대한 질의 사항을 등록할 수 있다. 또한 자신의 이용 기록을 조회할 수 있지만 용량 상의 이유로 일정 기간 이후에는 이용 내용이 삭제된다.

이용자는 각자 운전자와 탑승자에 대한 평가가 가능하며 서비스 품질을 개선할 수 있으며 결제 내용에 관해 확인이 가능하다. 결제는 포인트로 진행되며 포인트는 충전 방식으로 이루어진다.

이 서비스는 모바일 애플리케이션으로 개발된다. 데이터의 일관성과 무결성이 유지되고, 보안을 유지하기 위해 데이터베이스를 사용한다. 또한 여러 사용자가 동시에 사용하기 때문에 서버를 통해 서비스를 제공한다. AWS를 통해 서버를 개발하며 데이터베이스는 MySQL을 사용한다.



## 1.2 기능 요구사항 (Functional Requirement)

### 1.2.1 계정 관련 기능

1) REQ-F-01 : 회원가입

- (1) 사용자는 로그인 화면에서 회원가입을 눌러 회원가입 화면으로 이동
- (2) 계정 및 사용자 정보를 입력하여 계정 생성 가능
- (3) 가입 완료 시 로그인 화면으로 전환

2) REQ-F-02 : 로그인

- (1) 어플을 실행하면 로그인 화면으로 전환
- (2) 로그인 화면에서 ID/PW를 입력하면 로그인되어 홈 화면으로 이동

3) REQ-F-03 : 계정 정보 찾기

- (1) 로그인 화면에서 계정 정보 찾기를 통하여 본인인증 화면으로 전환
- (2) 사용자 정보 등을 입력하고 본인인증 API를 통해 인증

4) REQ-F-04 : 계정 정보 수정

- (1) 마이페이지에서 정보 수정을 선택하여 정보 수정 페이지로 이동
- (2) 수정하고자 하는 데이터를 수정하여 저장

5) REQ-F-05 : 회원 탈퇴

- (1) 마이페이지에서 회원 탈퇴를 선택하여 회원 탈퇴 가능
- (2) 회원 탈퇴 선택 시 재확인 알림창이 출력되며 확인 및 취소 선택
- (3A) 재확인 알림창에서 확인을 선택하면 알림창이 종료되며 회원 탈퇴 진행
- (3B) 재확인 알림창에서 취소를 선택하면 알림창이 종료되며 취소 메시지 출력

### 1.2.2 운전자 서비스

1) REQ-F-06 : 서비스 요청 확인

- (1) 배차가 되면 지도에 경로 출력

2) REQ-F-07 : 운행 여부 변경

- (1) 운전자는 Navigation 화면에서 운전 여부 변경 가능

### 1.2.3 탑승자 서비스 기능

- 1) REQ-F\_08 : 배차 요청
  - (1) 탑승자는 배차를 선택하여 배차 페이지로 이동
  - (2) 출발지와 목적지를 선택하여 차량 서비스 이용

### 1.2.4 사용자 서비스 기능

- 1) REQ-F-09 : 분실물 찾기
  - (1) 메뉴 페이지에서 분실물 찾기를 선택하여 분실물 페이지로 이동
  - (2) 분실물 리스트를 확인하여 분실물 확인
- 2) REQ-F-10 : QnA
  - (1) 메뉴 페이지에서 QnA를 선택하여 QnA 페이지로 이동
  - (2) 사용자는 질문사항을 작성하고 관리자는 질문에 대한 답변을 작성
- 3) REQ-F-11 : FAQ
  - (1) 메뉴 페이지에서 FAQ를 선택하여 FAQ 페이지로 이동
- 4) REQ-F-12 : 이용 기록 조회
  - (1) 메뉴 페이지에서 Hlstory를 선택하여 이용 기록 페이지로 이동
  - (2) 사용자의 서비스 이용 기록 확인 가능

### 1.2.5 결제 서비스

- 1) REQ-F-13 : 운전 종료 및 평가 (운전자)
  - (1) 운전자는 운행 종료를 선택하여 사용자들에게 결제 영수증 전송
- 2) REQ-F-14 : 결제 확인 및 평가 (탑승자)
  - (1) 탑승자는 사용한 서비스에 대한 결제 영수증을 확인
  - (2) 평가 페이지로 이동하여 운전자와 동승자들을 평가하여 쾌적한 서비스 환경 조성
- 3) REQ-F-15 : 포인트 충전
  - (1) 메뉴 페이지에서 포인트를 선택하여 포인트 관리 페이지로 이동
  - (2) 포인트 조회와 충전 가능
  - (3) 충전을 선택하면 충전 페이지로 이동하여 충전량을 입력하고 결제하여 충전 완료

## 1.3 비기능 요구사항 (Non-Functional Requirement)

### 1.3.1 REQ-NF-01 : Client/Server 애플리케이션

- 1) 서버와의 연동을 위해 Client/Server application으로 개발
- 2) 개발 기간과 비용을 고려하여 AWS 클라우드 서비스를 이용

### 1.3.2 REQ-NF-02 : 사용자 친숙성

- 1) Mobile application이 낯선 사용자도 쉽게 사용할 수 있도록 직관적인 UI 구현

### 1.3.3 REQ-NF-03 : 시스템 확장성

- 1) 실제 환경에서의 사용을 위해 기능 추가가 가능
- 2) 지속적인 서비스 제공을 위해 유지 보수가 용이

### 1.3.4 REQ-NF-04 : 시스템 연결성

- 1) 효율적인 배차를 위해 차량들의 위치를 알고 있어야 하며 차량의 위치를 DB로 관리
- 2) 차량의 위치 변화를 추적하기 위해 DB를 지속해서 업데이트

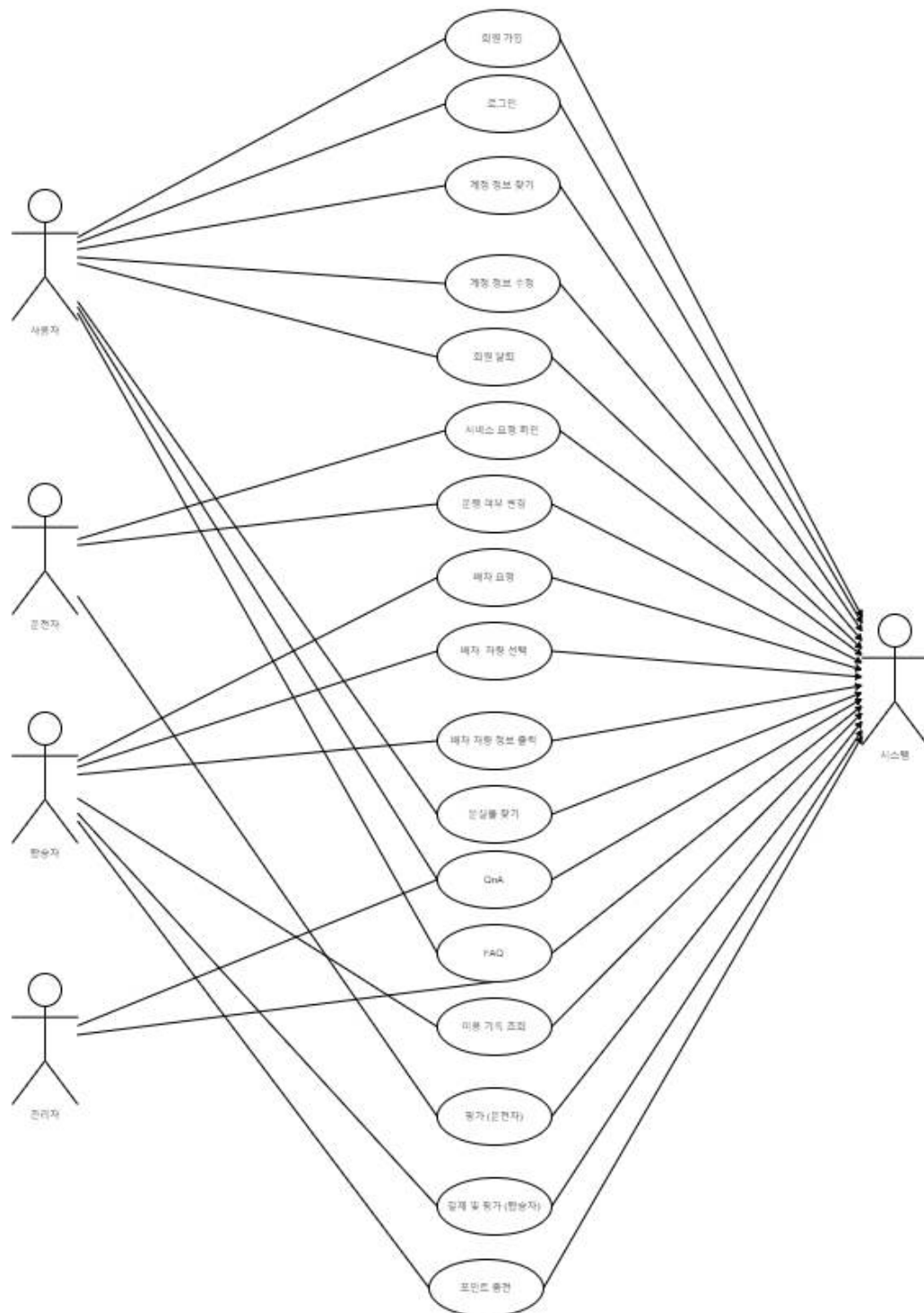
### 1.3.5 REQ-NF-05 : 서비스 응답속도

- 1) 사용자의 단일 동작에 대한 서비스의 응답속도는 1초 이내로 제한

### 1.3.6 REQ-NF-06 : 보안성

- 1) 사용자들의 계정 정보와 개인 정보는 타인에 노출 금지
- 2) 개인정보 보호법에 따라 개인 정보를 보호하도록 설계

## 2. Use case diagram



### 3. Use case 기술서

#### 3.1 계정 유스케이스 기술서

##### 3.1.1 회원가입 유스케이스 기술서

유스케이스 ID	usecase_account_01		
유스케이스 명	회원가입		
작 성 자	김동현	마지막 수정자	
작성 일시	22.03.26	마지막 수정 일시	
액터	사용자		
설명	TANNAE 서비스를 이용할 수 있는 회원 권한을 얻기 위해 회원가입		
트리거			
사전조건			
사후조건			
기본흐름	<ul style="list-style-type: none"> <li>- 사용자는 로그인 화면에서 회원가입 버튼을 눌러 회원가입 요청</li> <li>- 시스템은 사용자에게 회원가입 화면 출력</li> <li>- 사용자는 계정 정보(ID, PW), 사용자 정보(개인정보)를 입력</li> <li>- 입력된 사용자의 ID가 중복되었는지 검사 (E-1)</li> <li>- 본인인증을 통해 등록된 사용자인지 확인 (E-2)</li> <li>- 입력된 정보가 유효하다면 User DB에 신규 데이터를 삽입</li> <li>- 회원 가입이 완료되었다면 로그인 화면으로 전환</li> </ul>		
대체흐름			
예외흐름	E-1 : 중복 아이디일 경우 아이디 재입력 요청 E-2 : 이미 가입된 사용자일 경우 가입 거부		
포함			
우선순위	하		
사용빈도	하		
비즈니스 규칙			
특수요구사항	PW는 최소 8글자 이상을 입력해야 User DB에 데이터를 삽입 가능		
가정			
노트 및 이슈			

### 3.1.2 로그인 유스케이스 기술서

유스케이스 ID	usecase_account_02		
유스케이스 명	로그인		
작 성 자	김동현	마지막 수정자	
작성 일시	22.03.26	마지막 수정 일시	
액터	사용자		
설명	사용자의 ID, PW를 입력하여 TANNAE 서비스 이용		
트리거			
사전조건	회원 권한 보유		
사후조건			
기본흐름	<ul style="list-style-type: none"> <li>- 사용자는 로그인 화면에 ID, PW 입력</li> <li>- 입력된 회원을 User DB에서 검색</li> <li>- 로그인한 회원이 운전자라면 운행 기능 활성화</li> <li>- 로그인한 회원이 관리자이면 관리자 기능 활성화</li> <li>- 입력된 데이터가 무효하면 오류 메시지 반환(E-1)</li> </ul>		
대체흐름			
예외흐름	E-1 : 무효한 데이터가 입력되면 재입력 요청		
포함			
우선순위	중		
사용빈도	중		
비즈니스 규칙			
특수요구사항	PW는 최소 8자리 이상 입력		
가정			
노트 및 이슈			

### 3.1.3 계정 정보 찾기 유스케이스 기술서

유스케이스 ID	usecase_account_03		
유스케이스 명	계정 정보 찾기		
작 성 자	김동현	마지막 수정자	
작성 일시	22.03.26	마지막 수정 일시	
액터	사용자		
설명	사용자의 계정 정보 검색		
트리거			
사전조건			
사후조건			
기본흐름	<ul style="list-style-type: none"> <li>- 로그인 화면에서 계정 정보 찾기를 통해 정보 찾기 페이지 이동</li> <li>- 사용자 개인 정보를 입력</li> <li>- 입력된 개인 정보가 회원이면 본인 인증 화면 출력</li> <li>- 입력된 개인 정보가 비회원이면 오류 메시지 출력 (E-1)</li> <li>- 문자 메시지를 통해 PIN 번호를 수신하며 화면에 입력</li> <li>- 입력된 PIN 번호가 유효하면 계정 정보 출력</li> <li>- 입력된 PIN 번호가 무효한 데이터이면 오류 메시지 출력 (E-2)</li> </ul>		
대체흐름			
예외흐름	E-1 : 비회원이면 사용자에게 오류 메시지 출력하며 재입력 요청 E-1 : 올바른 PIN 번호가 아니면 오류 메시지 출력 및 재입력 요청		
포함			
우선순위	하		
사용빈도	하		
비즈니스 규칙			
특수요구사항			
가정			
노트 및 이슈			

### 3.1.4 계정 정보 수정 유스케이스 기술서

유스케이스 ID	usecase_account_04		
유스케이스 명	계정 정보 수정		
작 성 자	김동현	마지막 수정자	
작성 일시	22.03.26	마지막 수정 일시	
액터	사용자		
설명	사용자의 계정 정보 수정		
트리거			
사전조건	회원 권한 보유		
사후조건			
기본흐름	<ul style="list-style-type: none"> <li>- 사용자는 마이페이지에서 수정 버튼을 통해 수정 페이지 전환</li> <li>- 수정 페이지에서 새로운 데이터를 입력하여 User DB 업데이트</li> </ul>		
대체흐름			
예외흐름			
포함			
우선순위	하		
사용빈도	하		
비즈니스 규칙			
특수요구사항			
가정			
노트 및 이슈			



### 3.1.5 회원 탈퇴 유스케이스 기술서

유스케이스 ID	usecase_account_05		
유스케이스 명	회원 탈퇴		
작 성 자	김동현	마지막 수정자	
작성 일시	22.03.26	마지막 수정 일시	
액터	사용자		
설명	사용자의 계정 탈퇴		
트리거			
사전조건	회원 권한 보유		
사후조건			
기본흐름	<ul style="list-style-type: none"> <li>- 마이페이지에서 회원 탈퇴 버튼을 통해 회원 탈퇴 요청</li> <li>- 회원 탈퇴 재확인 알림창 출력</li> <li>- 확인을 통해 회원 탈퇴</li> <li>- User DB에서 해당 튜플 삭제</li> <li>- 취소 버튼 클릭 시 재확인 알림창 종료 및 회원 탈퇴 취소</li> </ul>		
대체흐름			
예외흐름			
포함			
우선순위	하		
사용빈도	하		
비즈니스 규칙			
특수요구사항			
가정			
노트 및 이슈			

## 3.2 운전자 서비스 유스케이스 기술서

### 3.2.1 서비스 요청 확인 유스케이스 기술서

유스케이스 ID	usecase_driver_01		
유스케이스 명	서비스 요청 확인		
작 성 자	이승찬	마지막 수정자	
작성 일시	22.03.26	마지막 수정 일시	
액터	운전자		
설명	사용자로부터 걸려온 배차 요청 내용 확인		
트리거	탑승자가 배차 요청 기능을 실행		
사전조건	회원(운전자) 권한 보유, 탑승자의 배차 요청		
사후조건			
기본흐름	<ul style="list-style-type: none"> <li>- 탑승자가 배차 요청 기능을 통해 해당 차량에게 배차 요청</li> <li>- 새롭게 수정된 경로를 배차된 운전자 화면에 출력</li> </ul>		
대체흐름			
예외흐름			
포함			
우선순위	상		
사용빈도	상		
비즈니스 규칙			
특수요구사항			
가정			
노트 및 이슈			

### 3.2.2 운행 여부 변경 유스케이스 기술서

유스케이스 ID	usecase_driver_02		
유스케이스 명	운행 여부 변경		
작 성 자	이승찬	마지막 수정자	
작성 일시	22.03.26	마지막 수정 일시	
액터	운전자		
설명	운전자의 운행 여부 상태 변경		
트리거			
사전조건	회원(운전자) 권한 보유		
사후조건			
기본흐름	<ul style="list-style-type: none"> <li>- Navigation 화면에서 switch 버튼을 통해 운행여부 변경</li> <li>- 시스템은 운행 여부를 변경</li> </ul>		
대체흐름			
예외흐름			
포함			
우선순위	중		
사용빈도	중		
비즈니스 규칙			
특수요구사항			
가정			
노트 및 이슈			

### 3.3 탑승자 서비스 유스케이스 기술서

#### 3.3.1 배차 요청 유스케이스 기술서

유스케이스 ID	usecase_passenger_01		
유스케이스 명	배차 요청		
작 성 자	최재원	마지막 수정자	
작성 일시	22.03.25	마지막 수정 일시	
액터	탑승자		
설명	교통 서비스 이용을 위해 배차 요청		
트리거			
사전조건	회원 권한 보유		
사후조건			
기본흐름	<ul style="list-style-type: none"> <li>- 배차 요청 선택</li> <li>- 출발지와 도착지를 선택</li> <li>- 동승 여부 선택</li> <li>- 배차 요청</li> <li>- 배차 가능한 차량을 탐색</li> <li>- 배차 가능 차량이 있을 경우에는 해당 차량으로 배차</li> <li>- 배차 가능한 차량이 없을 경우 서비스 이용 거부</li> </ul>		
대체흐름			
예외흐름			
포함			
우선순위	상		
사용빈도	상		
비즈니스 규칙			
특수요구사항			
가정			
노트 및 이슈			

## 3.4 사용자 서비스 유스케이스 기술서

### 3.4.1 분실물 유스케이스 기술서

유스케이스 ID	usecase_user_01		
유스케이스 명	분실물 찾기		
작 성 자	이승찬	마지막 수정자	
작성 일시	22.03.26	마지막 수정 일시	
액터	사용자		
설명	차량에 두고 내린 물건을 분실물 리스트에서 조회		
트리거			
사전조건	회원 권한 보유		
사후조건			
기본흐름	< 분실물 찾기 > - Menu 화면에서 '분실물 찾기'를 선택 - 분실물 리스트에서 분실물 확인		
대체흐름			
예외흐름			
포함			
우선순위	하		
사용빈도	중		
비즈니스 규칙			
특수요구사항			
가정			
노트 및 이슈			

### 3.4.2 QnA 유스케이스 기술서

유스케이스 ID	usecase_user_02		
유스케이스 명	QnA		
작 성 자	이승찬	마지막 수정자	
작성 일시	22.03.26	마지막 수정 일시	
액터	사용자		
설명			
트리거			
사전조건	회원 권한 보유		
사후조건			
기본흐름	< QnA 등록 > - Menu 화면에서 'QnA' 선택 - QnA 리스트 화면 출력 - 리스트 화면에서 기존 질의응답 조회 - 원하는 질의응답이 없을 경우, 질문 작성 - 질문 작성 화면 출력 - 질문 작성 화면에 세부 질문 사항 입력 및 등록 - Content DB에 Insert		
대체흐름	- 관리자의 경우 QnA 질문 세부사항에서 댓글을 통해 답변		
예외흐름			
포함			
우선순위	중		
사용빈도	중		
비즈니스 규칙			
특수요구사항			
가정			
노트 및 이슈			

### 3.4.3 FAQ 유스케이스 기술서

유스케이스 ID	usecase_user_03		
유스케이스 명	FAQ		
작 성 자	이승찬	마지막 수정자	
작성 일시	22.03.26	마지막 수정 일시	
액터	사용자		
설명	자주 묻는 질의응답을 조회		
트리거			
사전조건	회원 권한 보유		
사후조건			
기본흐름	< FAQ 조회 > - Menu 화면에서 'FAQ' 선택 - 시스템은 FAQ 리스트 화면을 출력		
대체흐름			
예외흐름			
포함			
우선순위	하		
사용빈도	하		
비즈니스 규칙			
특수요구사항			
가정			
노트 및 이슈			

### 3.4.4 이용 기록 조회 유스케이스 기술서

유스케이스 ID	usecase_user_04		
유스케이스 명	이용 기록 조회		
작 성 자	이승찬	마지막 수정자	
작성 일시	22.03.26	마지막 수정 일시	
액터	탑승자		
설명	탑승자의 차량 탑승 기록 조회		
트리거			
사전조건	회원 권한 보유		
사후조건			
기본흐름	<ul style="list-style-type: none"> <li>- Menu 화면에서 'History' 선택</li> <li>- 시스템은 History 리스트 출력</li> </ul>		
대체흐름			
예외흐름			
포함			
우선순위	하		
사용빈도	하		
비즈니스 규칙			
특수요구사항			
가정			
노트 및 이슈			



### 3.5 결제 서비스 유스케이스 기술서

#### 3.5.1 결제 및 평가 (운전자) 유스케이스 기술서

유스케이스 ID	usecase_pay_01		
유스케이스 명	결제 및 평가 (운전자)		
작 성 자	최재원	마지막 수정자	
작성 일시	22.03.25	마지막 수정 일시	
액터	운전자		
설명	결제 완료를 선택하고 서비스 평가		
트리거			
사전조건	회원 권한 및 운전자 권한 보유		
사후조건			
기본흐름	- 차량의 탑승자가 모두 하차 후 운행 종료 선택		
대체흐름			
예외흐름			
포함			
우선순위	하		
사용빈도	중		
비즈니스 규칙			
특수요구사항			
가정			
노트 및 이슈			

### 3.5.2 결제 및 평가 (탑승자) 유스케이스 기술서

유스케이스 ID	usecase_pay_02		
유스케이스 명	결제 및 평가 (탑승자)		
작 성 자	최재원	마지막 수정자	
작성 일시	22.03.25	마지막 수정 일시	
액터	탑승자		
설명	결제 완료 요청이 들어오면 포인트 차감 및 서비스 평가		
트리거			
사전조건	회원 권한 보유		
사후조건			
기본흐름	<ul style="list-style-type: none"> <li>- 탑승했던 차량의 결제 완료 요청</li> <li>- 충전했전 포인트에서 자동으로 차감</li> <li>- 결제 영수증 화면이 출력되며 확인</li> <li>- 서비스 평가 화면에서 운전자와 동승자 평가</li> </ul>		
대체흐름			
예외흐름	<ul style="list-style-type: none"> <li>- 포인트가 부족할 경우 충전을 요구</li> <li>- 미결제 포인트를 충전할 때까지 서비스 이용 불가</li> </ul>		
포함			
우선순위	중		
사용빈도	중		
비즈니스 규칙			
특수요구사항			
가정			
노트 및 이슈			

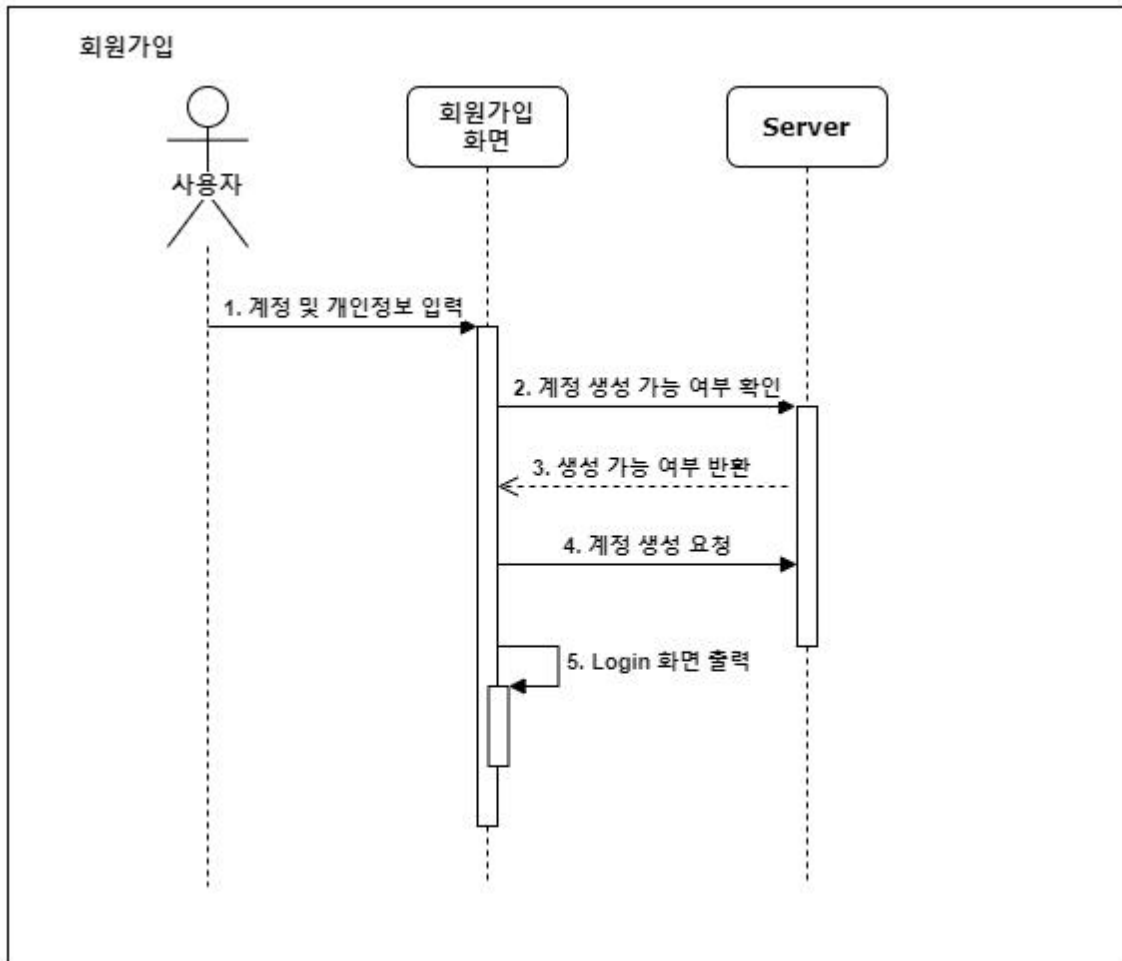
### 3.5.3 포인트 충전 유스케이스 기술서

유스케이스 ID	usecase_pay_03		
유스케이스 명	포인트 충전		
작 성 자	최재원	마지막 수정자	
작성 일시	22.03.25	마지막 수정 일시	
액터	사용자		
설명	서비스 이용을 위한 포인트 충전		
트리거			
사전조건	회원 권한 보유		
사후조건			
기본흐름	<ul style="list-style-type: none"> <li>- 메뉴 화면에서 포인트 충전 선택</li> <li>- 포인트 내역 확인 및 포인트 충전 선택</li> <li>- 충전량 선택 및 결제 서비스 요청</li> <li>- 결제를 통해 포인트 충전</li> </ul>		
대체흐름			
예외흐름			
포함			
우선순위	하		
사용빈도	하		
비즈니스 규칙			
특수요구사항			
가정			
노트 및 이슈			

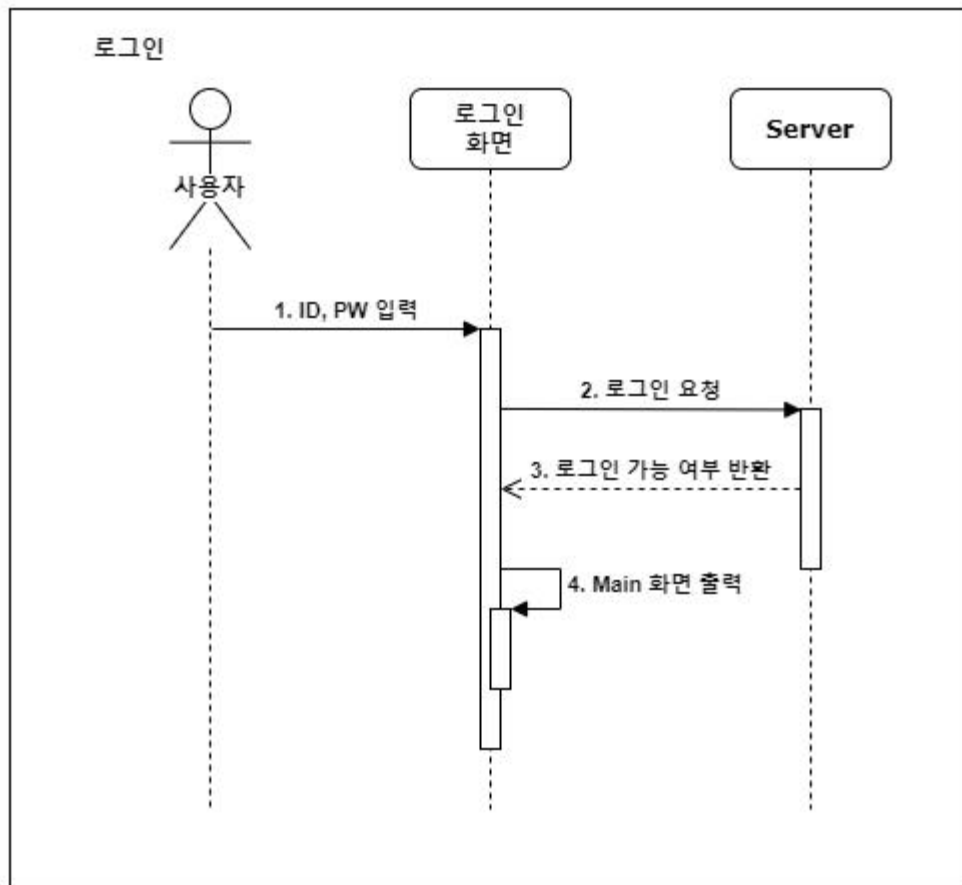
## 4. 순차 다이어그램 분석

### 4.1 계정 순차 다이어그램 분석

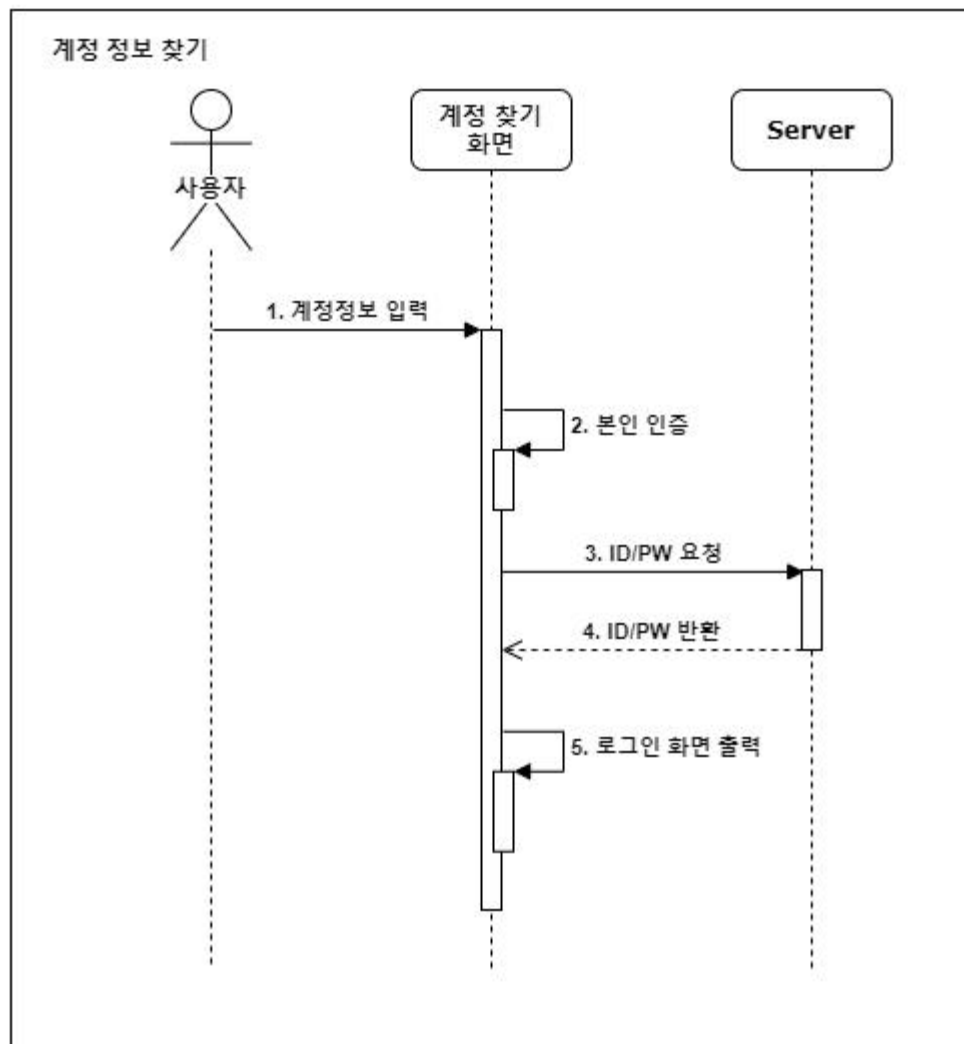
#### 4.1.1 회원가입 순차 다이어그램 분석



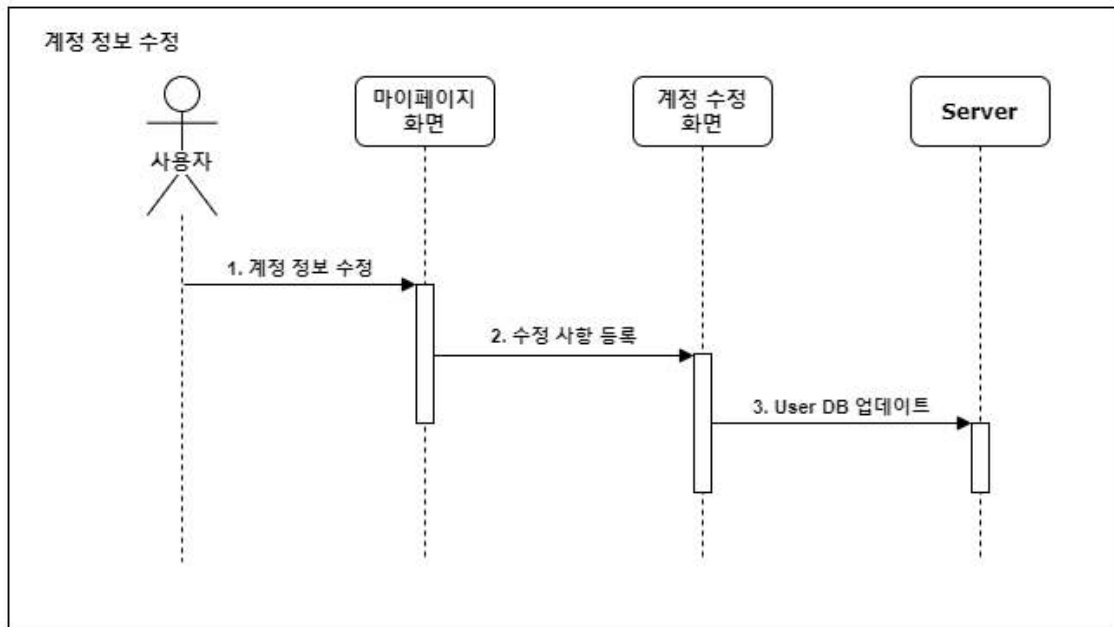
#### 4.1.2 로그인 순차 다이어그램 분석



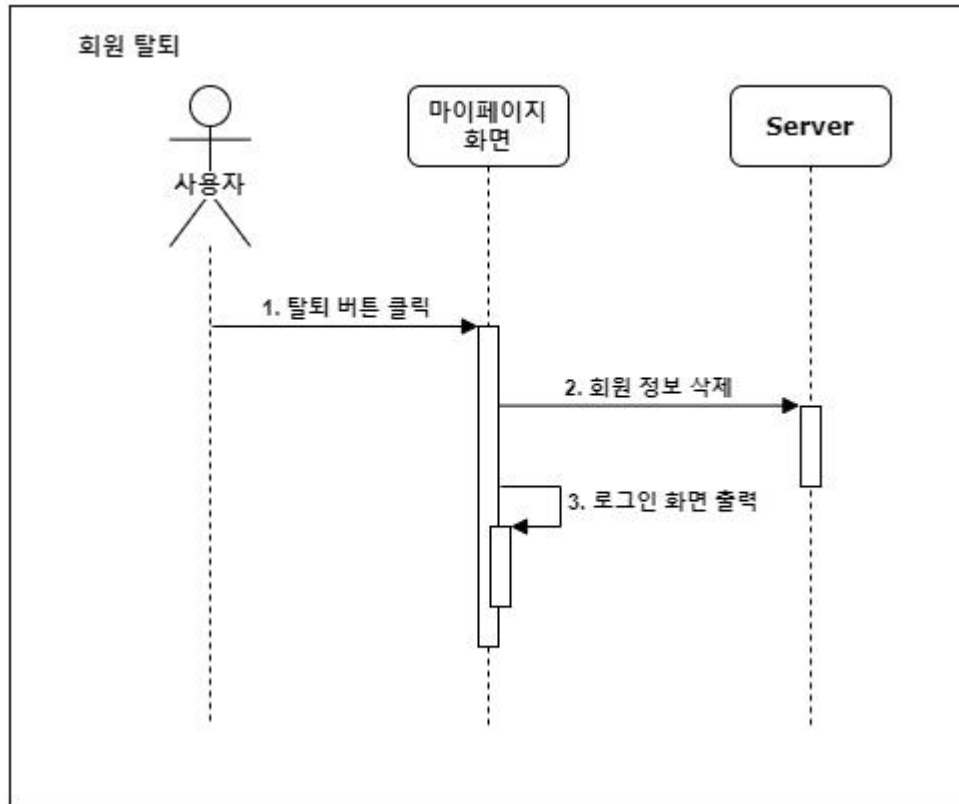
#### 4.1.3 계정 정보 찾기 순차 다이어그램 분석



#### 4.1.4 계정 정보 수정 순차 다이어그램 분석



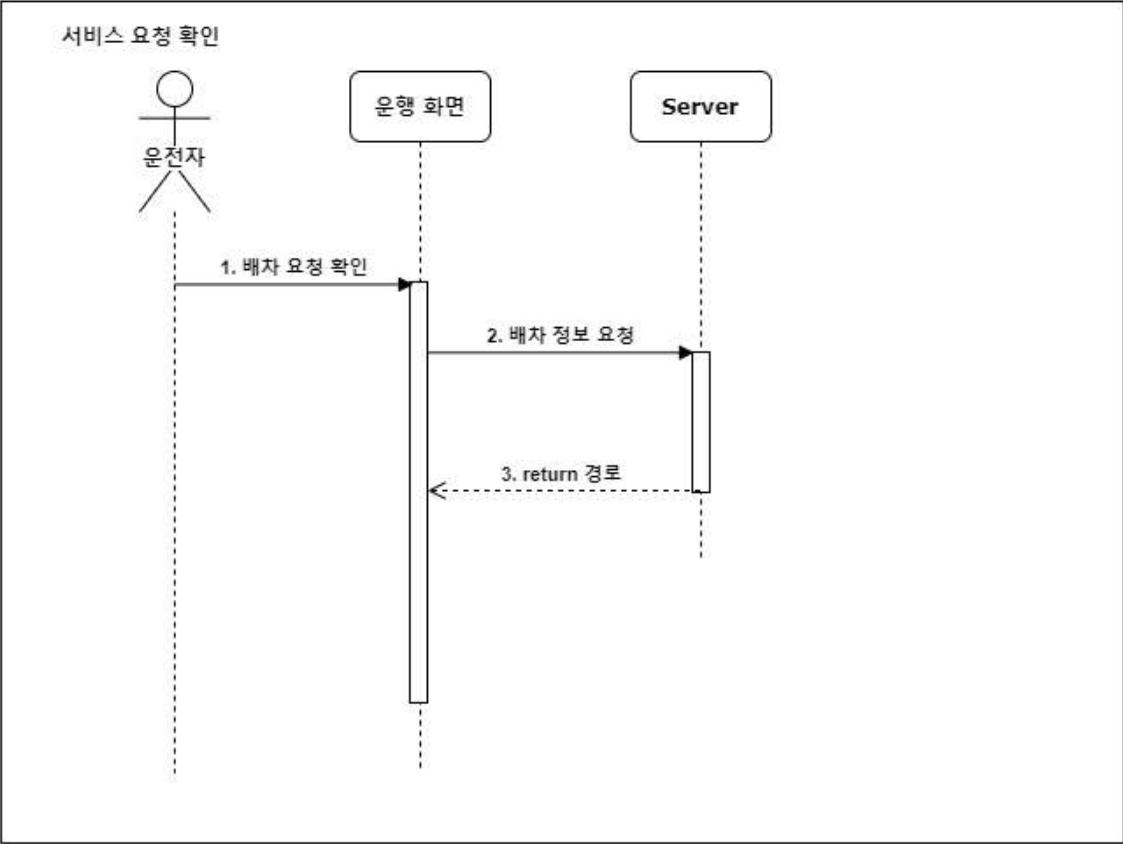
#### 4.1.5 회원 탈퇴 순차 다이어그램 분석



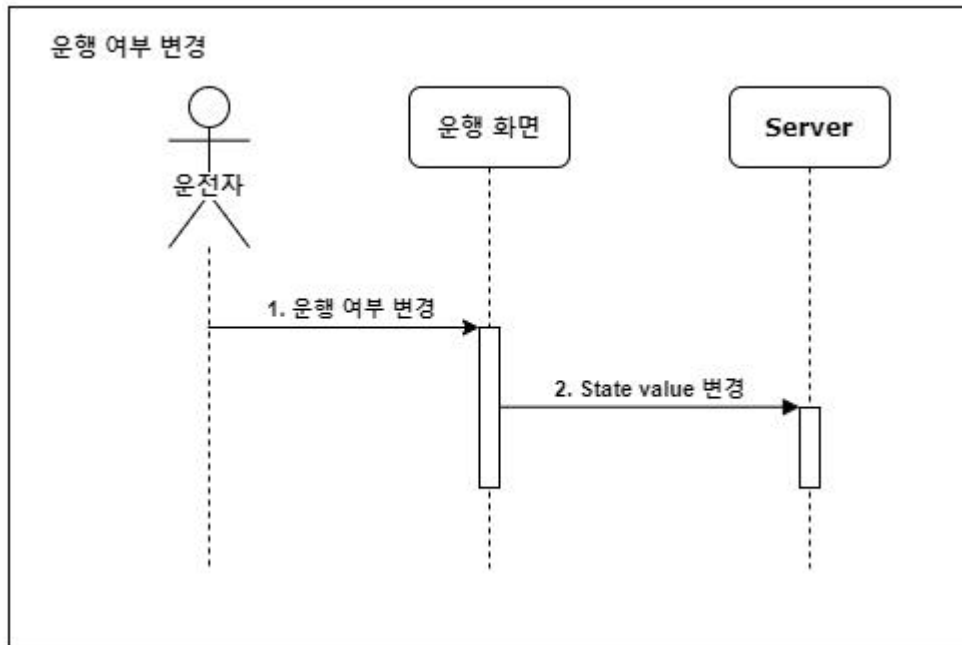


## 4.2 운전자 서비스 순차 다이어그램 분석

### 4.2.1 서비스 요청 확인 순차 다이어그램 분석

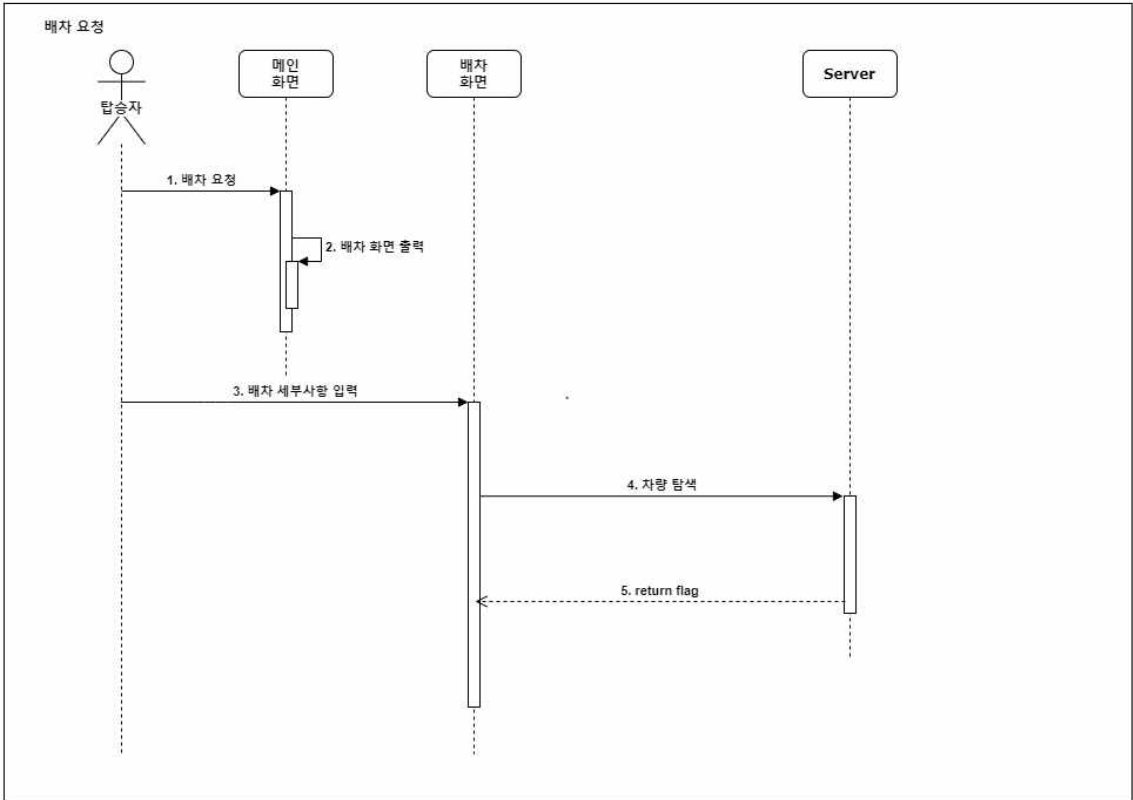


#### 4.2.2 운행 여부 변경 순차 다이어그램 분석



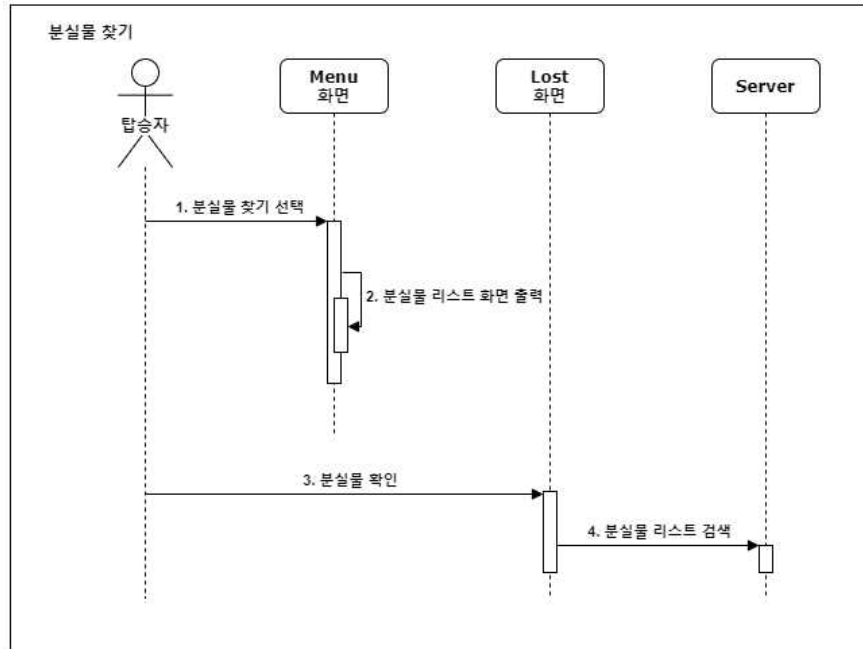
### 4.3 탑승자 서비스 순차 다이어그램 분석

#### 4.3.1 배차 요청 순차 다이어그램 분석

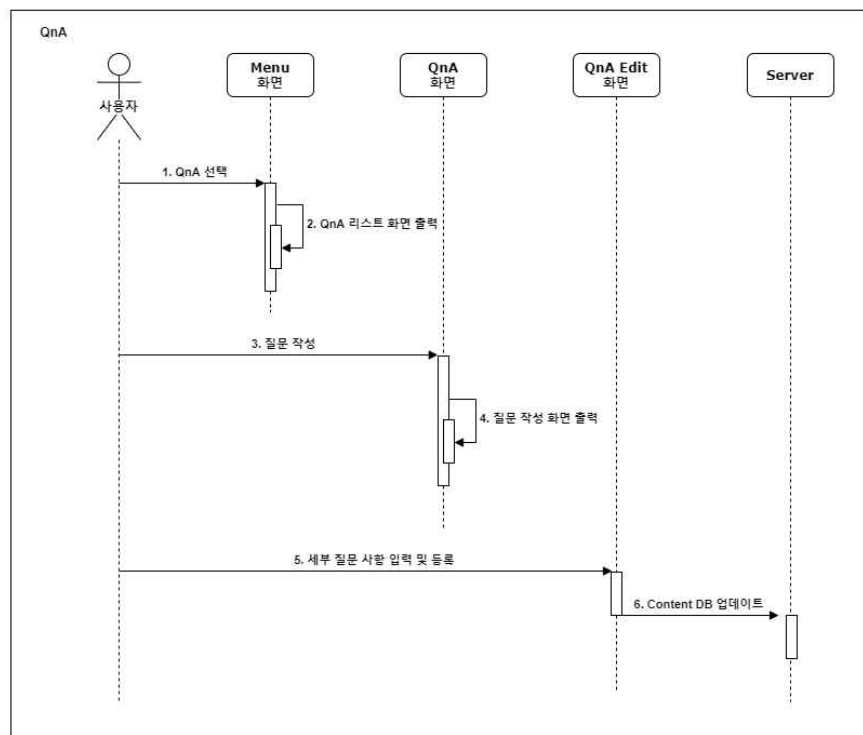


## 4.4 사용자 서비스 순차 다이어그램 분석

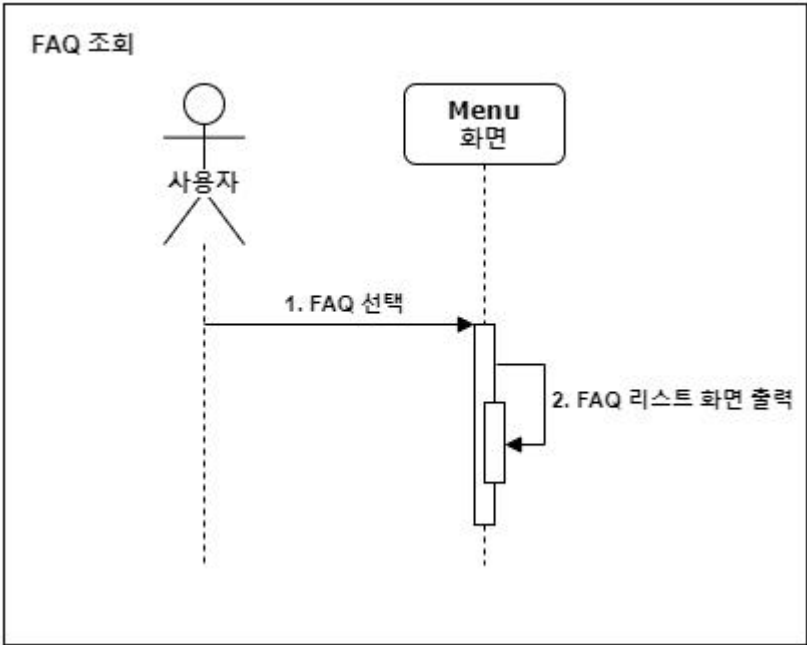
### 4.4.1 분실물 순차 다이어그램 분석



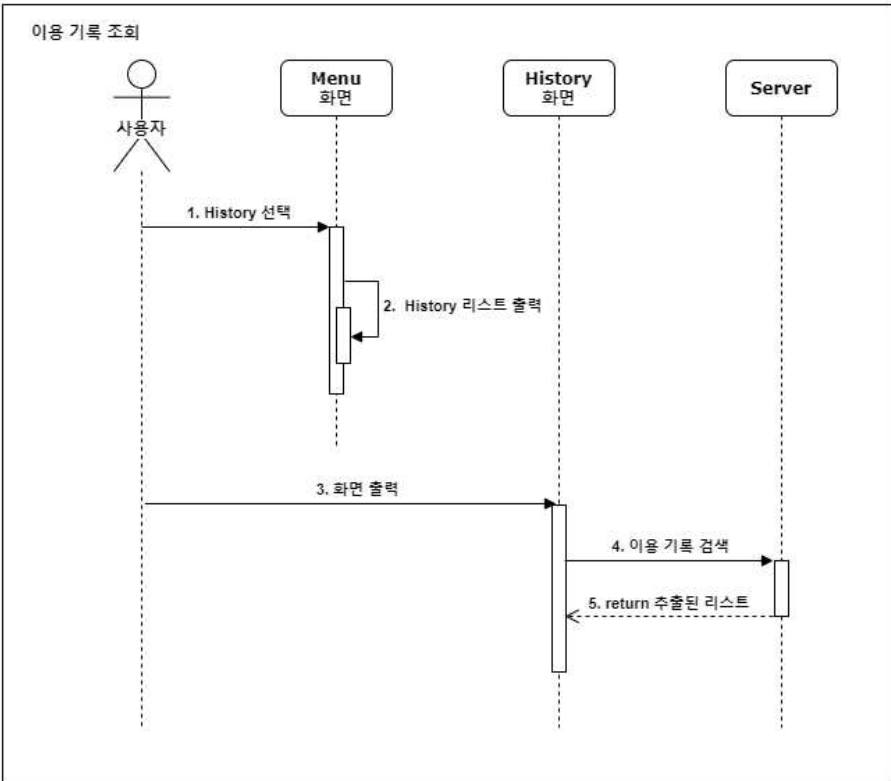
### 4.4.2 QnA 순차 다이어그램 분석



#### 4.4.3 FAQ 순차 다이어그램 분석

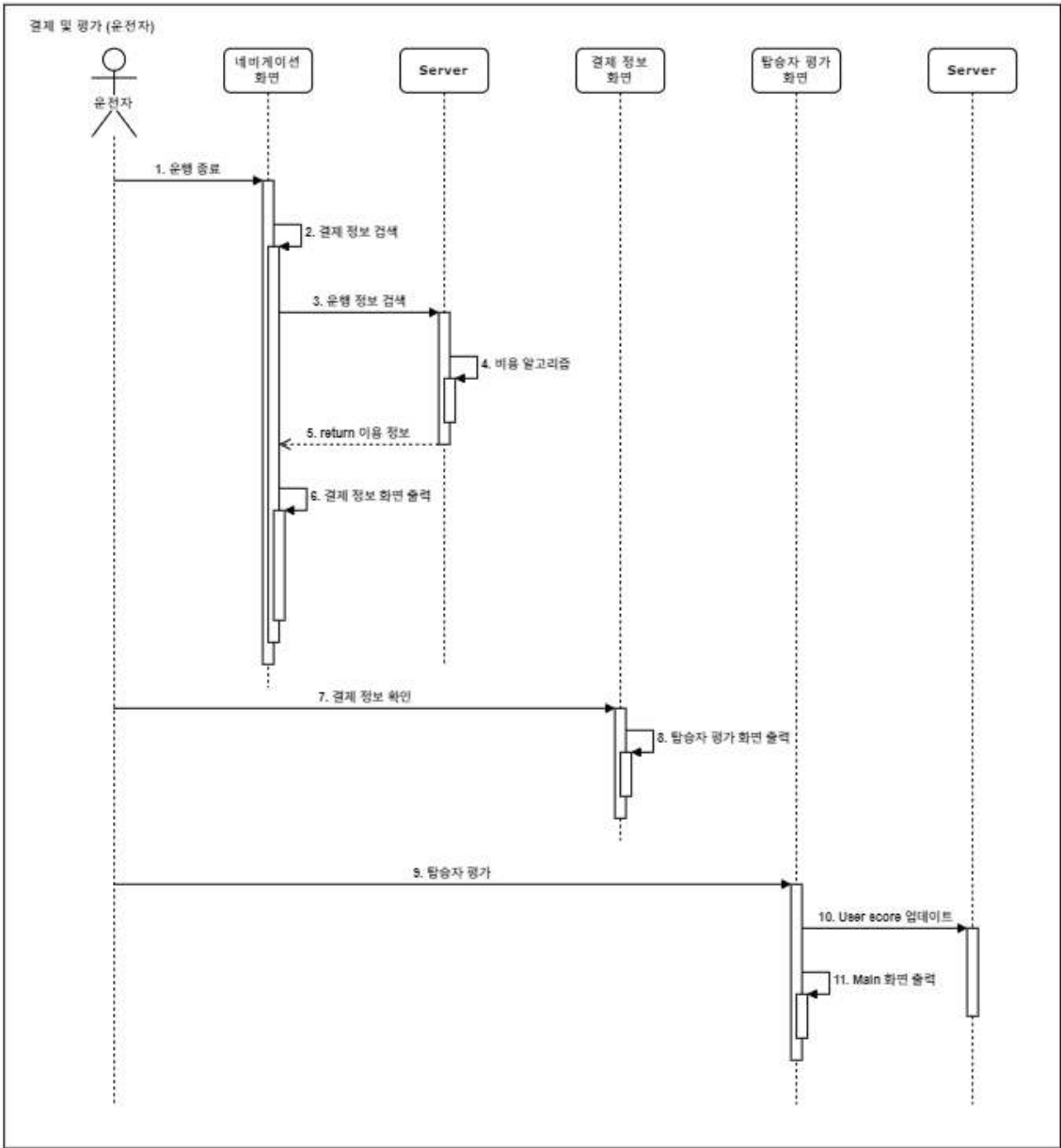


#### 4.4.4 이용 기록 조회 순차 다이어그램 분석

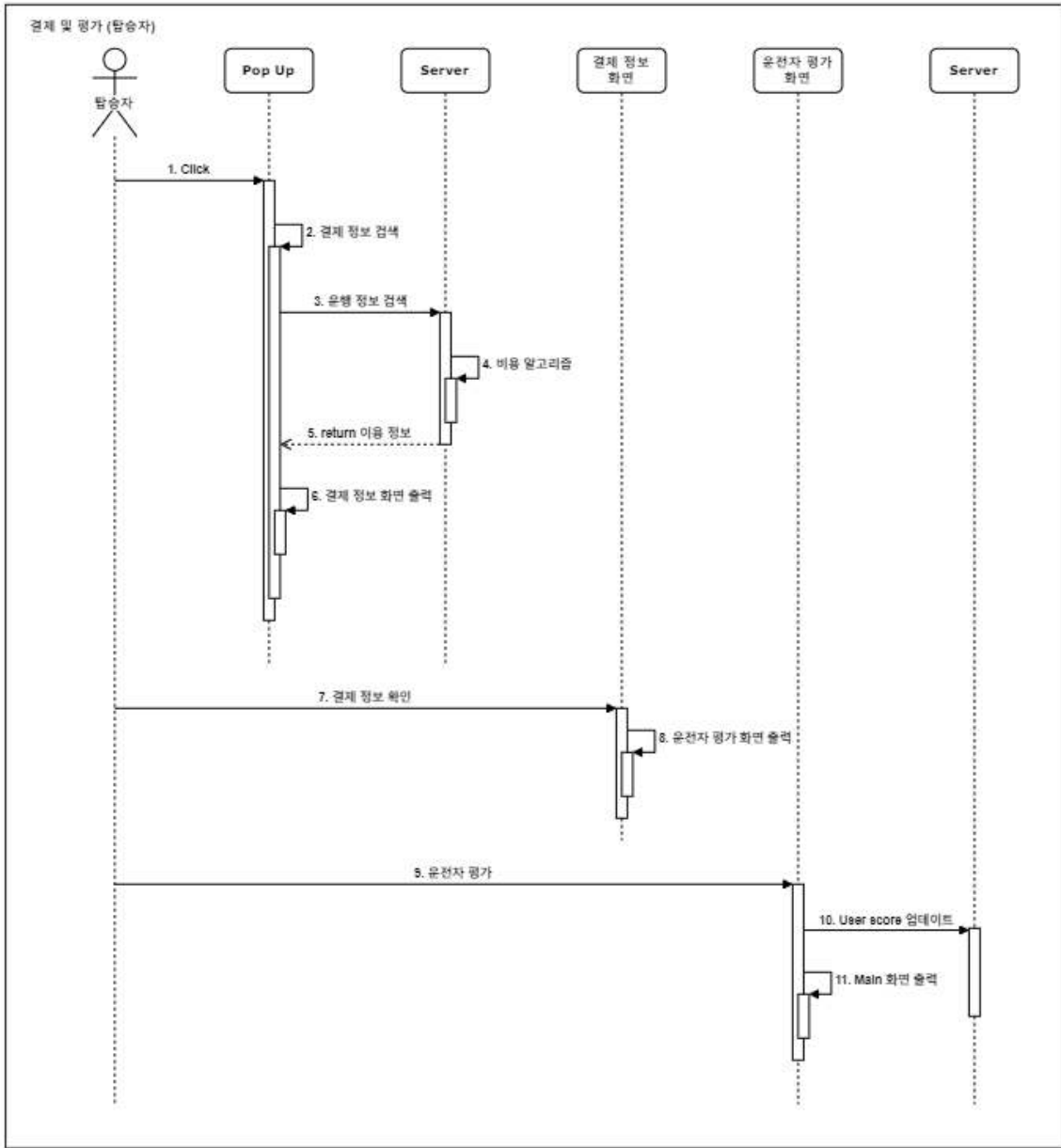


## 4.5 결제 서비스 순차 다이어그램 분석

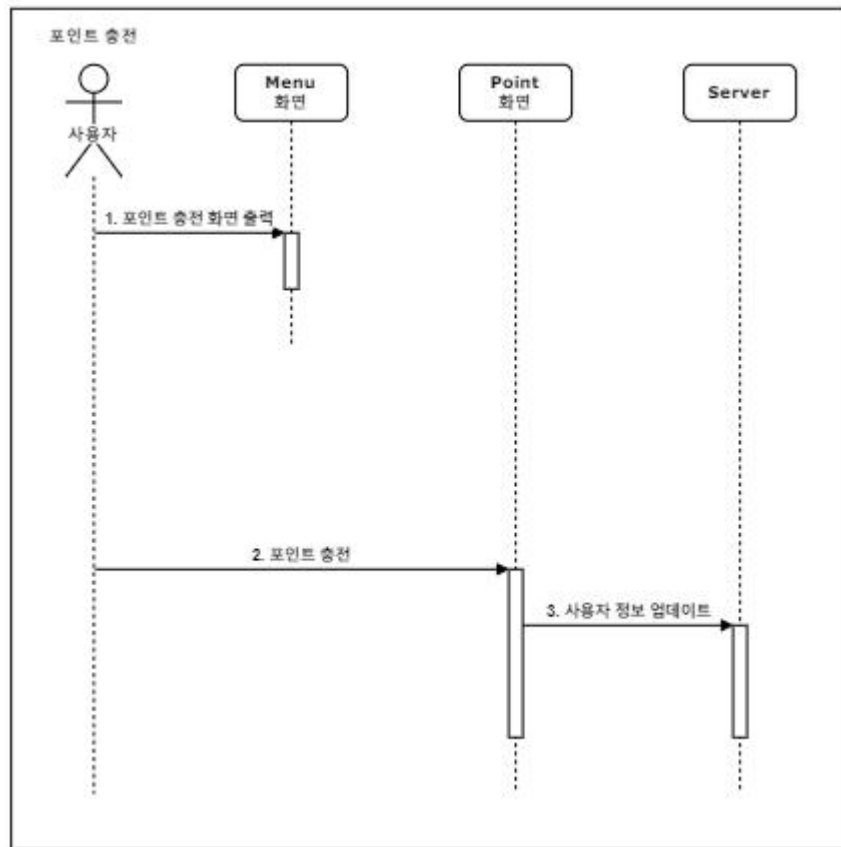
### 4.5.1 결제 및 평가 (운전자) 순차 다이어그램 분석



4.5.2 결제 및 평가 (탑승자) 순차 다이어그램 분석



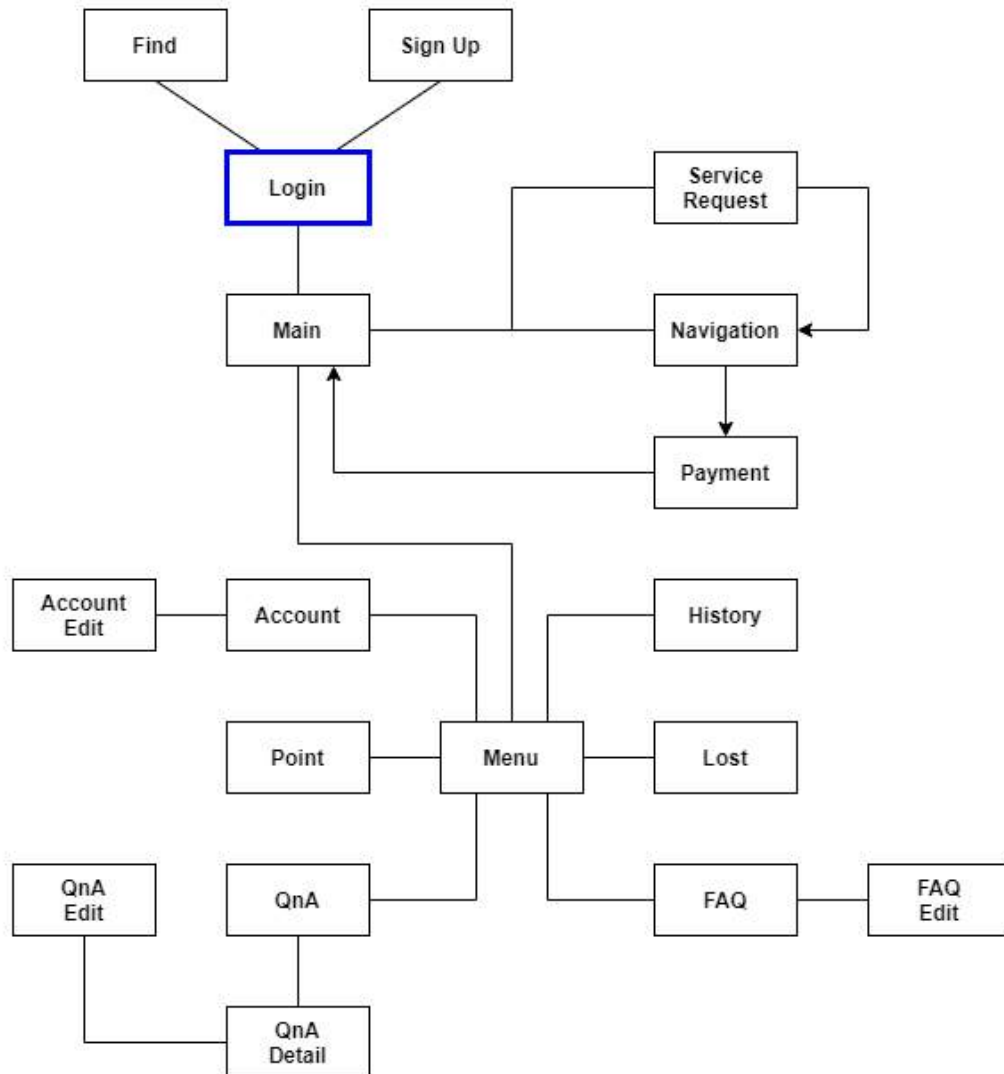
### 4.5.3 포인트 충전 순차 다이어그램 분석



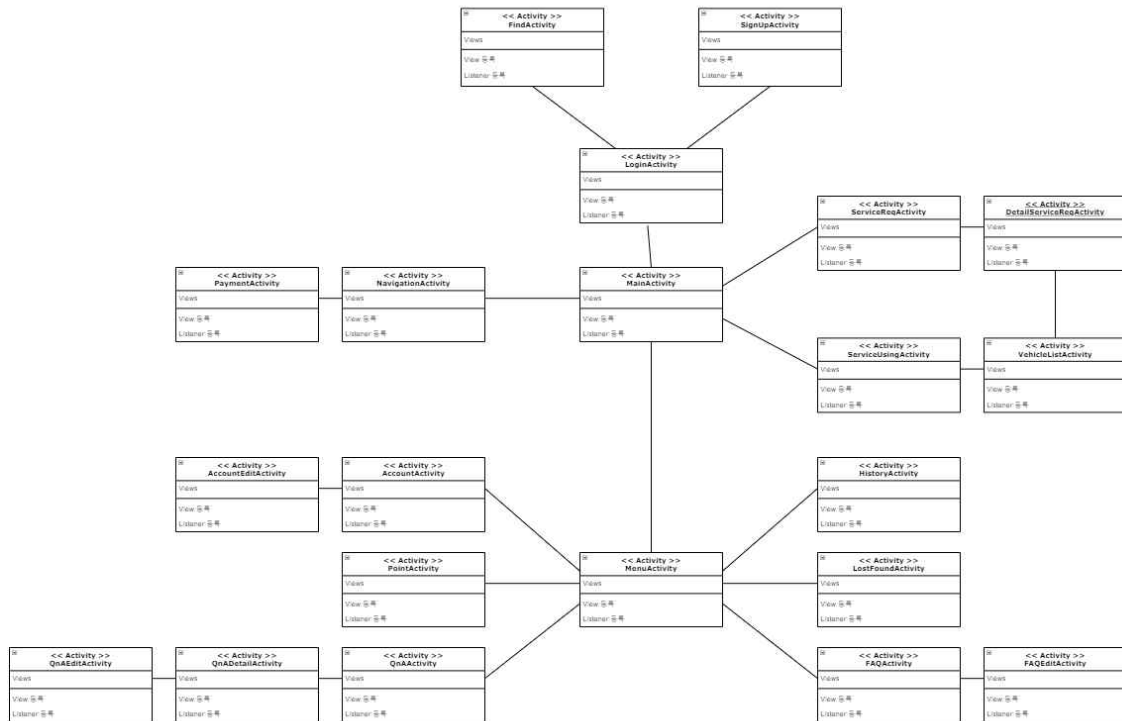


## 5. 클래스 다이어그램 분석

### 5.1 화면 흐름도 작성

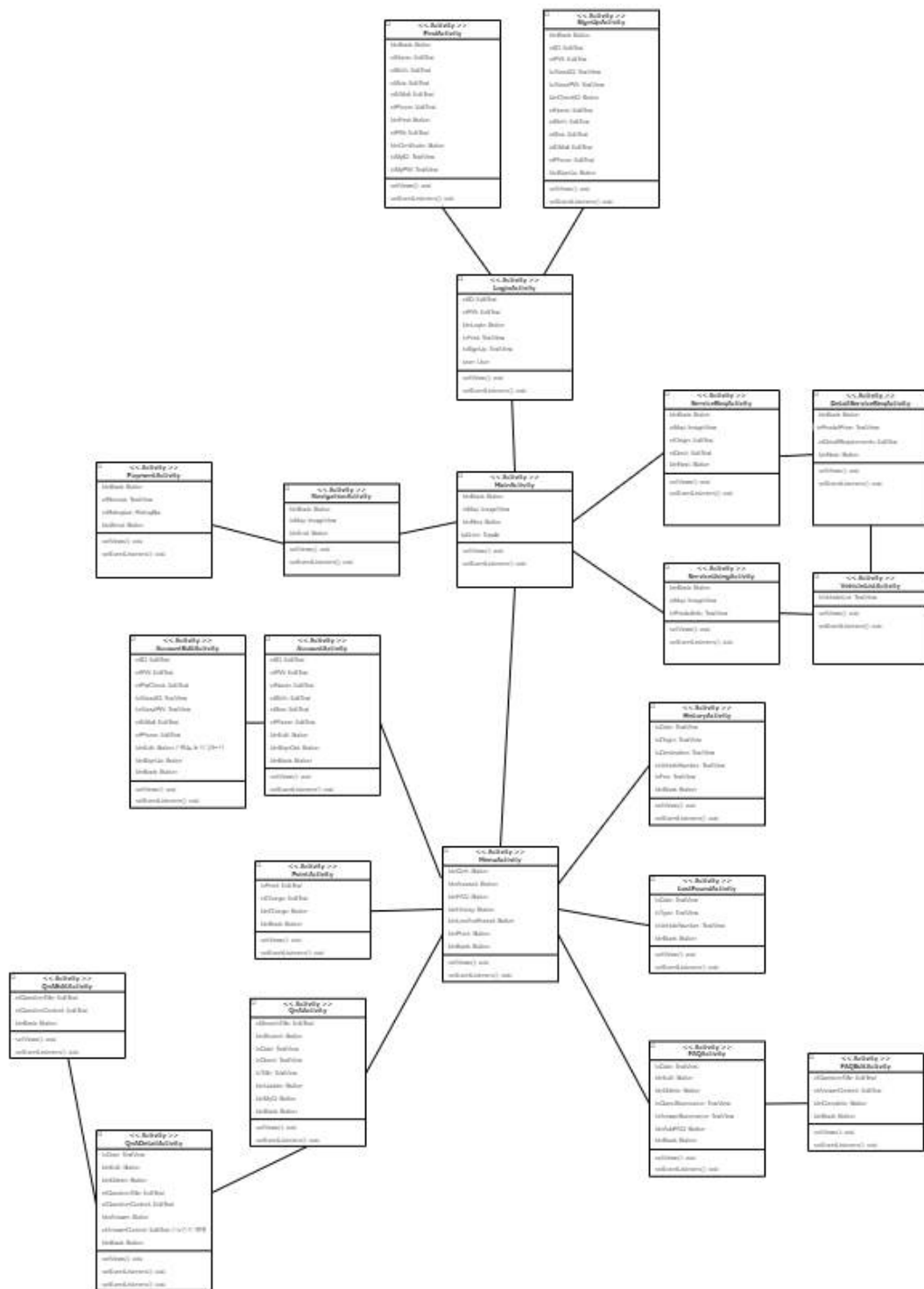


## 5.2 클래스 다이어그램



# 프로젝트 설계

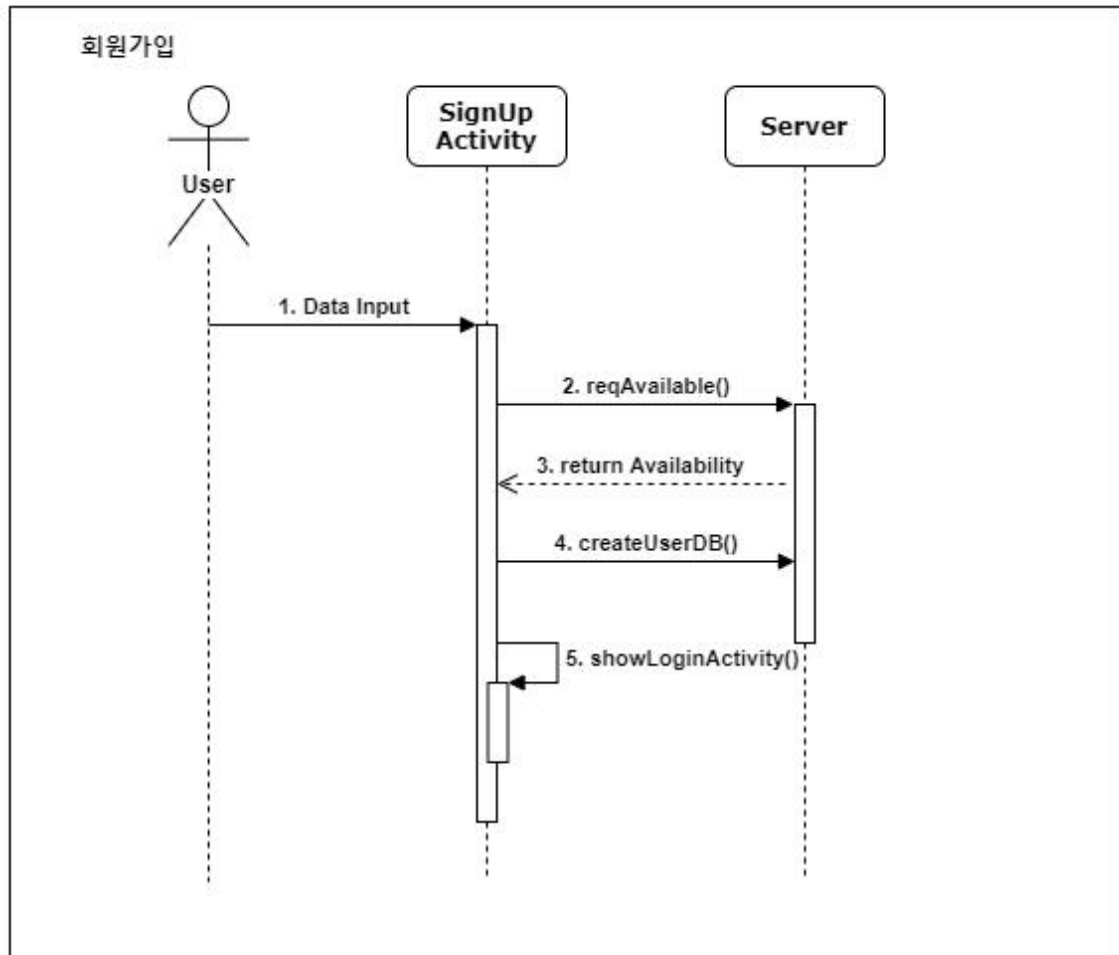
## 1. 클래스 다이어그램 설계



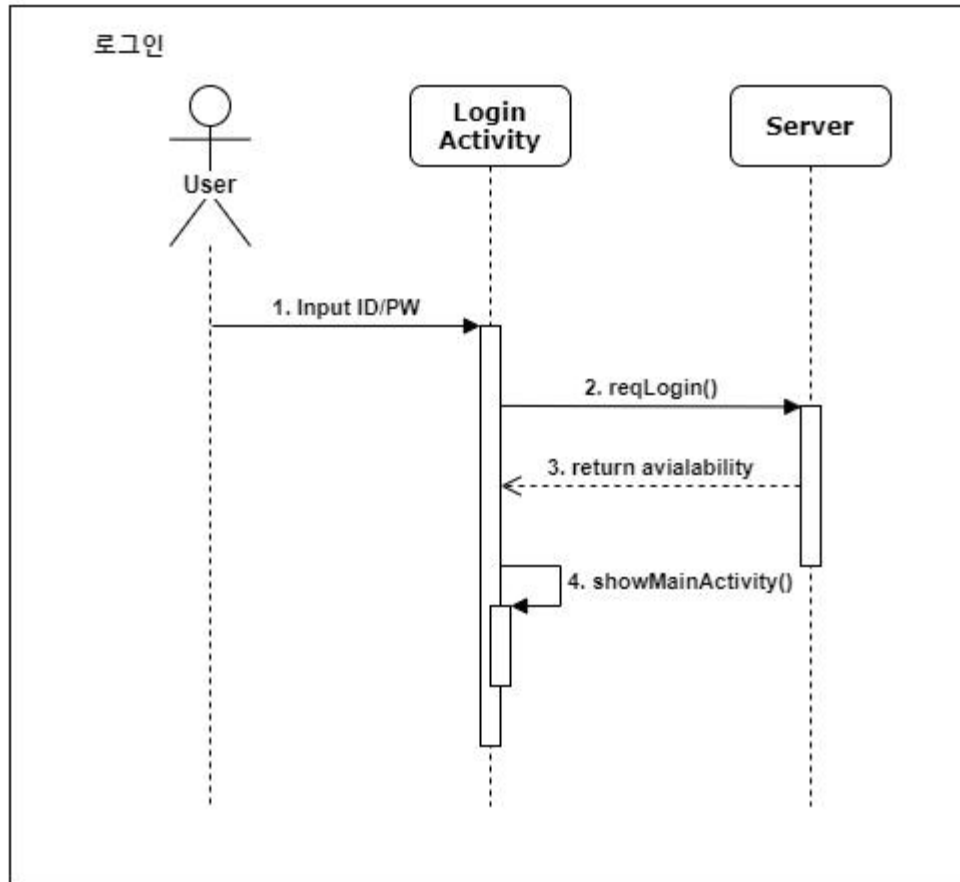
## 2. 순차 다이어그램 설계

### 2.1 계정 순차 다이어그램 설계

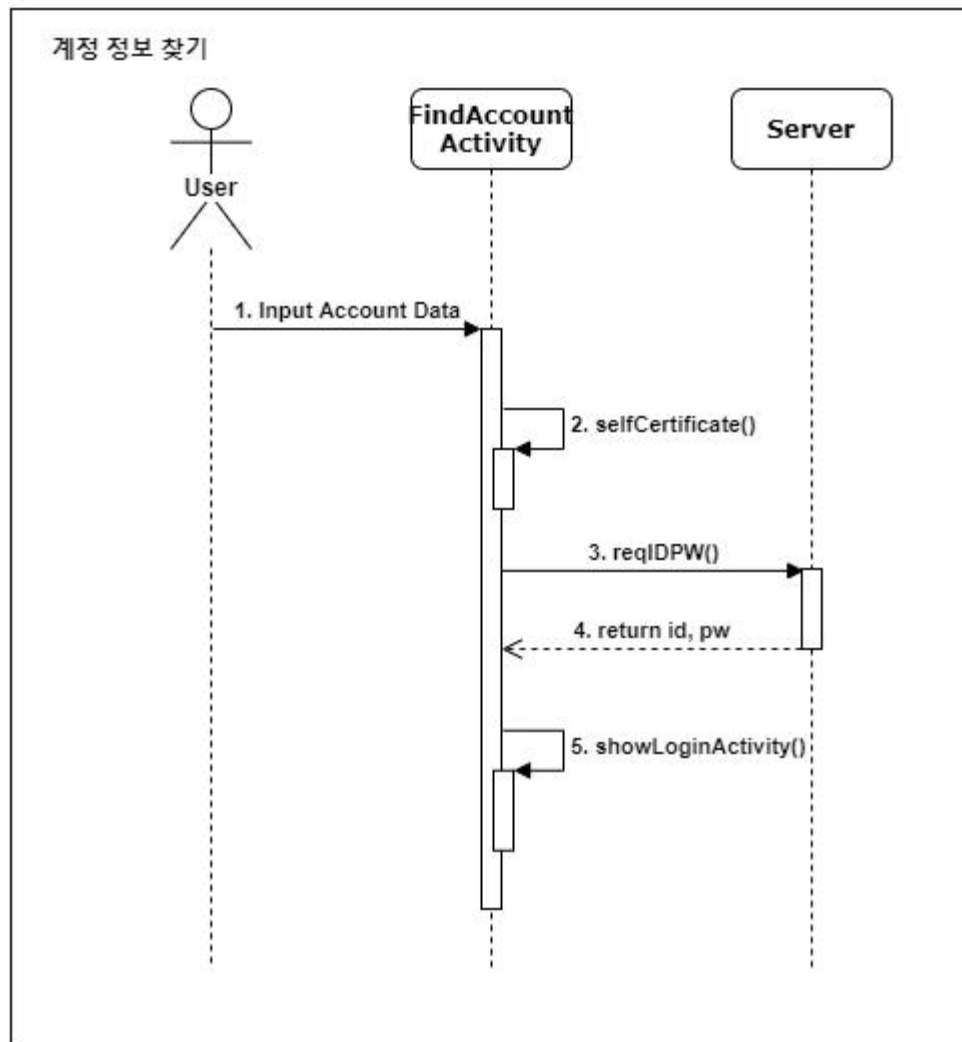
#### 2.1.1 회원가입 순차 다이어그램 설계



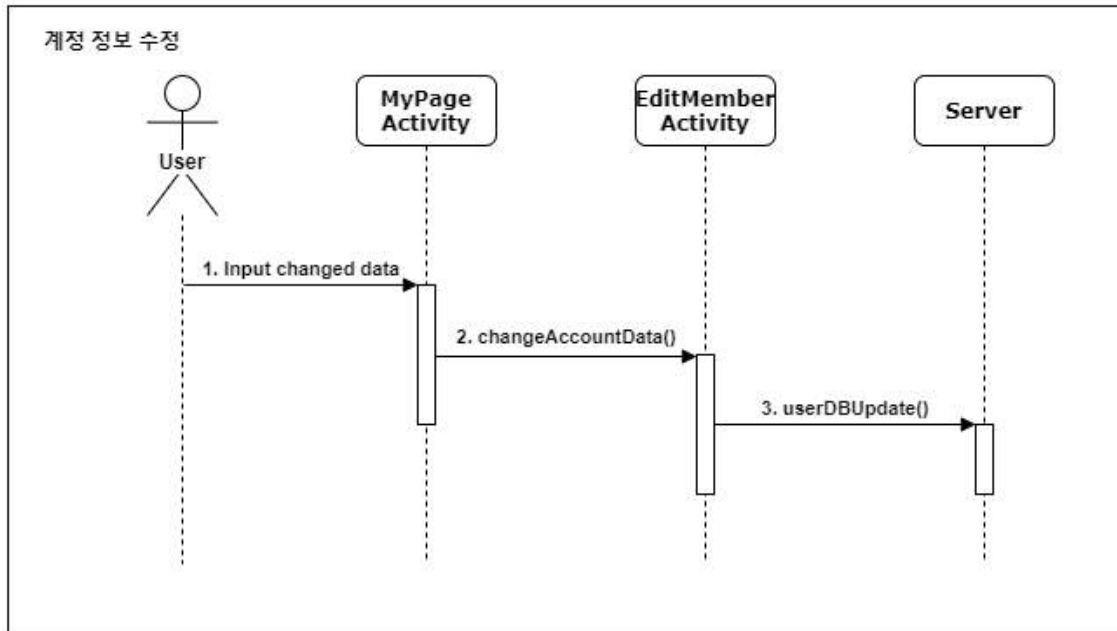
### 2.1.2 로그인 순차 다이어그램 설계



### 2.1.3 계정 정보 찾기 순차 다이어그램 설계

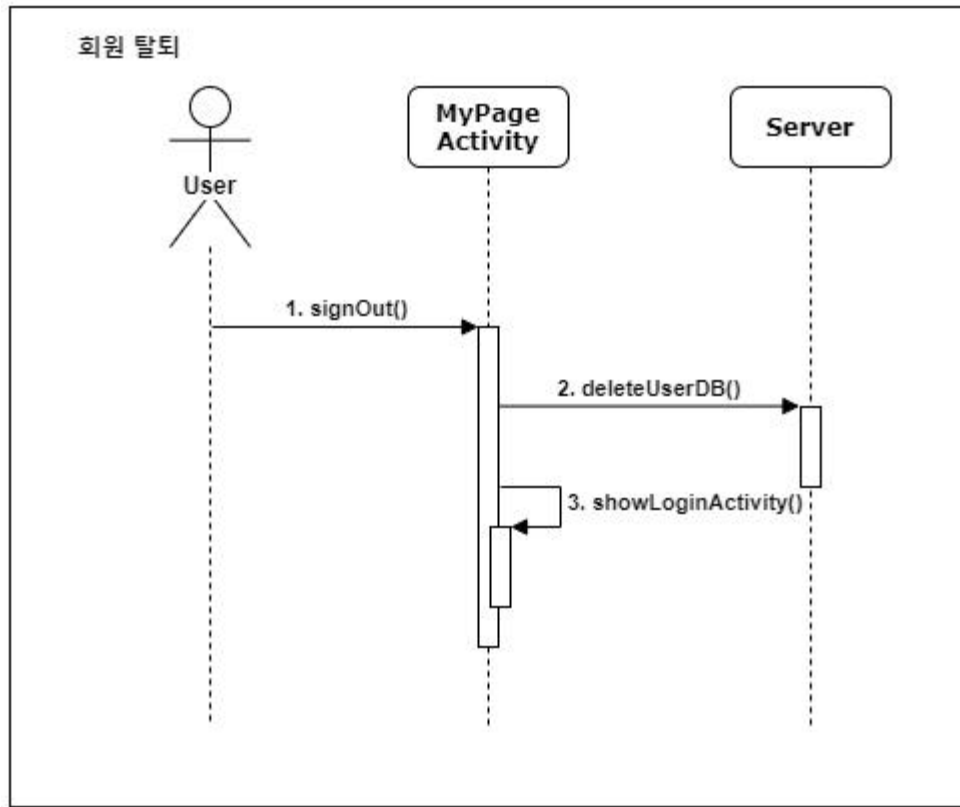


#### 2.1.4 계정 정보 수정 순차 다이어그램 설계



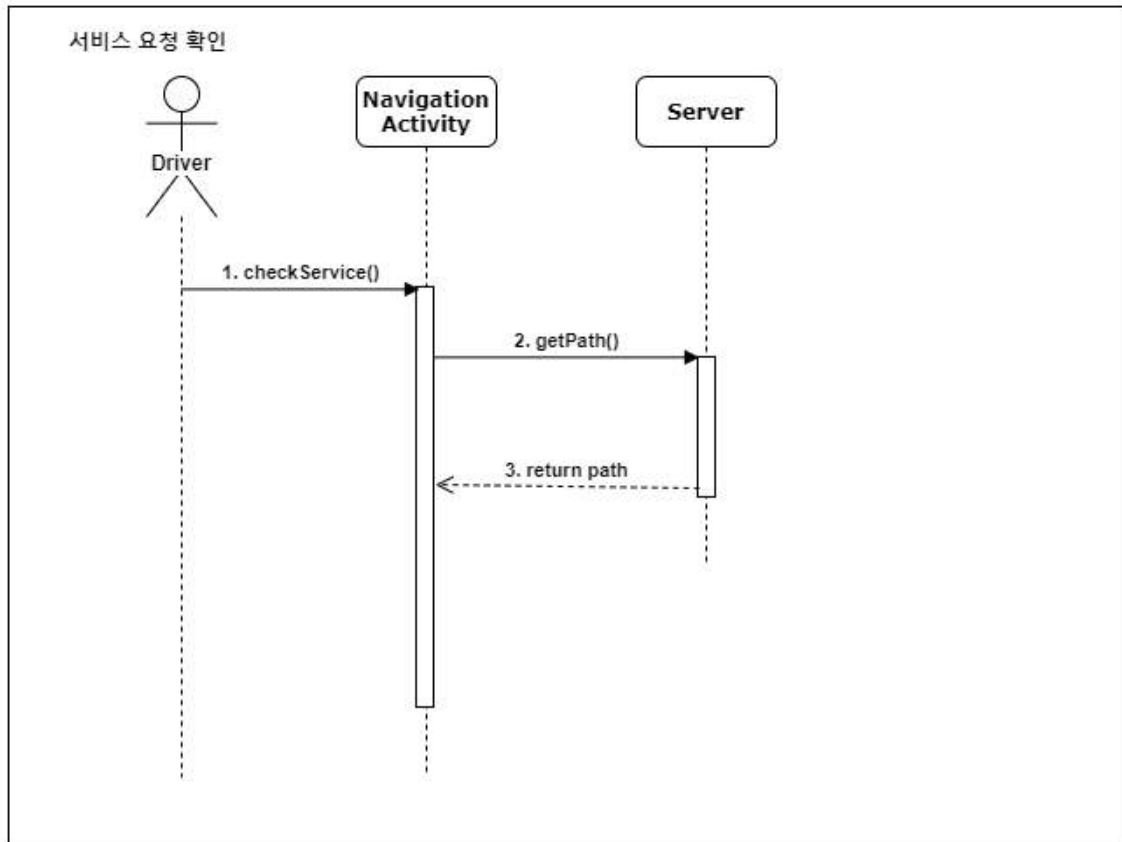


### 2.1.5 회원 탈퇴 순차 다이어그램 설계

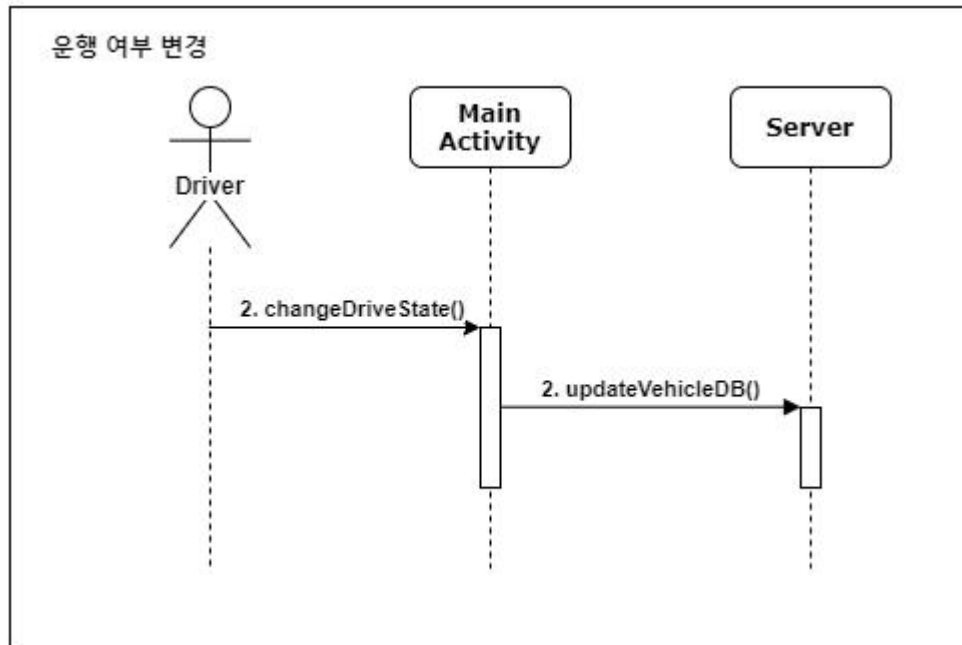


## 2.2 운전자 서비스 순차 다이어그램 설계

### 2.2.1 서비스 요청 확인 순차 다이어그램 설계

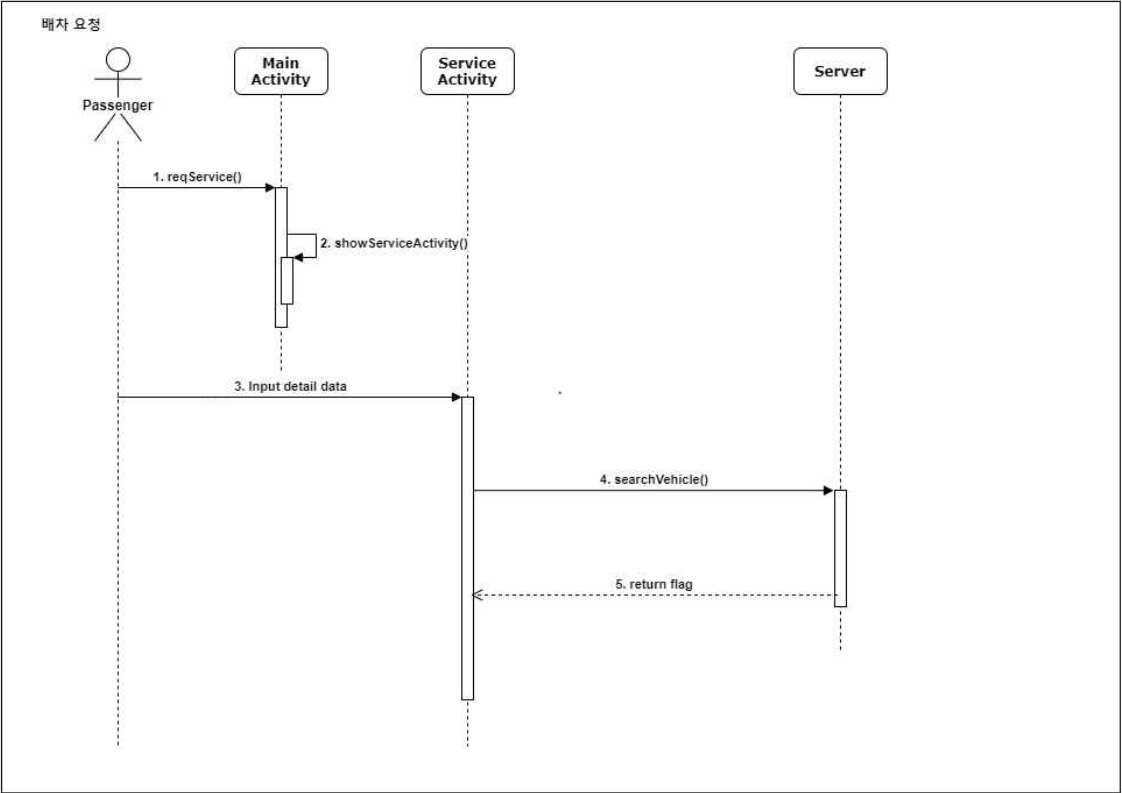


### 2.2.2 운행 여부 변경 순차 다이어그램 설계



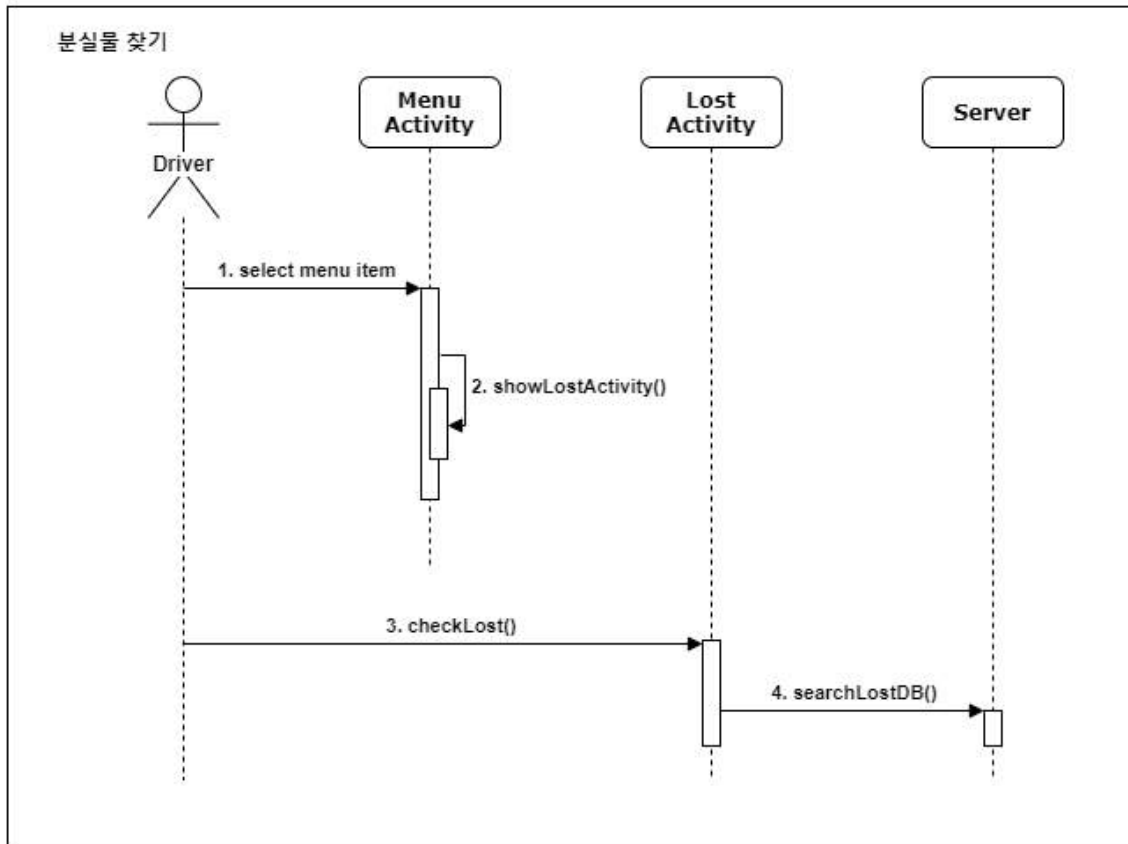
## 2.3 탑승자 서비스 순차 다이어그램 설계

### 2.3.1 배차 요청 순차 다이어그램 설계

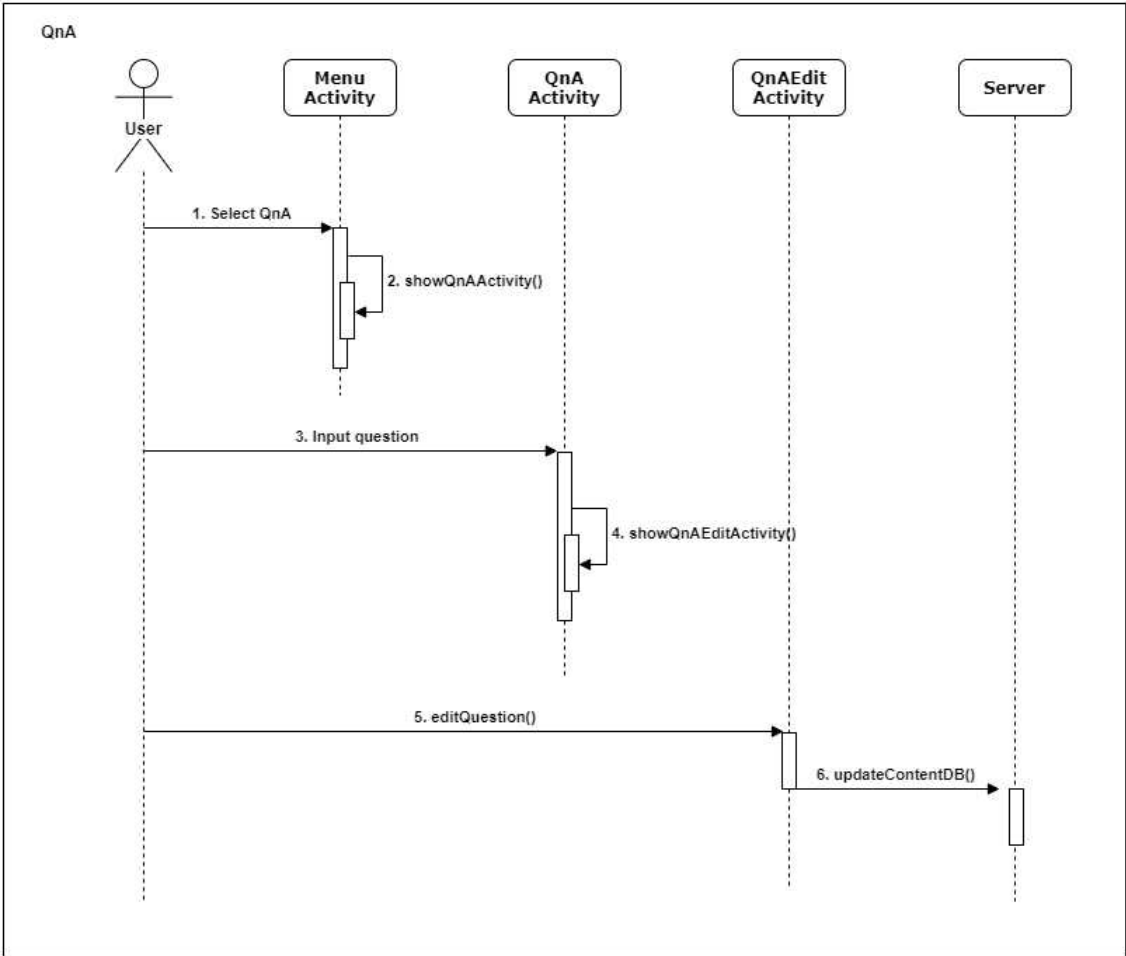


## 2.4 사용자 서비스 순차 다이어그램 설계

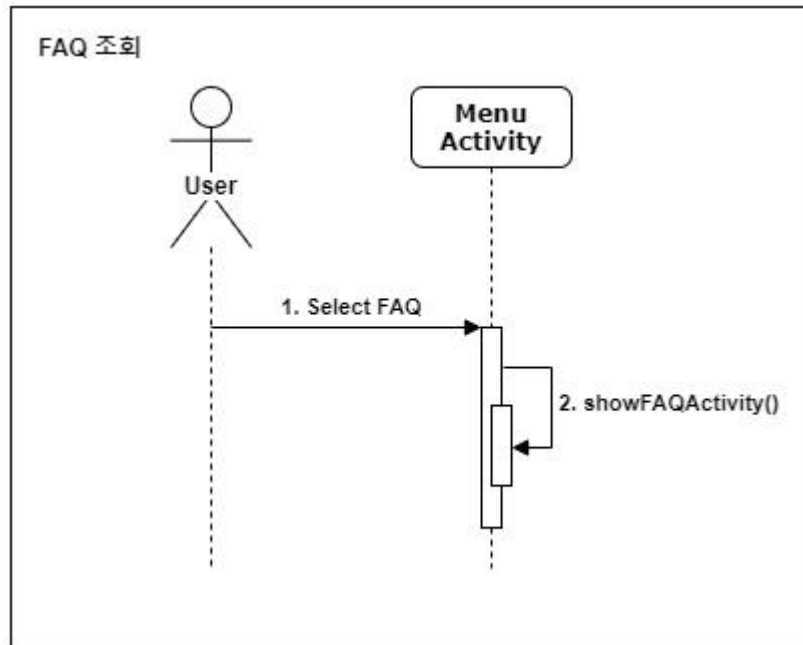
### 2.4.1 분실물 순차 다이어그램 설계



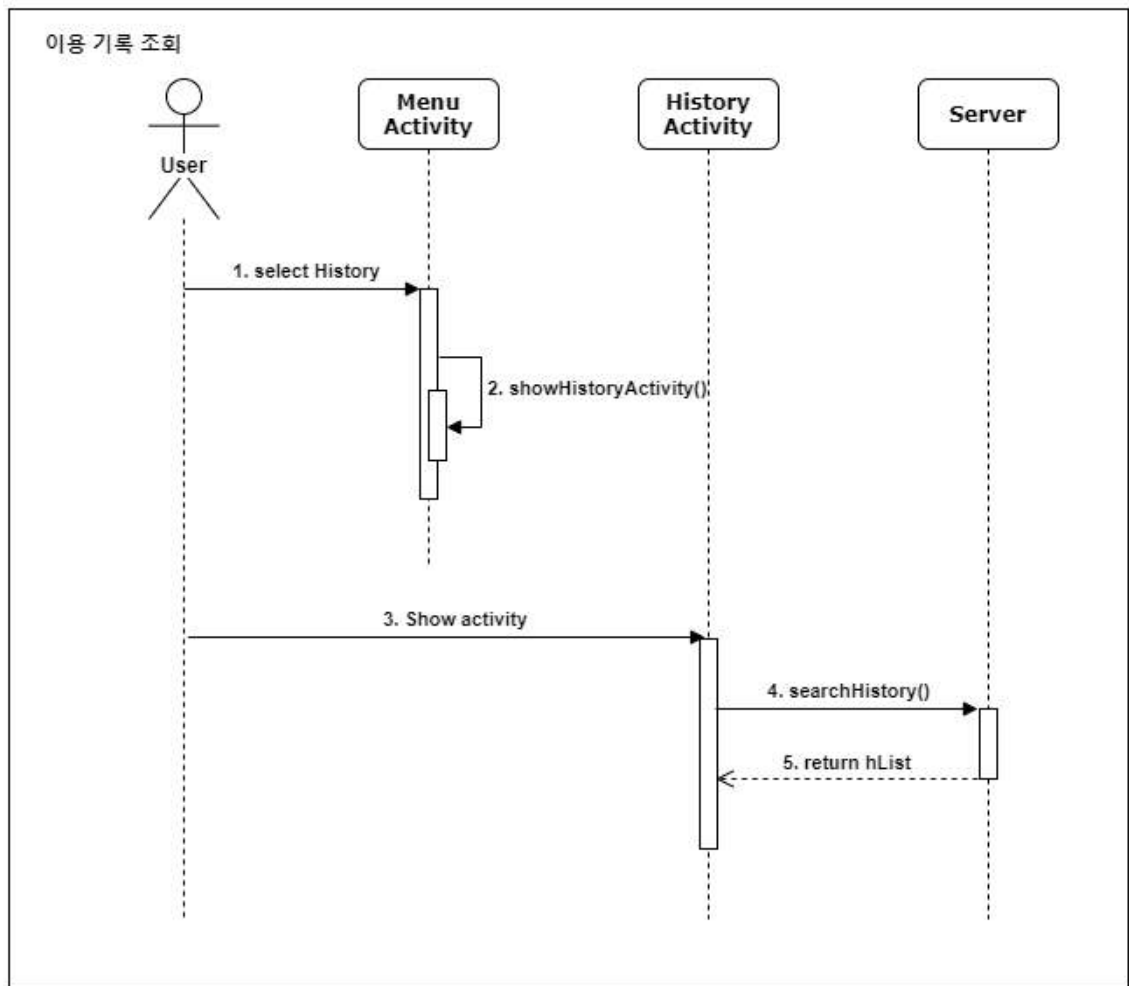
2.4.2 QnA 순차 다이어그램 설계



### 2.4.3 FAQ 순차 다이어그램 설계



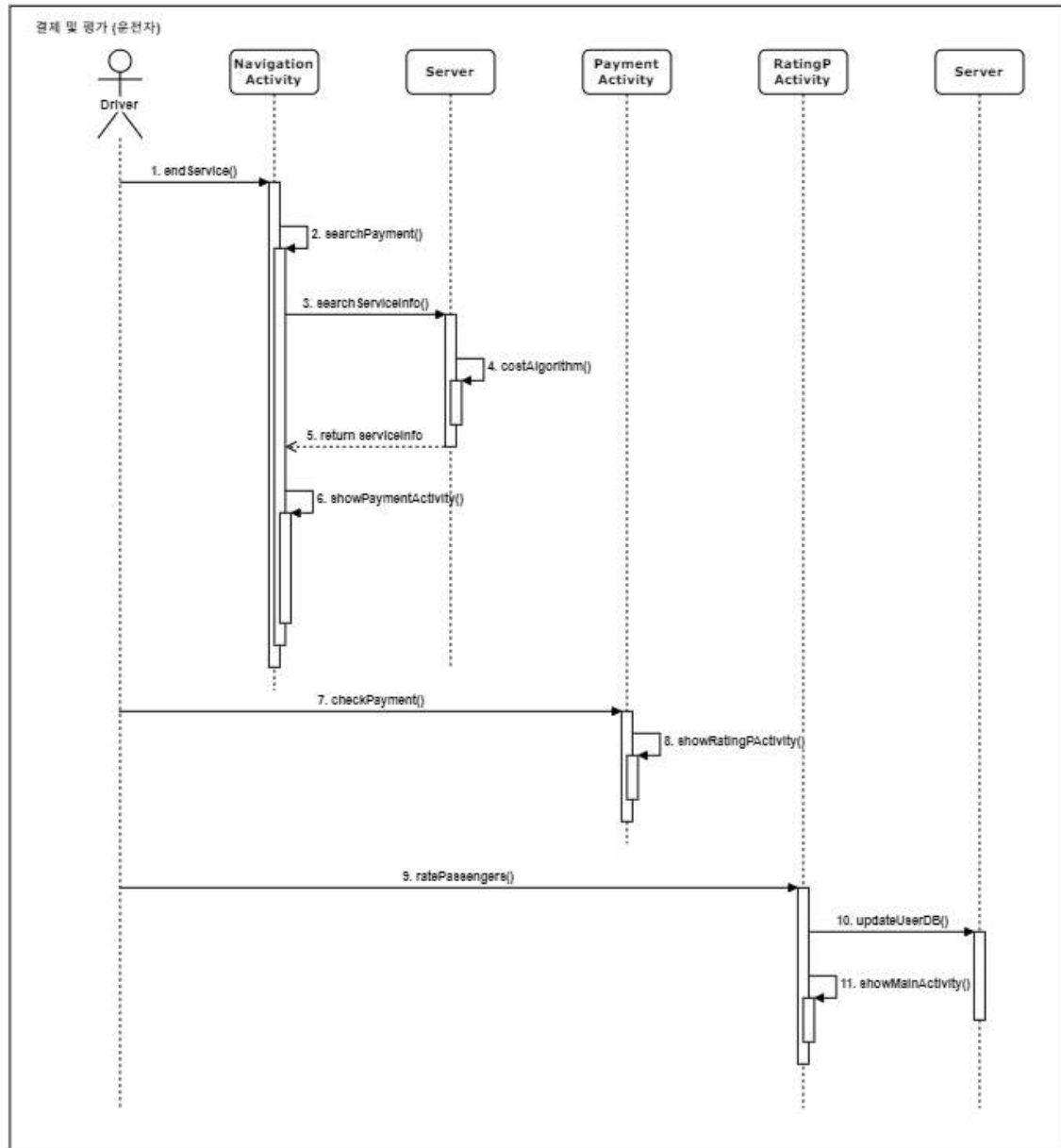
2.4.4 이용 기록 조회 순차 다이어그램 설계



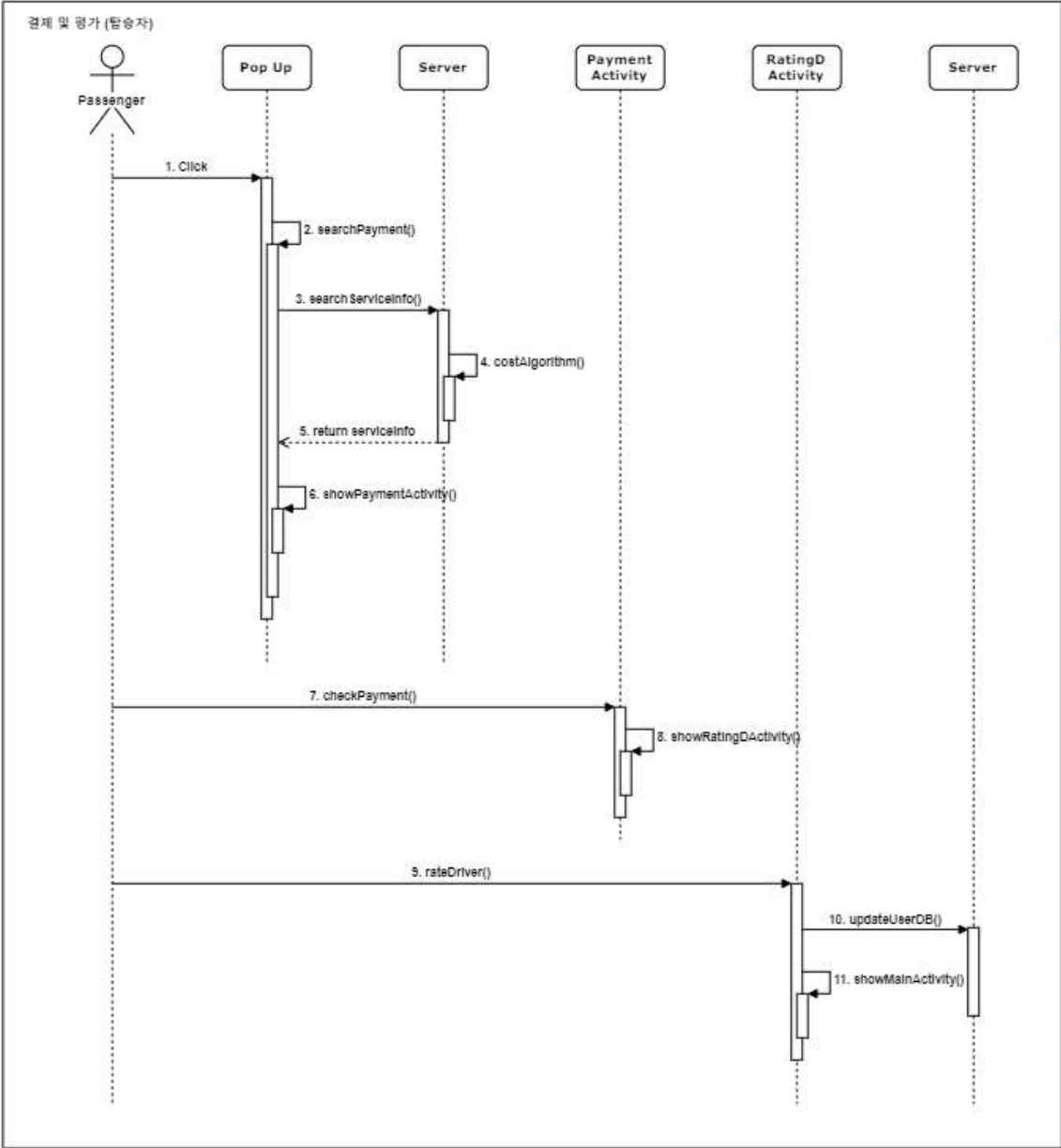


## 2.5 결제 서비스 순차 다이어그램 설계

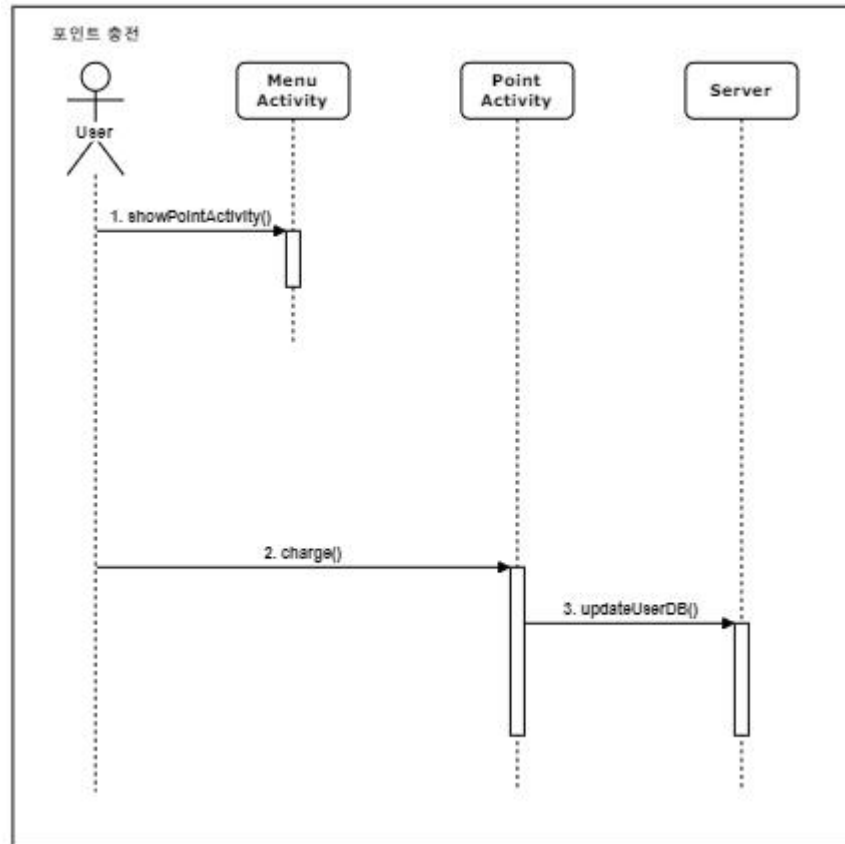
### 2.5.1 결제 및 평가 (운전자) 순차 다이어그램 설계



2.5.2 결제 및 평가 (탑승자) 순차 다이어그램 설계

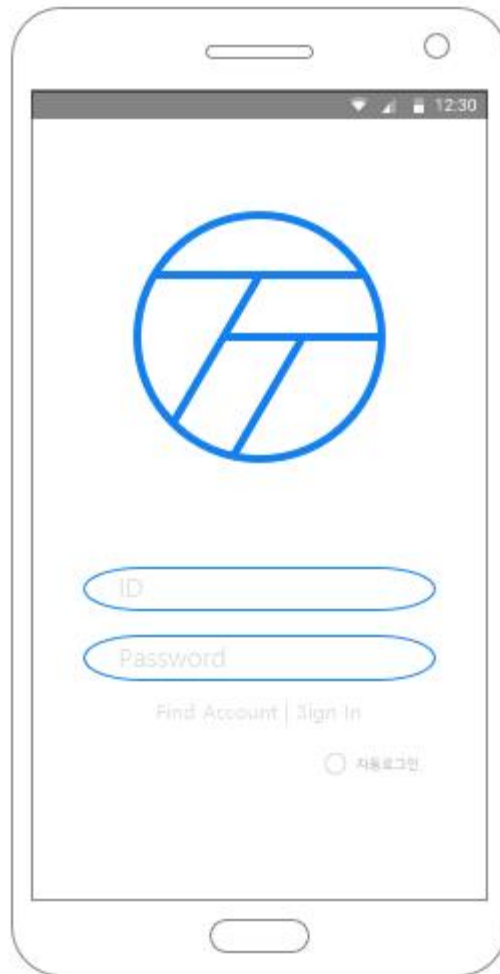


### 2.5.3 포인트 충전 순차 다이어그램 설계



### 3. User Interface 설계

#### 3.1 LoginActivity



### 3.2 SignUpActivity

The image shows a mobile application interface for a sign-up activity. The app is titled "TANNAE" and has a status bar at the top showing the time as 12:30. The interface is divided into two main sections: "Account" and "Personal".

**Account Section:**

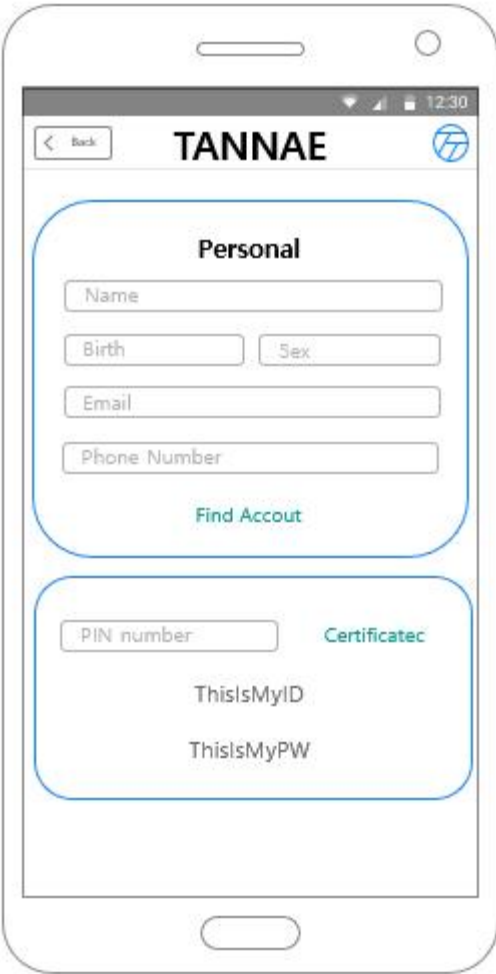
- Contains an "ID" input field with a green "이메일 검사" (Email Check) button to its right.
- Below the ID field is a message: "같은 ID 형식입니다." (Same ID format).
- Below the ID field is a "Password" input field.
- Below the Password field is a "Password Check" input field.
- Below the Password Check field is a message: "비밀번호 길이가 너무 짧습니다." (Password length is too short).

**Personal Section:**

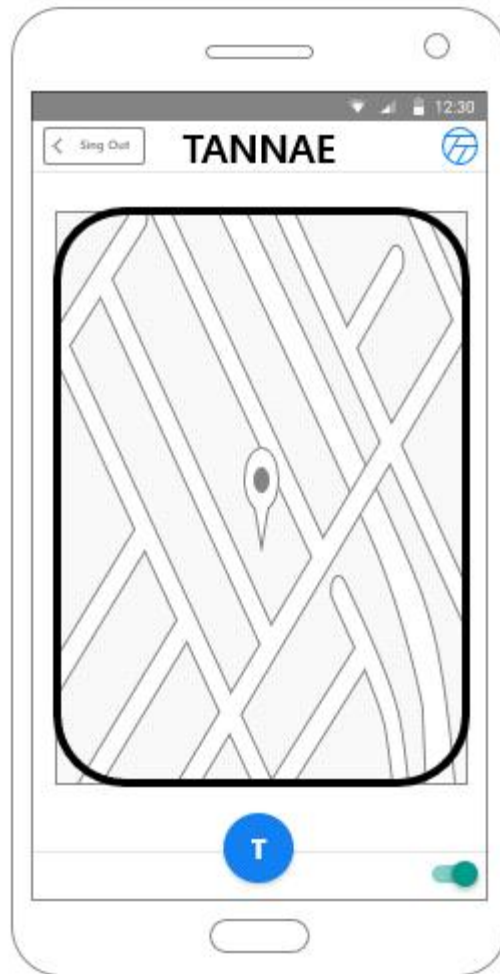
- Contains a "Name" input field.
- Below the Name field are two input fields: "Birth" and "Sex".
- Below the Birth and Sex fields is an "Email" input field.
- Below the Email field is a "Phone Number" input field.

At the bottom of the form, there is a green "Sign Up" button.

3.3 FindActivity



### 3.4 MainActivity

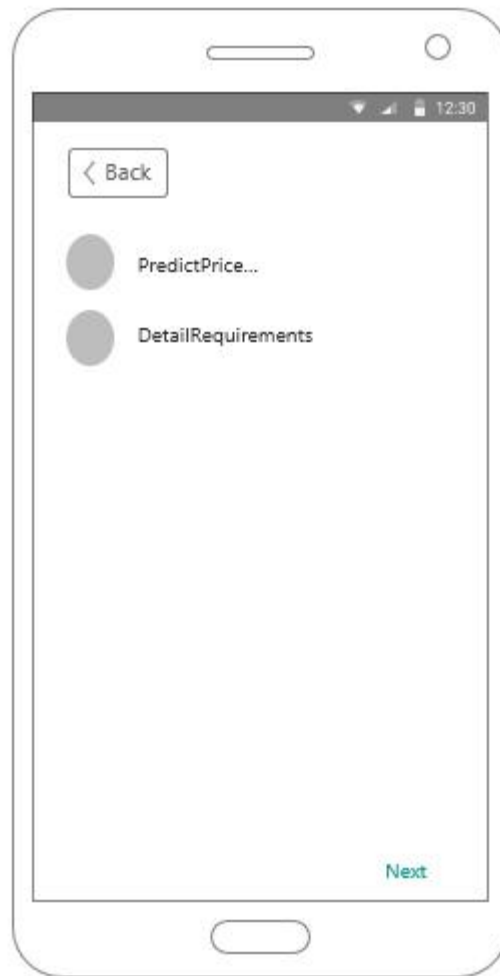


### 3.5 ServiceReqActivity

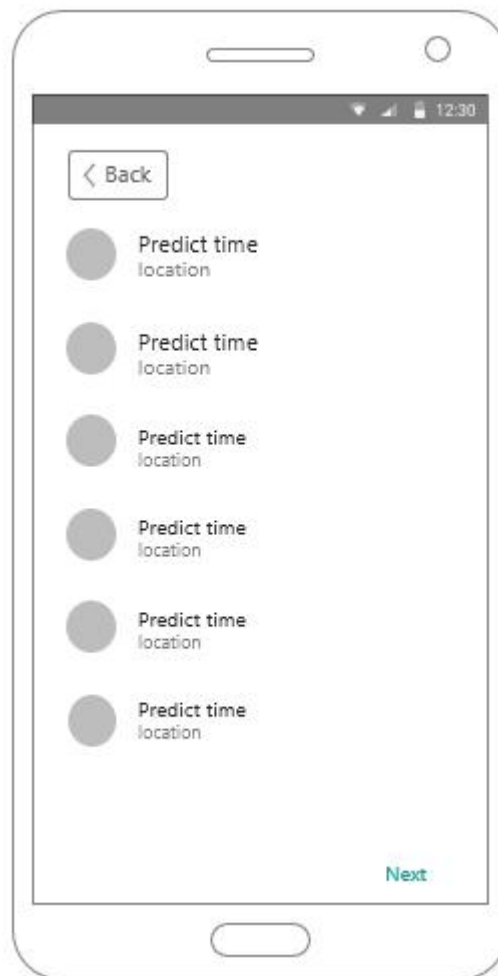




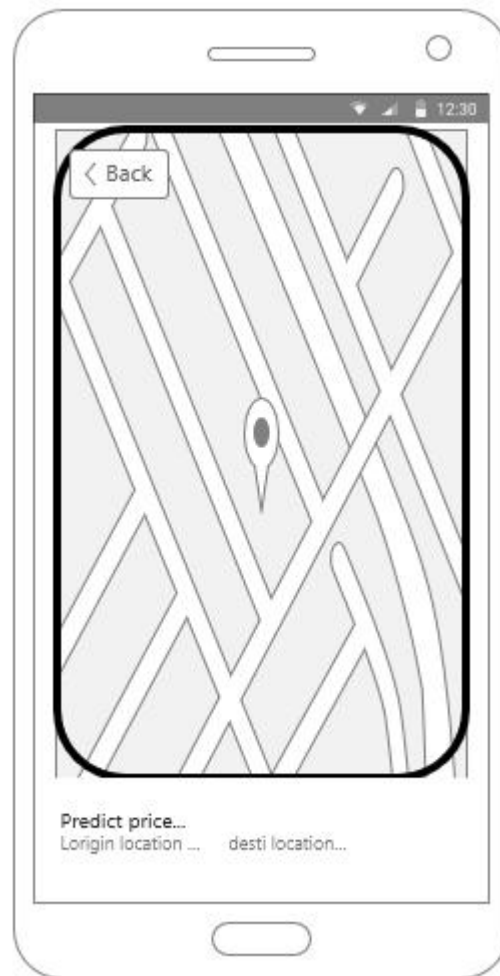
### 3.6 DetailServiceReqActivity



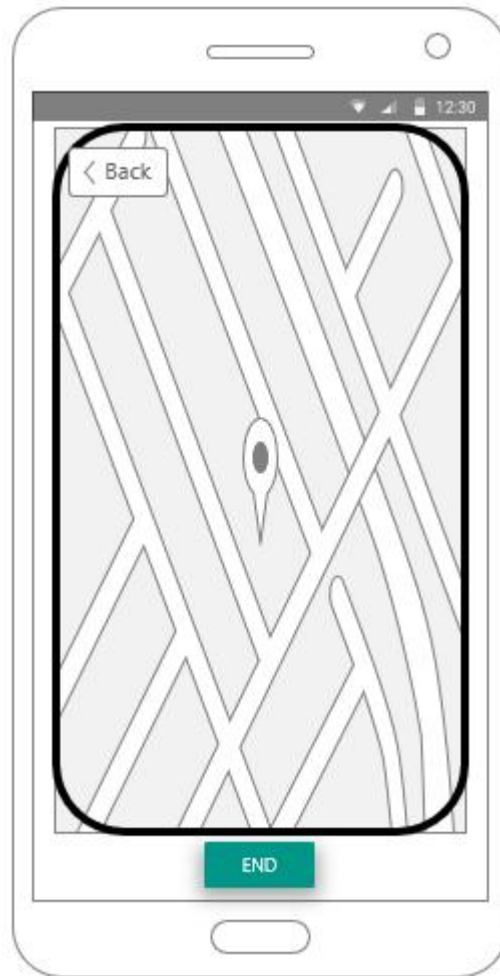
### 3.7 VehicleListActivity



### 3.8 ServiceUsingActivity



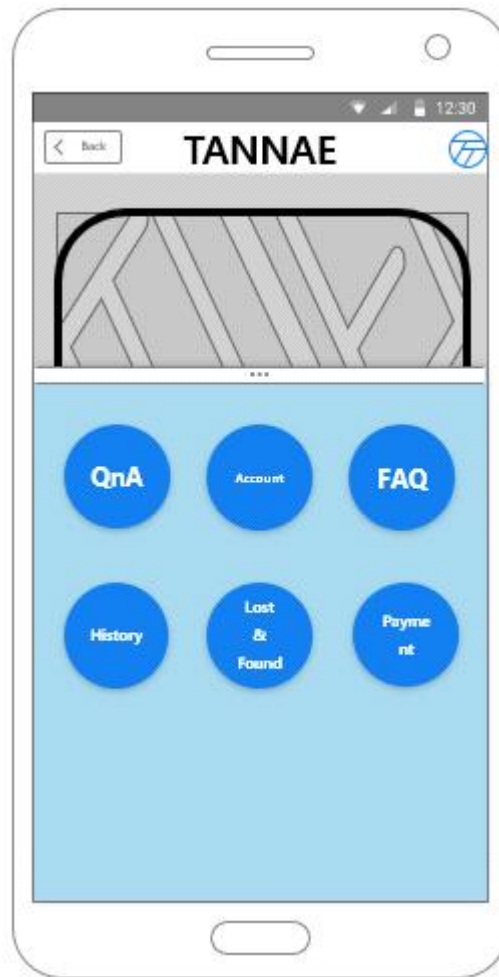
### 3.9 NavigationActivity



3.10 PaymentActivity



### 3.11 MenuActivity



### 3.12 AccountActivity

The image is a wireframe of a mobile application screen titled "AccountActivity". The screen is displayed on a smartphone mockup. At the top, there is a status bar showing the time as 12:30 and various icons. Below the status bar is a header area with a back button labeled "< Back", the app name "TANNAE", and a circular logo. The main content area features a rounded rectangle with the title "Account". Inside this rectangle are several input fields: "ID", "Password", "Name", "Birth", "Sex", "Email", and "Phone Number". The "Birth" and "Sex" fields are side-by-side. Below the rounded rectangle, there are two buttons: "Edit" and "Sign Out".

### 3.13 AccountEditActivity

The image shows a mobile application interface for editing an account. The app is titled "TANNAE" and has a status bar at the top showing the time as 12:30. The interface includes a "Back" button in the top left corner. The main content area is titled "Account" and contains several input fields: "ID", "Password", "Password Check", "Email", and "Phone Number". There are also two buttons at the bottom: "Edit" and "Sign Up".

Account

ID 중복검사  
잘못된 ID 형식입니다.

Password

Password Check  
비밀번호 길이가 너무 짧습니다.

Email

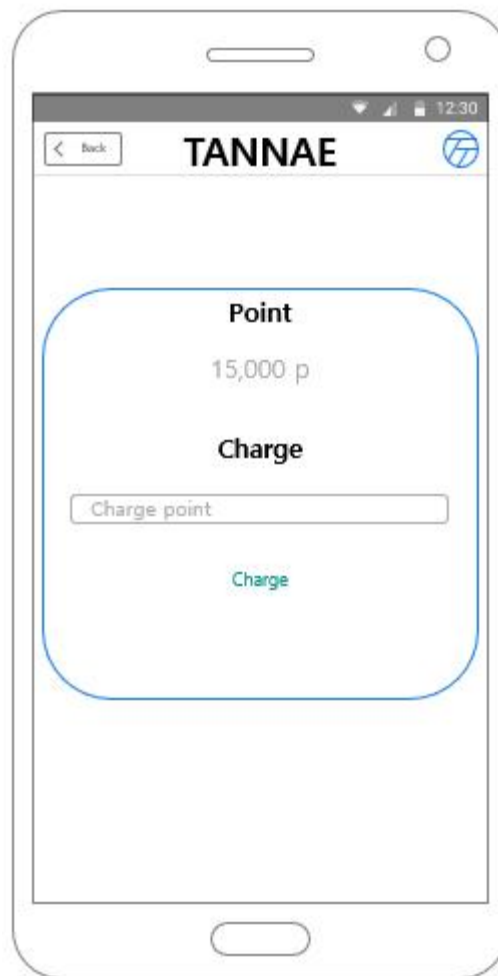
Phone Number

Edit

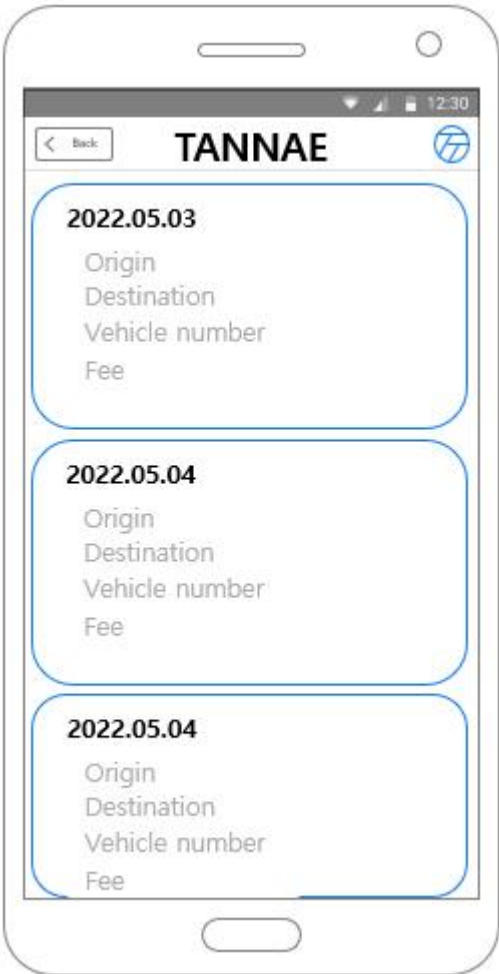
Sign Up



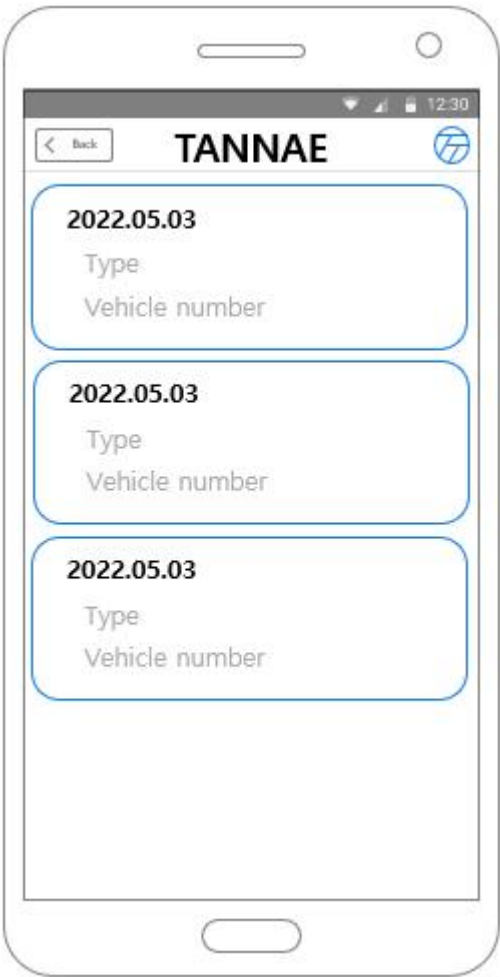
### 3.14 PointActivity



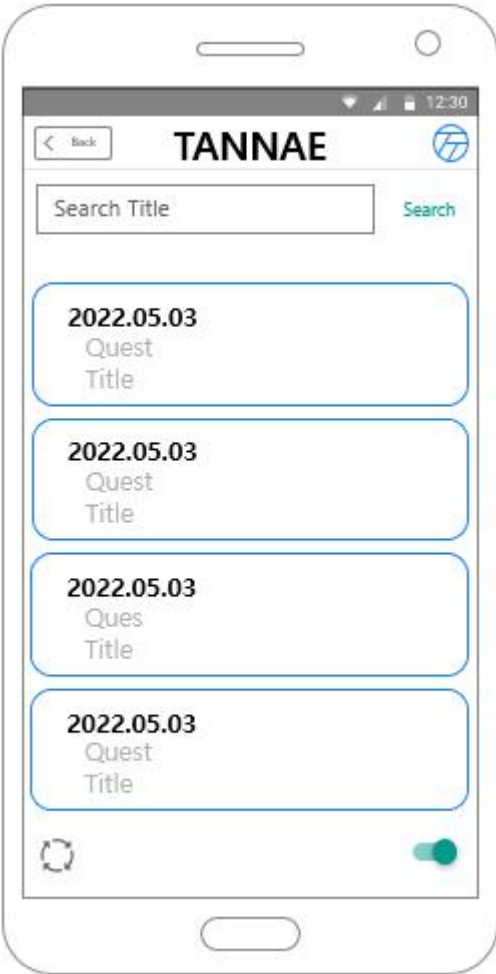
3.15 HistoryActivity



3.16 LostFoundActiviry



3.17 QnAActivity



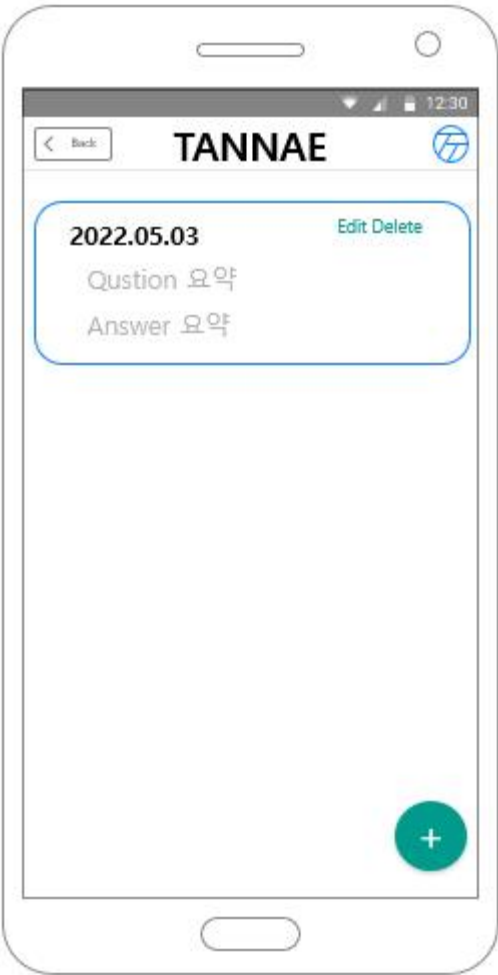
### 3.18 QnADetailAcitivity



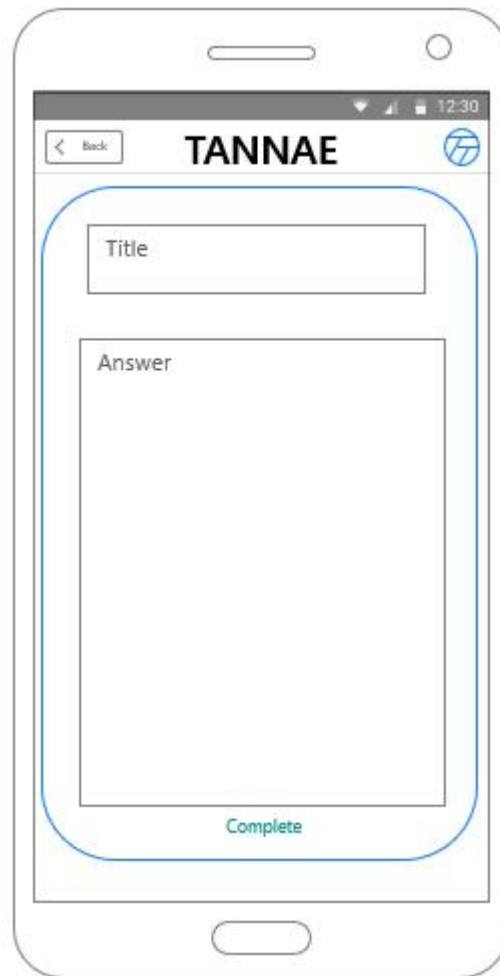
### 3.19 QnAEditActivity



3.20 FAQActivity



### 3.21 FAQEditActivity



The image shows a mobile application interface for editing a FAQ entry. The screen is titled "TANNAE" and features a "Back" button in the top left corner. The main content area is enclosed in a rounded rectangle and contains two input fields: a "Title" field and a larger "Answer" field. A green "Complete" button is located at the bottom of the form. The interface is displayed on a smartphone with a status bar at the top showing the time as 12:30.

Back TANNAE

Title

Answer

Complete



## 4. Database Table 명세서

### 4.1 User database table

테이블 명세서						
시스템명	TANNAE			작성일	2022.03.26	
테이블ID	userDB			작성자	최재원	
테이블명	User					
No	컬럼ID	컬럼명	Type	Length	NULL	Key
1	USN	회원 고유 번호	String	6	not	PK
2	ID	회원 ID	String	4 ~ 20	not	UQ
3	PW	회원 PW	String	4 ~ 20	not	
4	UNAME	이름	String	4 ~ 8	not	
5	RRN	주민등록번호	String	14	not	UQ
6	GENDER	성별	boolean	1	not	
7	PHONE	전화번호	String	11	not	
8	EMAIL	이메일	String	10 ~ 30	not	
9	DRIVE	운전자 여부	boolean	1	not	
10	POINTS	결제 포인트	int	1 ~ 10	not	
11	SCORE	평점	float	3	not	
12	STATE	서비스 이용상태	boolean	1	not	

## 4.2 Vehicle database table

테이블 명세서							
시스템명	TANNAE			작성일	2022.03.26		
테이블ID	vehicleDB			작성자	최재원		
테이블명	Vehicle						
No	컬럼ID	컬럼명	Type	Length	NULL	Key	
1	VSN	차량 고유 번호	String	6	not	PK	
2	STATE	차량 운행 상태	boolean	1	not		
3	POS	위치	String	1 ~ 50	not		
4	LICENSE	차량 번호	String	1 ~ 10	not		UQ
5	NUM	탑승 인원	int	1	not		
6	PASS	통과 경로	String	5000			
7	UNPASS	미통과 경로	String	5000			
8	SHARE	동승 여부	boolean	1			
9	GENDER	탑승객 성별	boolean	1			FK
10	COST	비용	int	1 ~ 10			
11	NAMES	경유지와 USN	String	5000			
12	USN	회원 고유 번호	String	6			

### 4.3 History database table

테이블 명세서						
시스템명	TANNAE			작성일	2022.03.26	
테이블ID	historyDB			작성자	최재원	
테이블명	History					
No	컬럼ID	컬럼명	Type	Length	NULL	Key
1	HSN	히스토리 고유 번호	String	6	not	PK
2	LICENSE	차량 번호	String	1 ~ 10	not	
3	DATE	날짜	int	1 ~ 30	not	
4	ORIGIN	출발지	String	1 ~ 30	not	
5	DEST	목적지	String	1 ~ 30	not	
6	COST	비용	int	1 ~ 10	not	
7	USN	회원 고유 번호	String	6	not	FK

#### 4.4 Lost database table

테이블 명세서						
시스템명	TANNAE			작성일	2022.03.26	
테이블ID	lostDB			작성자	최재원	
테이블명	lost					
No	컬럼ID	컬럼명	Type	Length	NULL	Key
1	LSN	분실물 고유 번호	String	6	not	PK
2	DATE	날짜	int	1 ~ 30	not	
3	TYPE	분실물 내용	String	1 ~ 20	not	
4	LICENSE	차량 번호	String	1 ~ 10	not	
5	VSN	차량 고유 번호	String	6	not	FK

#### 4.5 Content database table

테이블 명세서						
시스템명	TANNAE			작성일	2022.03.26	
테이블ID	contentDB			작성자	최재원	
테이블명	Content					
No	컬럼ID	컬럼명	Type	Length	NULL	Key
1	CSN	내용 고유 번호	String	6	not	PK
2	TITLE	제목	String	1 ~ 20	not	
3	CONTENT	질문	String	1 ~ 500	not	
4	ANSWER	답변	String	1 ~ 100		
5	STATE	답변 여부	boolean	1	not	
6	USN	회원 고유 번호	String	6	not	FK

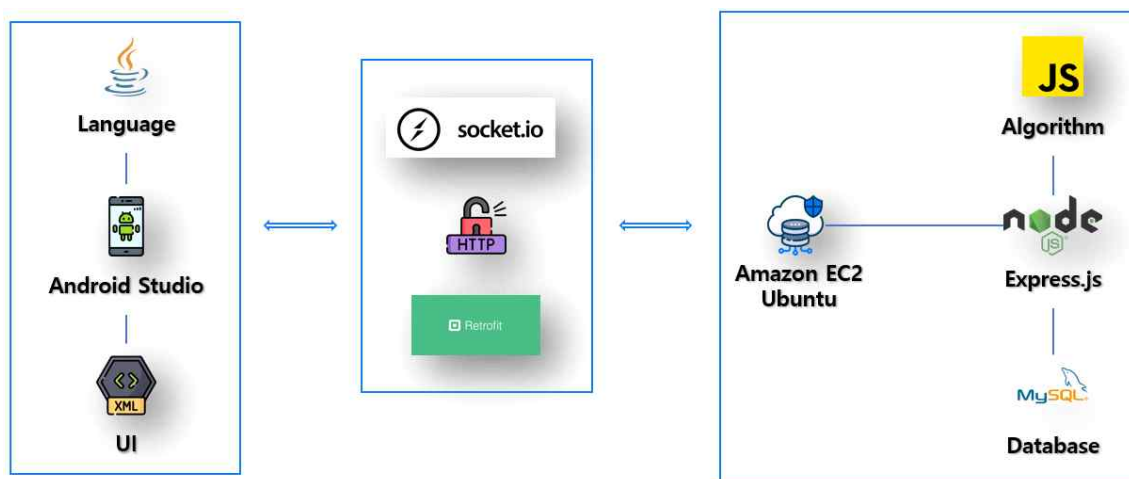
# 프로젝트 구현

## 1. 구현

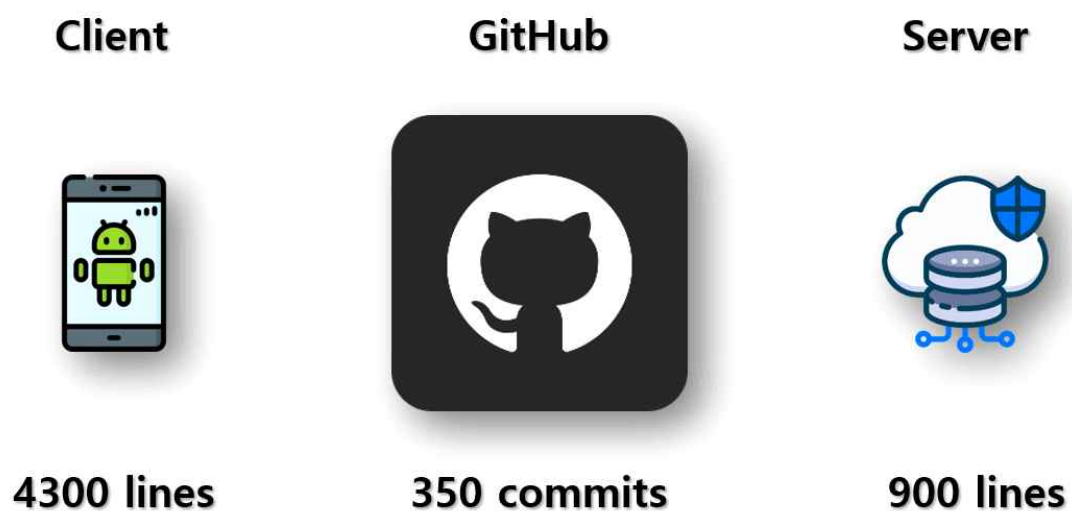
### 1.1 GitHub Link

Link : [jaewonchoi716/Capstone: Capstone Design \(github.com\)](https://github.com/jaewonchoi716/Capstone)

### 1.2 구현 구조



### 1.3 구현 크기



\* 시연 영상은 GitHub README.md 참고

## 2. 알고리즘 효율성

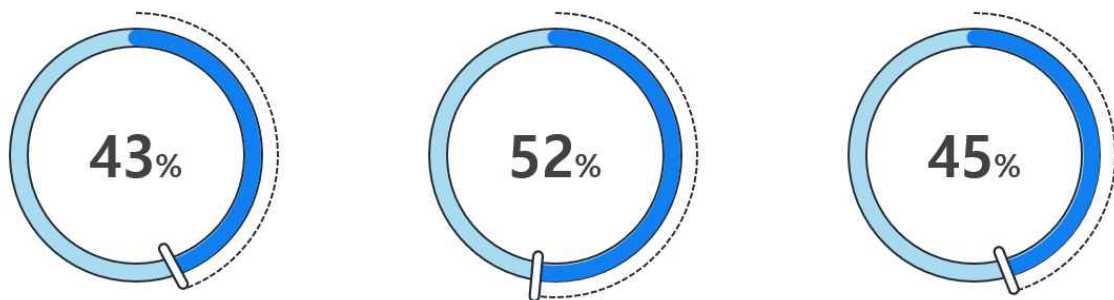
### 2.1 테스트 방법

총 100번의 테스트 케이스를 생성하여 테스트하며 테스트 방법은 다음과 같다. 동일한 차량으로 동승을 하여 서비스를 이용하는 사용자는 총 3명이다. 하나의 테스트는 3명 모두가 탑승하고 하차하였을 때 종료된다. 이 때 서비스를 요청하는 순서는 동일하게 유지하지만 각 탑승자별 탑승 위치와 하차 위치는 모든 경우의 상황을 테스트한다. 또한 거리에 비례하여서 최종 요금을 정산하기 때문에 거리 또한 고려하여 테스트를 하며 서비스 이용 지역은 광주 지역을 한정한다.

### 2.2 테스트 결과

총 100번의 테스트 케이스를 생성하여 테스트 한 결과 첫 번째 서비스 요청자의 요금 할인율은 43% 이다. 두 번째 서비스 요청자의 요금 할인율은 52% 이다. 세 번째 서비스 요청자의 요금 할인율은 46 %으로 평균 할인율은 46%이다. 해당 수치는 처음 설정한 요금 할인율 40~50% 이상에 부합하는 수치로 성공적인 알고리즘을 구현한 것으로 판단이 된다.

이때 두 번째 탑승자의 할인율이 다른 두 탑승자의 할인율보다 큰 이유는 첫 번째 탑승자와 세 번째 탑승자는 처음과 끝에서 혼자 타는 구간이 평균적으로 많이 생기는 반면에 두 번째 탑승자는 상대적으로 더 많은 테스트에서 혼자 타는 구간이 없기 때문이다. 이러한 이유 때문에 향후 Version 2.0에서는 첫 탑승자와 마지막 탑승자의 요금 비율을 낮추고 이를 다른 사용자들이 나누어 내는 요금 계산 방법을 적용하면 처음과 마지막 탑승자의 요금 부담율을 낮출 수 있을 것이라고 예상이 된다.





## 3. 기능 테스트

### 3.1 주요 기능 테스트

#### 3.1.1 운전 상태 변경

1) 운전 상태 default 값

문제 : Default 값을 꺼져 있는 상태로 하였으나 이는 비효율적인 UX로 판단

수정 : Default 값을 true로 설정하여 UX 개선

2) 운전 상태 버튼 위치

문제 : 운전자와 탑승자의 서비스 화면 통일로 Main 화면의 운전 상태 버튼 불필요

수정 : Main 화면과 Navigation 화면의 운전 상태 버튼 통일

3) 운행 종료 시의 운전 상태 버튼 비활성화

문제 : 운행 종료 시 버튼의 수동 활성화로 추가 배차 오류 발생

수정 : 운행이 종료되면 자동으로 버튼이 비활성화

#### 3.1.2 지도

1) 위치 시각화

문제 : 사용자가 지도에서 정확한 위치를 파악 불가능

수정 : 지도의 중심에 PIN을 위치시켜서 위치 파악 UI/UX 개선

2) 경로 가시성 개선

문제 : 경로가 지도 색상과 비슷하여 사용자의 경로 확인 어려움 발생

수정 : 경로 색상 변경 및 두께 향상으로 경로 가시성 개선

#### 3.1.3 서비스

1) 카카오 navigation API 오류

문제 : 실제 도로 교통 상황에 문제가 발생하였을 때 경로가 거부되어 서버 미응답

수정 : 경로가 거부되었을 때의 경우를 처리하여 클라이언트에 응답

2) 서비스 이용 중의 탑승자 화면 전환

문제 : 서비스 이용 중 탑승자가 화면을 전환하면 예기치 못한 activity 생명주기 발생

수정 : Activity 생명주기 코드를 강화하여 화면 전환상의 오류 수정

## 3.2 기타 기능 테스트

### 3.2.1 회원가입

#### 1) 본인인증

문제 : 동일 인물의 다중 계정 생성 가능

수정 : 본인인증 과정을 도입하여 다중 계정 생성 방지

#### 2) ID 중복 검사

문제 : ID/PW가 binary로 저장되어 String과의 비교 연산 문제 발생

수정 : Binary 값을 고려한 비교 연산으로 해결

### 3.2.2 로그인

#### 1) 자동 로그인

문제 : 애플리케이션 실행 시마다 번거로운 로그인

수정 : 자동 로그인 기능 추가로 해결

#### 2) 서버 통신 로그인

문제 : 자동 로그인 시 서버와 통신을 하지 않아 정보 미업데이트

수정 : 서버와 통신하여 로그인 시마다 정보 업데이트

### 3.2.3 회원 정보

#### 1) ID 수정 방지

문제 : 계정 정보 수정 시 ID 강제 변경 문제 발생

수정 : ID 수정 기능을 삭제

### 3.2.4 포인트 충전

#### 1) 입력 타입 수정

문제 : 숫자 이외의 기호 입력 가능으로 오류 발생

수정 : EditText의 type을 number로 수정하여 숫자 입력만 허용

#### 2) 공백 충전 불가

문제 : 공백을 입력한 상태에서 충전할 시 충전 요청이 되면서 서버에서 오류 발생

수정 : 공백을 입력한 상태에서 충전을 요청하면 충전 거부

# 마무리

## 1. 활동

### 1.1 진행한 활동

#### 1.1.1 산학 협력 프로젝트

1) 멘토 : 메디큐 교육개발원 원장 김현철

2) 회의 진행 방법, 일자 및 주제

회의 방법	회의 일자	회의 내용
온라인	2022년 04월 22일	주제 설명 및 개발 계획서 확인 향후 분석 및 설계에 있어서 고려 사항 정리
오프라인	2022년 04월 29일	분석 및 설계 보고서 확인 문제점 추출과 해결방안 회의
온라인	2022년 05월 13일	중간 구현 상태 확인 UI/UX 측면에서의 개선 사항 회의
오프라인	2022년 05월 27일	최종 구현 결과 확인 및 발표 준비 유지보수와 사업적 측면에 대한 회의

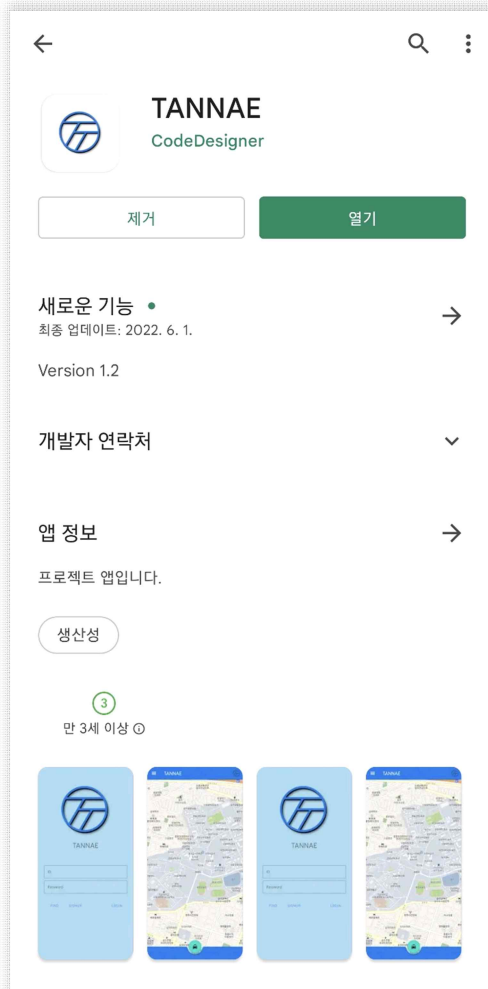
### 3) 회의 내용

개발자에게 있어서 중요한 것은 새로운 기술을 개발하거나 문제를 해결하는 능력도 있지만, 유지보수 또한 굉장히 중요한 능력이다. 이에 따라 이번 캡스톤 디자인 프로젝트 구현 결과를 Version 1.0으로 설정하고 Version 2.0에 대해 계획을 하여 더 좋은 서비스를 제공할 수 있도록 하여야 한다. 이에 따라 Version 2.0에서는 Version 1.0의 기술적인 부분으로 개선하며 사업적인 부분까지도 다루는 것을 계획하였다.

먼저 기술적으로는 IOS 개발도 진행할 예정이다. 현재 스마트폰 시장은 크게 안드로이드와 IOS 시장으로 나뉘며 IOS 이용자들도 아주 많은 상황이다. 따라서 서비스 이용자들을 증가시켜서 서비스를 활성화하기 위해서는 IOS 개발도 필요하다. 이에 맞춰서 인공지능 기능도 탑재할 예정이다. 현재 내부 알고리즘은 고정된 수치로 문제를 해결하며 사용자에게 서비스를 제공한다. 그러나 여러 가지 상황을 고려하였을 때 배차 가능 영역 등은 동적으로 변화하여야 한다. 다양한 상황들을 고려하는 인공지능을 탑재하면 알고리즘 관련 수치들은 인공지능이 동적으로 변경시키며 탑승자들의 할인율을 극대화할 수 있을 것이다. 그러나 Version 1.0의 가장 큰 단점은 서버 중심의 서비스라는 것이며 이는 Version 2.0에서 가장 우선 해결되어야 할 과제이다. 예를 들어 탑승자들은 서비스를 이용할 때 서버에서 지정해준 차량을 이용할 수밖에 없다. 차량의 상황이나 탑승자들의 수를 고려할 수 없으며 출발지/목적지와 동승 여부를 제외하고는 다른 요청사항을 등록할 수 없다. 이는 서비스의 품질을 감소시키는 주된 원인이며 이를 해결하기 위해 여러 부분에 있어서 사용자들에게 직접 결정을 할 수 있는 부분을 제공하는 서비스로 변화하는 것을 계획하였다.

개발자는 개발만을 생각하면 되겠지만 하나의 프로젝트를 구상하고 설계부터 구현, 유지보수까지 진행하는 캡스톤 디자인에서는 사업적인 부분 또한 제외할 수는 없을 것이다. 사업에서 가장 중요한 것은 이익을 창출하는 것이다. 이익 창출에는 다양한 방법이 있겠지만 이번 프로젝트에서 회사의 주요 인원들인 택시 기사들이 월급을 받기 때문에 택시 요금에서 수수료를 내는 것 같은 방법은 적절하지 못한 방법이다. 그렇기에 수입에서 사업적 이익을 창출하는 것이 아니라 수입 자체를 극대화하는 방법을 구상하여야 했다. 이에 대해 서비스가 활성화되고 사용자들이 증가하면 자동으로 수입이 증가할 것으로 판단하여 사용자 유치를 위한 다양한 프로모션을 진행하는 방법을 채택하였다. 또한 서비스를 탑승자들에게 제공하여 주는 운전자들이 이번 프로젝트에 참여하도록 유도를 하여야 한다. 따라서 운전자들이 이번 서비스를 충분히 이해할 수 있도록 교육하고 한 명의 직원으로서 다양한 복지를 제공하면 사업적 이익은 자동으로 발생할 것으로 판단하였다.

## 1.1.2 구글 플레이 스토어 배포



## 2. 질의 사항

### 2.1 탑승자들의 하차 순서

#### Q. 최대 동승 가능한 인원은 몇 명인가?

동시에 동승 가능한 인원은 3명입니다. 다만, 한 번의 운행에서 최대 3명이 탑승할 수 있는 것은 아닙니다. 예를 들어 2명이 동승을 하는 상태에서는 한 명이 추가로 동승을 할 수 있습니다. 이때 한 명이 하차하게 되면 또다시 추가 동승 가능 인원이 1명이 됩니다. 즉, 특별한 경우가 발생하지 않는 한 최대 3명을 기준으로 계속해서 동승자들이 탑승했다 하차했다 할 수 있고 이론적으로는 운행이 무한히 지속될 수 있습니다.

#### Q. 뒷좌석에서 안쪽에 앉아 있는 사람이 우선 하차하는 경우는 어떻게 하는가?

2가지 방법으로 이를 Version 2.0에서 해결할 예정입니다. 먼저 서버 중심의 서비스에서 사용자 중심의 서비스로 개편하면서 탑승자는 자동 배차가 아니라 서버에서 추천해준 차량 리스트를 바탕으로 타고자 하는 차량을 선택할 수 있습니다. 이때 차량 리스트에는 차량별 세부 사항들이 명시되어 있으며 현재 탑승자들이 앉아 있는 자리를 확인할 수 있습니다. (이는 차량 시트에 센서를 부착하여 구현 가능할 것으로 생각합니다). 만약 탑승자가 뒷좌석에서 안쪽에 앉은 탑승자를 위해 잠깐 하차하는 상황을 원하지 않으면 조수석에 앉을 수 있는 차량을 선택하면 됩니다.

하지만 이러한 해결책은 사용자들에게 해결법을 어느 정도 말긴 상황이 됩니다. 따라서 Version 2.0의 배차 알고리즘에는 이러한 부분까지 고려할 예정입니다. 현재 차량 탑승자들의 탑승 좌석(조수석/뒷좌석)과 하차 순서를 고려하여 탑승자가 새롭게 동승이 되었을 때 불편함을 감수해야 하는 경우를 계산하여 배차 가능 여부를 반환합니다. 이렇게 알고리즘에 하차 순서 데이터를 추가하면 질문의 문제는 해결이 됩니다.

### 2.2 운전자의 이익

#### Q. 탑승자의 요금 감소에 따른 운전자 이익 감소는 어떻게 해결하였는가?

‘TANNAE’ 서비스의 가장 큰 장점은 연속성 있는 서비스 제공입니다. 일반적으로는 승객을 태우면 운행이 종료될 때까지 다른 승객을 태우지 못합니다. 반면에 동승 서비스를 이용하면 연속적으로 승객을 태울 수 있으며 더 많은 요금을 받을 수 있습니다. 물론 그렇다고 해서 전체적인 수익이 증가하는 것은 아닙니다. 실질적인 수익 증가의 원인은 동승으로 인한 경로 변화에 있습니다. 원래의 경로에서 동승자 추가에 따라 새로운 탑승자를 태우거나 하차시키기 위해 이동 거리가 증가합니다. 거리 증가와 함께 전체 비용도 증가하면서 수익이 증가합니다. 하지만 동승을 하면 그만큼 요금을 나누어서 내기 때문에 여전히 탑승자의 요금은 일반 서비스와 비교하였을 때 감소한 요금이며 이에 따라 서비스 제공자와 이용자 모두 경제적 이익을 얻을 수 있습니다.

## 2.3 탑승자들 간의 마찰

### Q. 이성 간의 동승이 문제가 되지는 않을 것인가?

택시 서비스에서의 동승이 합법화되기는 하였지만, 여전히 어느 정도의 규제는 있으며 동성 간의 동승만이 가능하고 이성 간의 동승은 금지되어 있습니다. 이러한 규제를 사전에 파악하였고 이를 알고리즘에 반영하였습니다. 탑승자가 없는 차량과 새로운 탑승자가 매칭되면 차량의 성별 데이터는 탑승자의 성별로 초기화됩니다. 그리고 배차 알고리즘에서는 배차 가능한 차량을 탐색할 때 서비스 요청자의 성별을 판단하여 탑승할 수 없는 차량은 사전에 배제합니다. 이를 통해 이성 간의 동승을 방지하며 모든 탑승자가 차량에서 하차하면 차량의 성별 데이터는 다시 null 값으로 초기화됩니다.

### Q. 서비스의 품질을 악화시킬 수 있는 탑승자들은 어떻게 해결할 것인가?

한 번의 서비스가 종료되면 탑승자들은 운전자에 대해 평가를 하게 되어 있어서 부족한 서비스를 제공한 운전자는 관리를 할 수 있습니다. 초기에는 탑승자들도 운전자들처럼 평점을 부가할 생각이었으나 이는 메인 서비스에 집중하여 구현하자는 의견에 따라 우선순위에 밀려나 구현은 되어 있지 않습니다. 하지만 데이터베이스 상에서는 모든 서비스 이용자들에게 대한 평점 속성이 있는 상태이며 향후 Version 2.0에서는 탑승자의 평가도 추가하여 평점이 좋지 않은 탑승자들을 블랙리스트로 관리할 예정입니다.

## 2.4 연속 운행에 대한 운전자의 불편함

### Q. 지속해서 배차가 들어왔을 때 운전자는 계속해서 운행해야 하는가?

프로젝트를 구상하면서 'TANNAE' 서비스의 가장 큰 장점을 연속성 있는 서비스 제공으로 설정하였습니다. 이는 일반 택시 서비스보다 향상된 수익을 가져오는 방법으로 필수적인 요소입니다. 다만, 이론적으로는 한 차량에 계속해서 배차가 들어올 수 있으며 이러한 상황에서는 운전자가 쉬지 않고 운행을 해야 하는 경우가 발생합니다. 저희는 이를 방지하기 위하여 운전자의 운행 화면에 switch 버튼을 배치했습니다. 버튼이 활성화되어 있다면 계속해서 배차를 받을 수 있고 비활성화되어 있다면 더 이상의 배차는 들어오지 않습니다. 다만, 이러한 switch 버튼 2가지의 문제를 일으킵니다.

먼저 운전자들에게 월급의 형태로 임금을 지급하는 것으로 구상하였기 때문에 일을 하지 않는 상황이 발생할 수도 있습니다. 이러한 문제는 근무 시간에 휴식에 가능한 적절한 시간을 설정하여 해결할 수 있습니다. 반면에 switch 버튼을 비활성화시키면 현재 탑승한 사용자들은 서비스의 장점을 추가로 이용할 수 없게 됩니다. 이에 대해서는 switch 버튼 비활성화 조건을 설정하여 해결할 수 있을 것으로 보이나 이는 실제로 서비스를 운영하면서 운전자들과 탑승자들의 의견을 통해 고쳐나가야 한다고 생각합니다.