

### 3. Design

## Helmet Safer



Student NO	21912075
Name	박해세
E-mail	hessepark7@gmail.com

# [ Revision history ]

Revision date	Version #	Description	Author
2023/06/09	1.00	First draft	박혜세

= Contents =

1. Introduction .....	1
2. Class diagram .....	2
3. Sequence diagram .....	9
4. State machine diagram .....	15
5. Implementation requirements .....	15
6. Glossary .....	16
7. References .....	17

## 1. Introduction

### 1) Summary

공유 킥보드를 타고 다니면 이동 시간 단축과 걷지 않고 목적지까지 편안하게 이동할 수 있어서 많은 사람들과 특히 학생들이 애용을 한다. 하지만 공유 킥보드 시장이 활성화되고 나서 여러 문제점들이 발생했다. 하나의 큰 문제점은 교통사고인데, 사고 원인은 대부분 운전미숙과 과속에 따른 충돌로 이 사고로 인해 머리와 얼굴 부위를 다치는 사고가 전체 30%에 이르는 것으로 확인되었다. 사고 예방을 위하여 전동 킥보드도 오토바이 운전자와 같이 안전모 미착용 시 범칙금 2만 원을 내도록 전동 킥보드 법을 개정하였다. 하지만 킥보드 사용자들이 실제로 헬멧을 잘 착용하지 않고, 분실 문제 때문에 공유 킥보드 업체에서도 헬멧을 적극적으로 제공하지 않고 있다. 이러한 문제를 해결하는 방안으로 헬멧 안에 킥보드에 연동이 된 OTP 기능이 있는 소형 IOT 칩을 삽입하는 방법을 떠올려 보았다. **사용자는 로그인 후 먼저 헬멧에 적힌 고유 번호를 입력해야 한다. 이후 헬멧과 킥보드가 가까운 위치에 있는 것이 확인이 되지 않거나, 헬멧을 썼다는 압박이 감지되지 않으면 킥보드를 출발 시킬 수가 없는 형태, 이용자가 타고 가다가 헬멧을 버릴 수도 있으니 킥보드와 헬멧 사이의 거리가 일정 거리 이상이 되면 킥보드의 이용이 불가하거나 속도제한이 걸리는 상태로 프로그램을 설계해보려 한다.** 헬멧을 쓰지 않거나 불법 주차를 했다는 다른 사용자들의 신고 (Report에 킥보드 번호를 쓰면 시스템에서는 가장 최근 이용한 사용자를 알 수 있다.)를 받으면 경고문이 전송되기도 한다. 그리고 거리가 멀어진다는 것은 헬멧을 버렸거나, 헬멧이 손상됐거나 헬멧을 잃어버린 경우일 것인데 이때는 소비자가 수수료를 부과하게 된다. 처음부터 헬멧에 문제가 있었거나 특정 사유가 있다면 Report를 통하여 소비자는 사후 조치를 받을 수 있는 시스템으로 킥보드 사용자의 헬멧 문제를 해결해 보려고 한다.

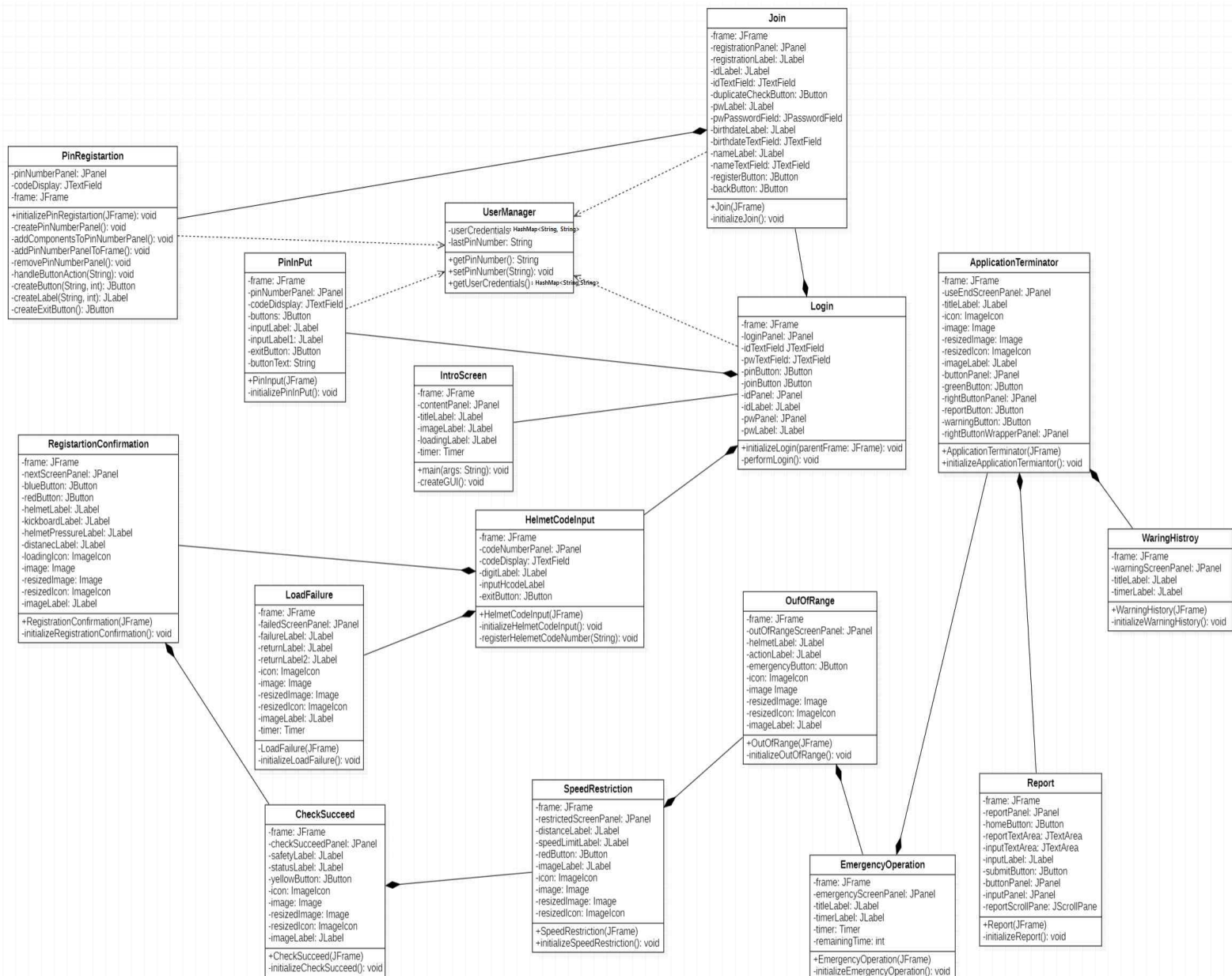
### 2) Features of project

Helemet Safer는 **사용자들이 안전하게 공유 킥보드를 사용할 수 있도록 돕는 시스템**이다. 또한 헬멧 안에 소형 IOT 칩이 삽입되어 있기 때문에 생산자들이 헬멧 회수나, 관리를 편하게 할 수 있도록 돕는다. 헬멧 파손이나 분실 시에는 사용자는 수수료를 부과하게 된다. 그러나 처음부터 헬멧에 문제가 있었거나 특정 사유가 있다면 Report를 통하여 소비자는 사후 조치를 받을 수 있는 시스템으로 킥보드 사용자의 헬멧 문제를 해결해 보려고 한다.

클래스들 간의 객체화를 최적화해서 함수로의 접근은 어느 정도 가능하지만 공유되는

변수는 거의 없도록 설정해야 할 것이다. 키보드를 사용하는 중 다른 곳에서의 접근이 들어오는 것을 제한시켜야 한다. private 변수 적절히 사용. 또한 하드웨어적 요소가 많이 첨가된 아이디어를 자바 GUI로 표현하게 되면서 화면을 분리하기 위해 클래스에 기능을 조금 구체적으로 쓰게 됐다.

## 2. Class diagram



### 1)IntroScreen

#### Attributes

- frame: JFrame 메인 프레임을 나타내는 JFrame 객체.
- contentPanel: JPanel 콘텐츠 패널을 나타내는 JPanel 개체
- titleLabel: JLabel 제목 레이블을 나타내는 JLabel 개체
- imageLabel: JLabel 이미지 레이블을 나타내는 JLabel 개체
- loadingLabel: JLabel 로딩 레이블을 나타내는 JLabel 개체
- timer: Timer 로딩 지연에 사용되는 Timer 개체

#### Methods

- + main(args: String[]): void: 애플리케이션의 진입점. IntroScreen 인스턴스를 생성하고 createGUI 메서드를 호출
- createGUI(): void: 프레임, 레이블 및 타이머를 포함한 GUI 구성요소를 생성하고 구성. 레이아웃을 설정하고 패널에 구성 요소를 추가하며 프레임을 표시. 지연 후 '로그인' 초기화 프로세스를 trigger. 초기 화면을 구성하는데 사용

## 2) Login

#### Attributes

- frame: Frame 상위 프레임을 나타내는 JFrame 개체
- loginPanel: JPanel 로그인 패널
- idTextField: JTextField 사용자 ID 입력 칸
- pwTextField: JTextField 사용자 PW 입력 칸
- pinButton: JButton PIN 버튼
- joinButton: JButton Join 버튼
- idPanel: JPanel ID 패널 나타내는 개체
- idLabel: JLabel ID 레이블 나타내는 개체
- pwPanel: JPanel PW 패널 나타내는 개체
- pwLabel: JLabel PW 레이블 나타내는 개체

#### Methods

- + initializeLogin(parentFrame: JFrame): void : Login 개체 초기화하고 로그인 프로세스 수행되는데 사용
- Login(JFrame parentFrame): 로그인 생성자
- performLogin(): void: 로그인 패널 설정, 구성 요소 추가 및 로그인 작업 처리

이 이후로 부터는 프레임과 라벨에 대한 설명은 중복될 거 같아 핵심 요소들만 추가했습니다.

## 3)Join

#### Attributes

- frame: JFrame

- registrationPanel: JPanel
- registrationLabel: JLabel
- idLabel: JLabel
- idTextField: JTextField ID 입력 칸
- duplicateCheckButton: JButton 중복 확인 버튼
- pwLabel: JLabel
- pwPasswordField: JPasswordField PW입력 칸
- birthdateLabel: JLabel
- birthdateTextField: JTextField 생년월일 입력 버튼
- nameLabel: JLabel
- nameTextField: JTextField 이름 입력 칸
- registerButton: JButton 등록 버튼
- backButton: JButton 뒤로가기 버튼

#### Methods

- initializeJoin(): void : Login 개체 초기화하고 로그인 프로세스 수행되는데 사용
  - + Join(JFrame): Join 개체 초기화하고 회원가입을 하는데 사용
- 빈칸 여부를 확인하고, UserManager클래스를 사용해서 아이디가 중복인지 검사

## 4)PinRegistration

#### Attributes

- pinNumberPanel: JPanel
- codeDisplay: JTextField
- frame: JFrame

#### Methods

- + initializePinRegistartion(in JFrame): void PinRegistration 개체 초기화. 화면 전환시 사용
- createPinNumberPanel(): void 핀 넘버를 만든다
- addComponentsToPinNumberPanel(): void 핀 넘버 패널에 요소들을 추가
- addPinNumberPanelToFrame(): void 핀 넘버 추가
- removePinNumberPanel(): void 핀 넘버 패널 삭제
- handleButtonAction(in String): void 버튼 액션 리스너 다룸
- createButton(in String, in int): JButton 핀 버튼들 생성
- createLabel(in String, in int): JLabel 라벨 생성
- createExitButton(): JButton eixt버튼 생성

4자리의 핀 번호를 등록하고 자리수가 틀리면 팝업을 띄운다.

## 5) PinInput

Attributes
<ul style="list-style-type: none"> <li>- codeDisplay: JTextField 코드를 보여주는 칸</li> <li>- buttons: JButton[] 키패드를 구성하는데 사용</li> <li>- exitButton: JButton 나가기 버튼</li> </ul>
Methods
<ul style="list-style-type: none"> <li>- initializePinInput(): void : PinInput 개체 초기화하고 PinInput 프로세스 수행되는데 사용. 화면이 넘어갈 때 사용하기 위해 만들었다</li> <li>+ PinInput(JFrame): PinInput 개체 초기화하고 핀 입력 받는 데 사용</li> </ul>
키패드를 생성해서 4자리의 핀번호를 입력받고 설정된 핀 번호가 맞는지 확인한다.

## 6) UserManager

Attributes
<ul style="list-style-type: none"> <li>- userCredentials: HashMap&lt;String, String&gt; 유저의 id, 비밀번호 입력을 받을 해시맵</li> <li>- lastPinNumBer: String 사용자의 마지막 핀 번호 입력을 저장할 변수</li> </ul>
Methods
<ul style="list-style-type: none"> <li>+ setPinNumber(String): void 핀 넘버값 설정</li> <li>+ getPinNumber(): String 핀 넘버값 가져옴</li> <li>+ getUserCredentials(): HashMap&lt;String, String&gt; id,pw 값 가져옴</li> </ul>

## 7) HelmetCodeInput

Attributes
<ul style="list-style-type: none"> <li>- codeDisplay: JTextField 헬멧 코드 입력 칸</li> <li>- codeNumberPanel: JPanel 헬멧 코드 패널</li> </ul>
Methods
<ul style="list-style-type: none"> <li>- registerHelemetCodeNumber(in String): void 헬멧 코드를 등록하는데 사용되는 함수 (하드웨어로 구현할 시 이때 전송)</li> <li>- initializeHelmetCodeInput(): void 헬멧 구성화면 초기화</li> <li>+ HelmetCodeInput(in JFrame) 헬멧코드 생성자 생성하는데 사용</li> </ul>

## 8) LoadFailure

Attributes
<ul style="list-style-type: none"> <li>- failureLabel: JLabel 실패 나타낼 패널</li> <li>- returnLabel: JLabel “정보 잘못 입력” 출력할 라벨</li> </ul>



- returnLabel2: JLabel “헬멧 코드 창으로 돌아갑니다” 출력할 라벨
- timer: Timer 시간 설정
Methods
- LoadFailure(in JFrame)
- initializeLoadFailure(): void 로그인 패널 설정, 구성 요소 추가 및 로그인 작업 처리
정보 잘못 입력 문구를 출력하고 설정된 시간이 다 되면 헬멧 코드 창으로 돌아가는 생성자와 함수

## 9) RegistartionConfirmation

Attributes
- blueButton: JButton 하드웨어적으로 헬멧 압박이 인지 되고, 키보드 등록이 완료 되고, 헬멧 코드가 입력됐을 때를 가정하는 버튼
- RedButton: JButton 등록 실패를 가정하는 버튼
Methods
+ RegistrationConfirmation(in JFrame) : RegistartionConfirmation 개체 초기화 하고 내부 동작들(버튼 처리) 수행되는데 사용
- initializeRegistrationConfirmation(): void 버튼들에 대한 작업 처리

## 10) CheckSucceed

Attributes
- safetyLabel: JLabel 안전 운전 문구 출력시 사용
- yellowButton: JButton 키보드와 헬멧의 거리가 멀어졌다는 것을 가정하는 버튼
Methods
+ CheckSucceed(in JFrame)
- initializeCheckSucceed(): void
문구를 띄우고 버튼 클릭에 따른 이벤트를 처리하는 생성자와 함수

## 11) SpeedRestriction

Attributes
- distanceLabel: JLabel 거리가 멀어짐을 나타냄
- speedLimitLabel: JLabel 속도 제한이 걸린다는 것을 나타냄
- redButton: JButton 헬멧과 키보드 사이의 거리가 일정 거리 이상으로 떨어진 것으로 간주하는 버튼
Methods
+ initializeSpeedRestriction(): void

- SpeedRestriction(in JFrame)

헬멧과 키보드가 거리가 떨어진것을 나타내고 빨간버튼을 처리해주는 함수와 생성자  
다른 함수에서 부를 수 있게 initializeSpeedRestriction() 함수는 public으로 설정

## 12) OutOfRange

### Attributes

- helmetLabel: JLabel 헬멧이 범위 안에 없다는 것을 알림
- actionLabel: JLabel 곧 작동을 멈춘다는 것을 알림
- emergencyButton: JButton “거리 멀어짐” 화면에서 일정 거리의 기준치를 “초과”한 것으로 간주하는 버튼

### Methods

+ OutOfRange(in JFrame)

- initializeOutOfRange(): void

헬멧이 범위 안에 없다는 것을 알리고 비상 작동을 활성화시키는 생성자와 함수

## 13) EmergencyOperation

### Attributes

- titleLabel: JLabel 비상 작동 중 문구 띄움
- timerLabel: JLabel 남은 시간 3:00부터 줄이면서 띄움
- timer: Timer 경고문을 띄우고 3초 후에 종료 창으로 넘어가기 위해 시간 설정

### Methods

+ EmergencyOperation(in JFrame)

- initializeEmergencyOperation(): void

비상작동 문구를 띄우고 3초 후에 종료 창으로 넘어가기 위해 필요한 생성자와 함수

## 14) ApplicationTerminator

### Attributes

- greenButton: JButton 홈 화면으로 돌아가기 위한 버튼
- rightButtonPanel: JPanel 오른쪽 패널에 report와 warning버튼을 배치하기 위한 패널
- reportButton: JButton 리포트 버튼.
- warningButton: JButton warning 버튼.
- rightButtonWrapperPanel: JPanel

### Methods

+ ApplicationTerminator(in JFrame)

- initializeApplicationTerminator(): void 키보드 사용 종료를 알리고 홈 화면으로 돌아가기 위한 버튼, report 버튼, warning 버튼을 활성화 시키기 위한 생성자와 함수
--

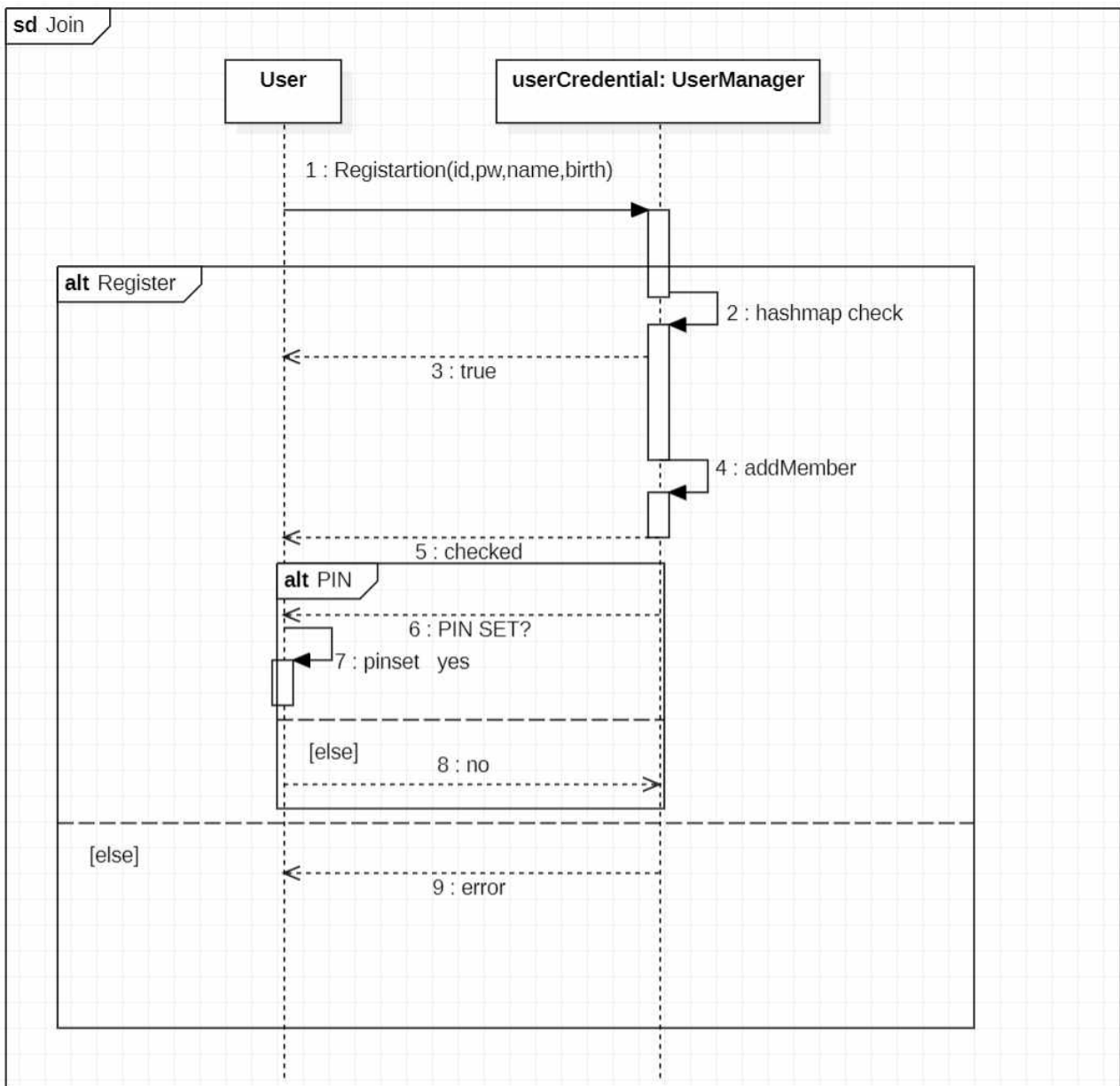
## 15) Report

Attributes
- homeButton: JButton 홈 버튼 - reportTextArea: JTextArea 리포트 작성 란 - submitButton: JButton 제출 버튼 - reportScrollPane: JScrollPane 리포트 내용 스크롤
Methods
+ Report(in JFrame) - initializeReport(): void 이 생성자와 함수를 사용하여 사용자는 자신의 요청사항을 제출하고 홈 화면으로 돌아갈 수 있다.

## 16) WarningHistory

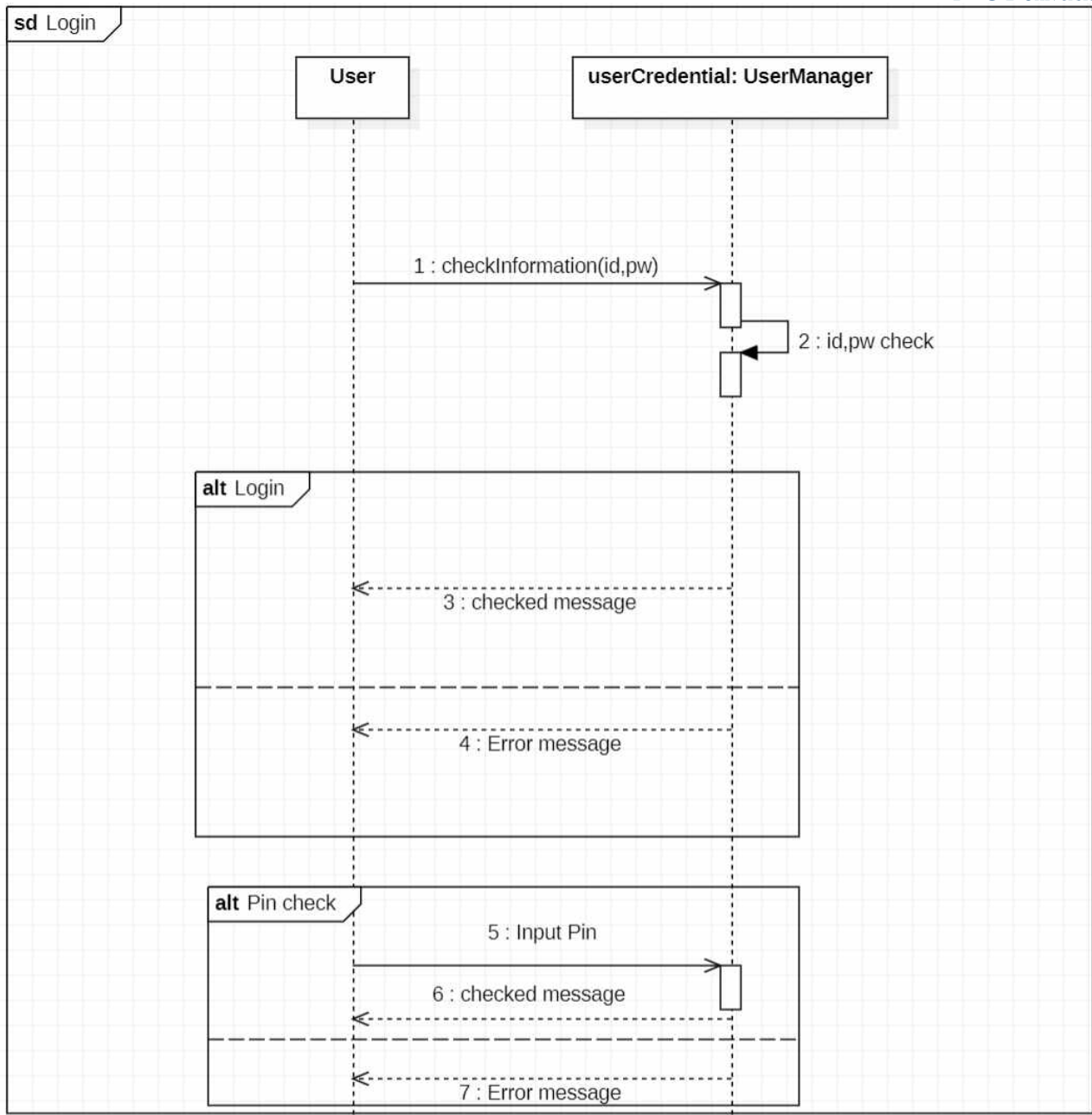
Attributes
warningScreenPanel: JPanel 경고 누적 횟수를 보여주기 위한 패널 timerLabel: JLabel 시간 설정
Methods
+WarningHistory(in JFrame) - initializeWarningHistory(): void 생성자와 함수를 사용하여 사용자의 누적 경고 횟수를 보여주고, 홈 화면으로 돌아간다

### 3. Sequence diagram

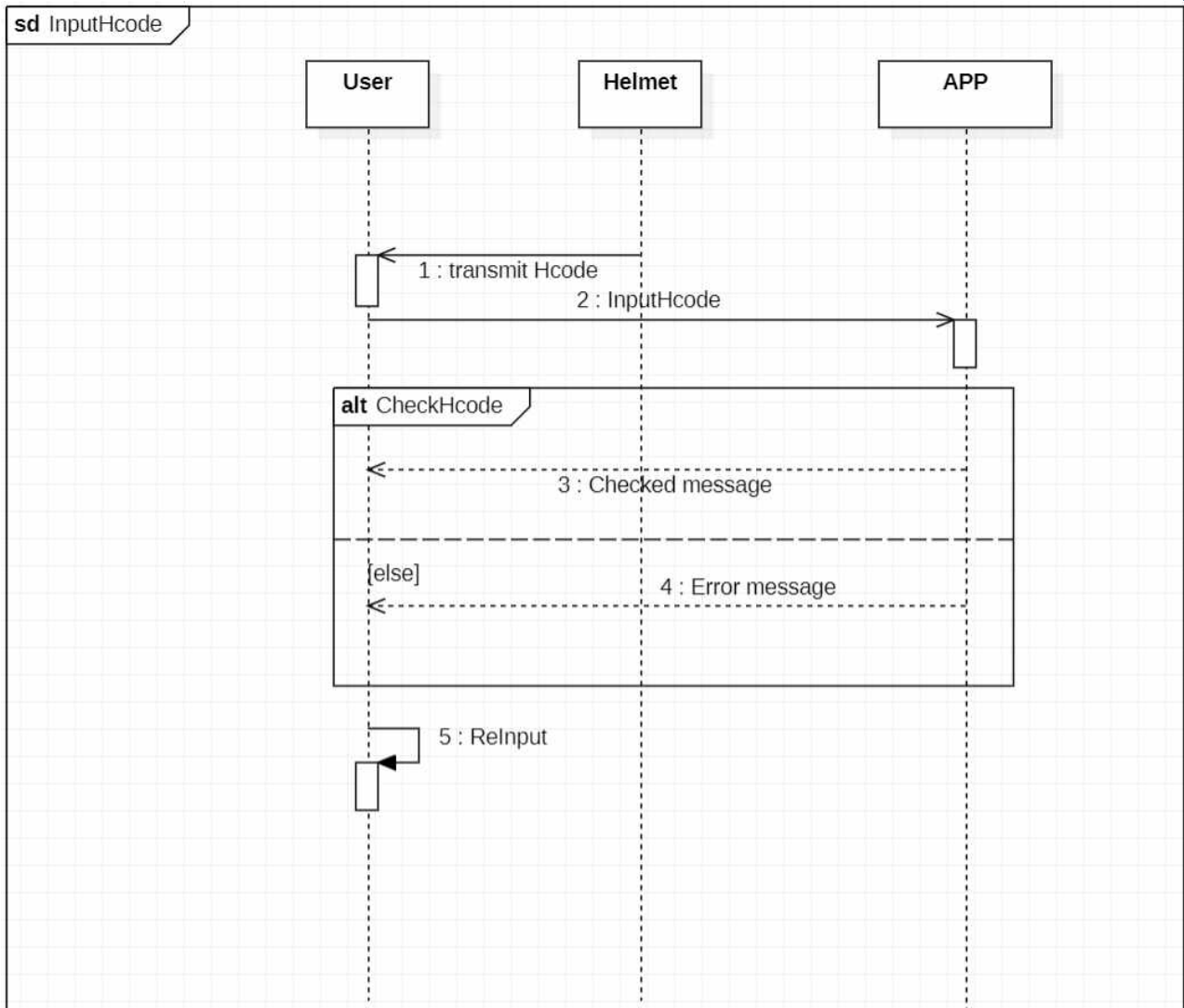


회원가입 시 UserManager 클래스의 userCredential 해시 맵에 등록을 요청해서 없는 사용자 정보면 아이디를 추가하고 등록 확인 메시지를 보내준다.

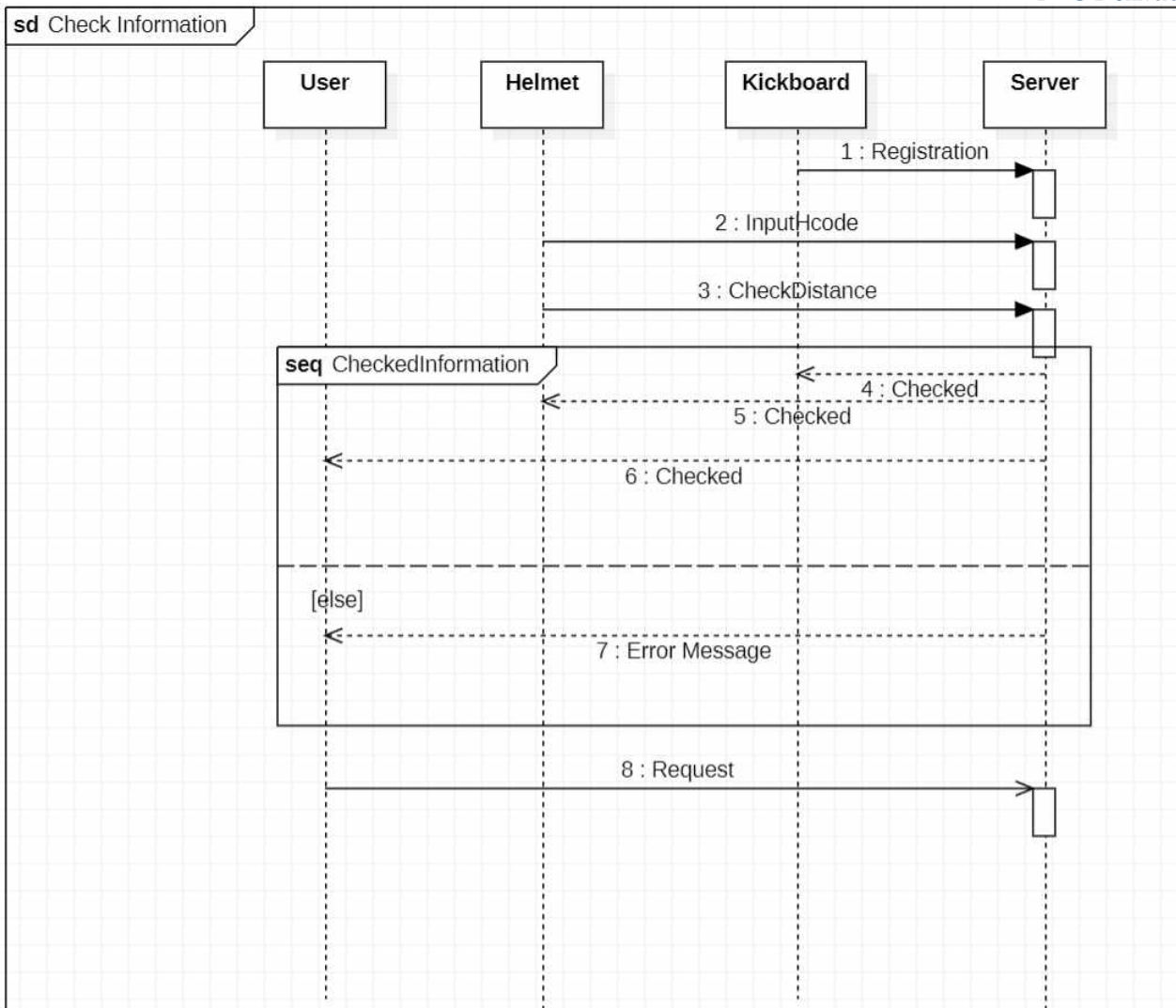
그 이후 핀 번호 설정을 하게 되고, 에러 메시지를 받으면 팝업을 띄워준다.



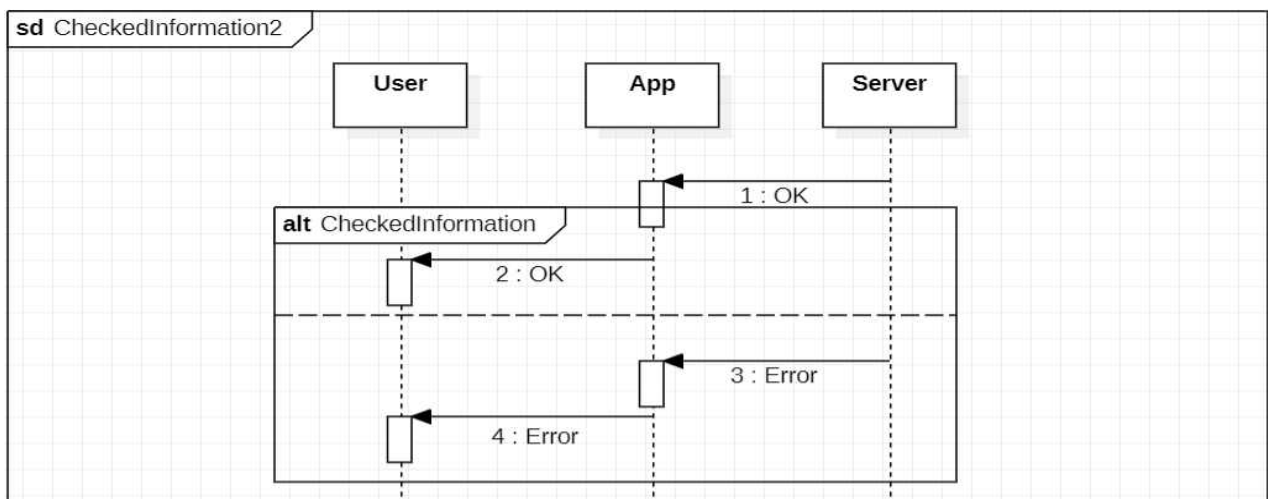
ID, PW 값을 입력해서 정보가 해시 맵에 있으면 로그인을 성공하고, 없으면 에러 메시지를 출력한다. 그 이후 핀 번호 입력을 통해서도 로그인이 가능하다.



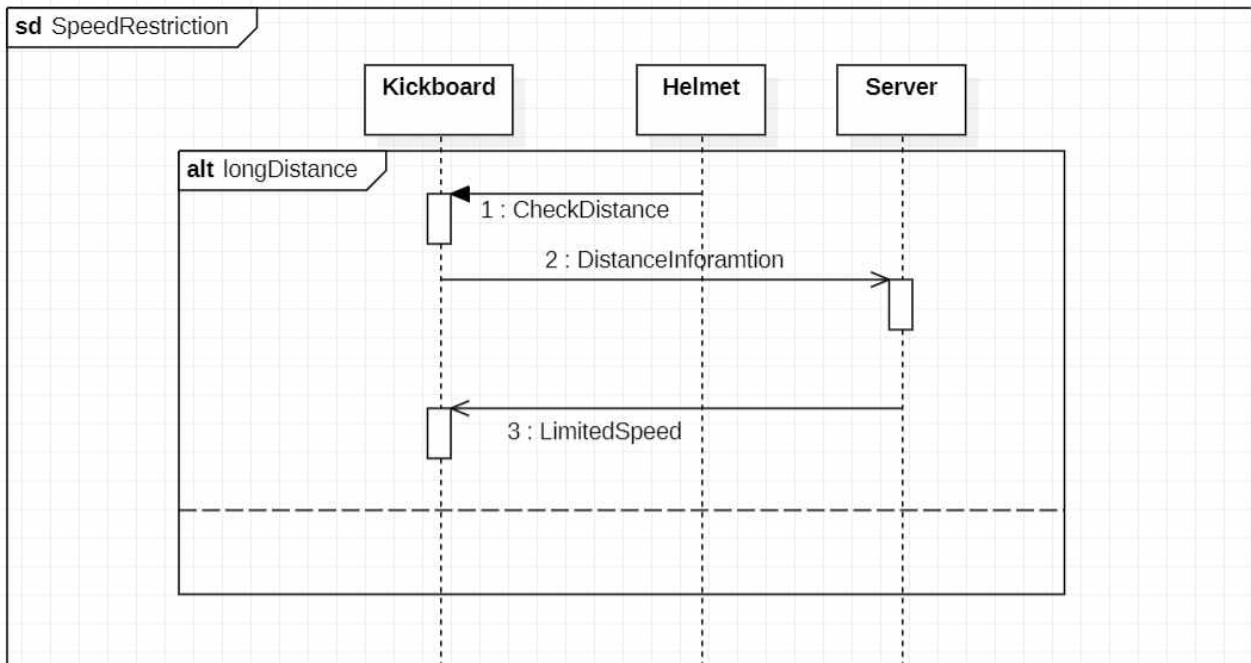
이후 헬멧으로부터 헬멧 코드를 입력받게 돼서 입력했을 시 헬멧 코드가 유효하면 다음 단계로 넘어가고, 유효하지 않다면 에러 메시지를 보내 재 입력받게 한다.



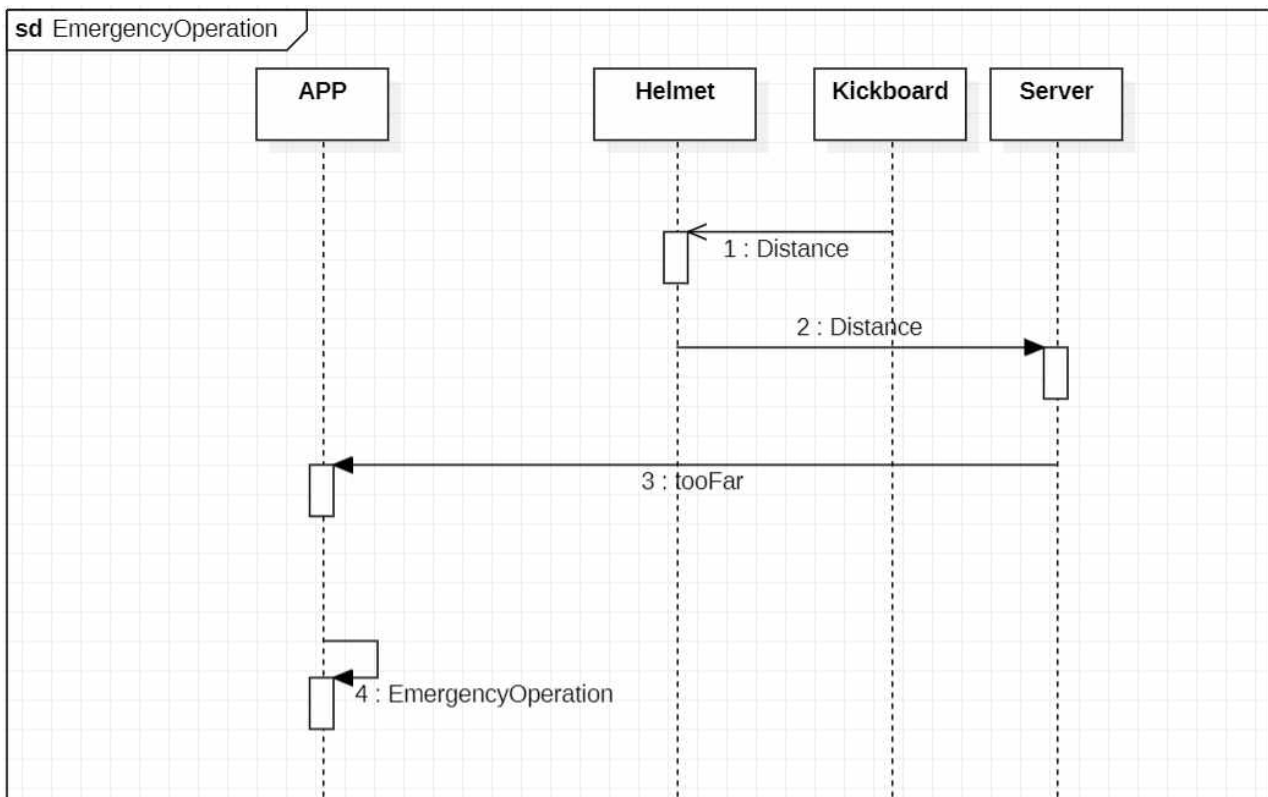
킥보드는 서버에 등록 요청, 헬멧은 입력받은 Hcode 정보를 전송하고 거리 정보를 전달한다 만약 서버에서 이 정보들이 확인이 될시 User는 다음 단계를 요청해서 수행할 수 있고, 그게 아닐시 에러 메시지를 받아 재 입력 요청을 해야한다.



만약 서버로부터 확인 신호가 들어오면 앱은 User에게 OK 신호를 보내고 그게 아닐 경우 에러 메시지를 보낸다.

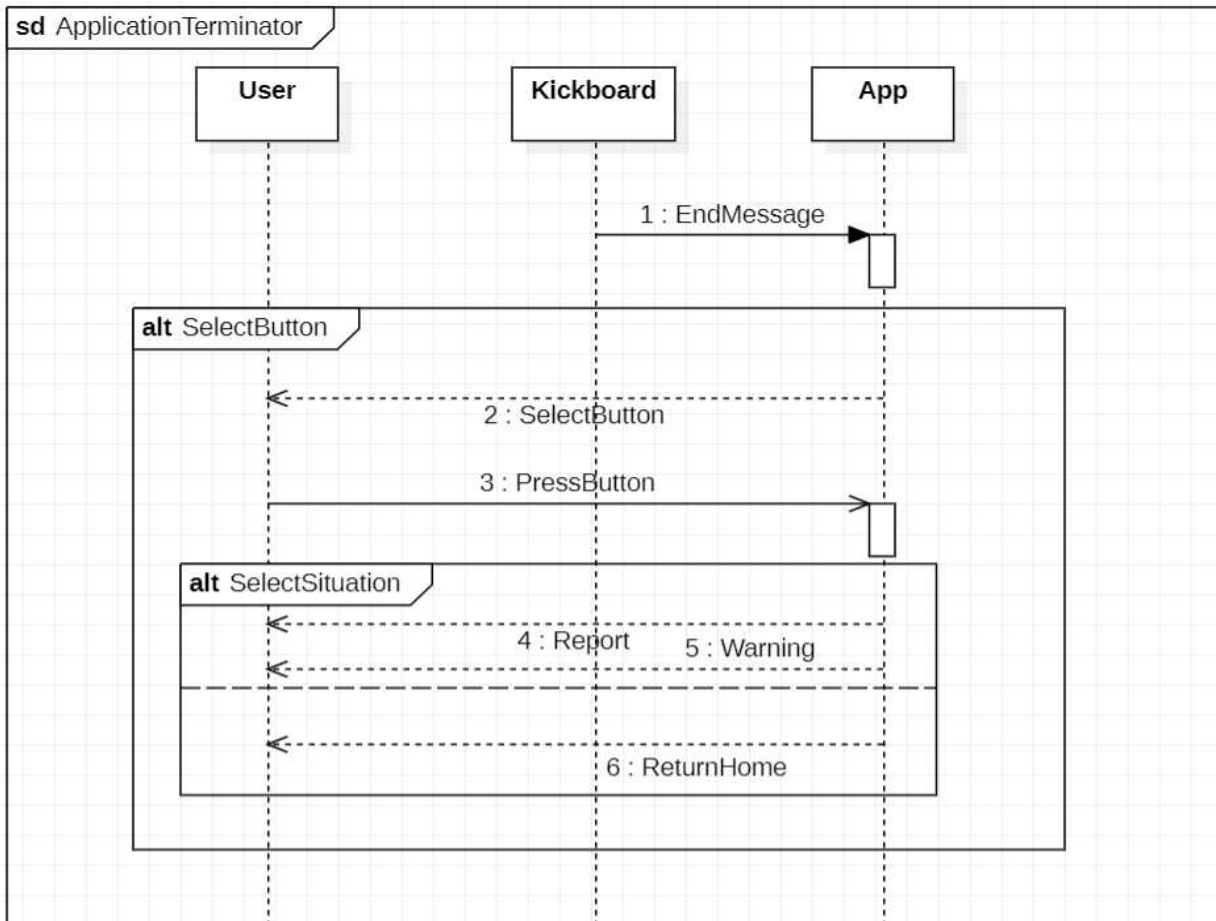


헬멧이 킥보드한테 거리 정보를 요청하면 킥보드를 헬멧한테 거리정보를 주고 이를 받아 서버한테 준다 이후 일정 거리가 이상이 된 것을 감지하면 속도제한 신호를 Kickboard에게 넘겨준다.

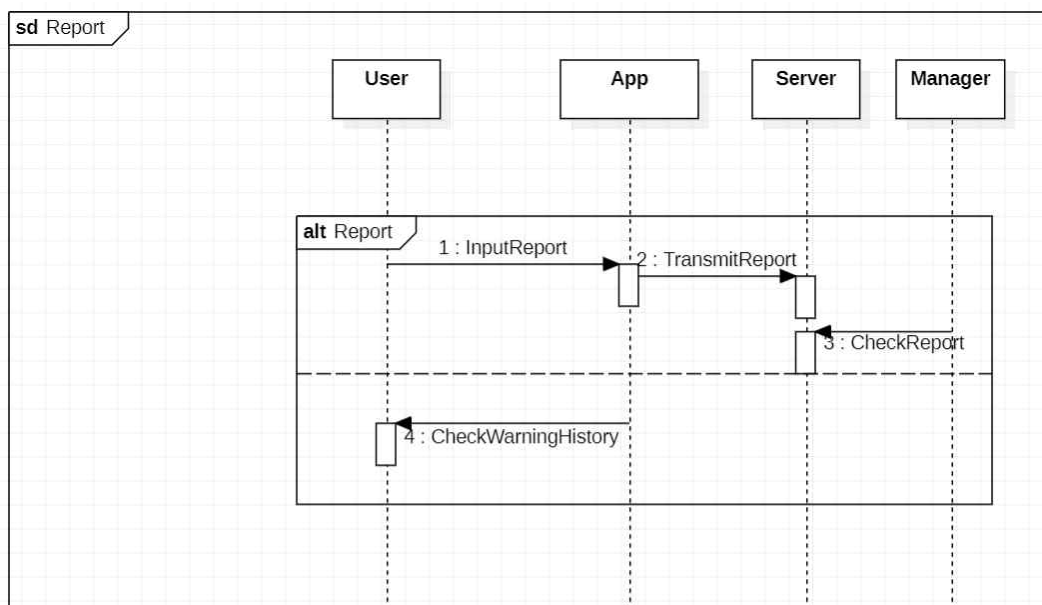


킥보드가 헬멧한테 거리 정보를 주고, 헬멧이 서버한테 거리 정보를 공유했을 때 거리가 너무 멀면 서버는 앱한테 너무 멀다는 신호를 전송한다. 그러면 앱은 비상작동 버튼을 활성화시킨다.





킥보드가 서버를 통해 앱에게 종료 메시지를 보낸다 그러면 앱은 버튼을 고르라는 정보를 사용자에게 전달하고 사용자는 버튼 입력을 해서 Report 작성할 것인지, 경고 누적 횟수를 볼 것인지, 홈 화면으로 돌아갈 것인지 정한다

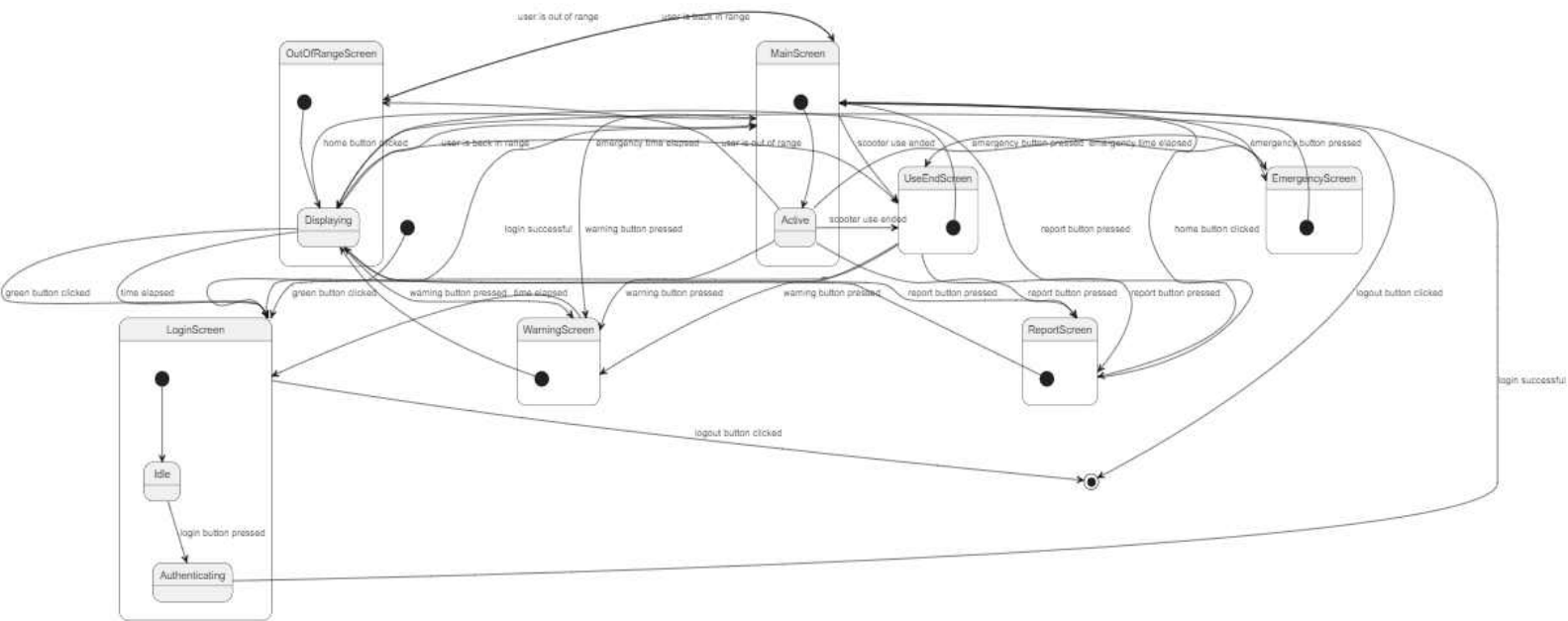


리포트 버튼을 누르고 유저가 리포트 내용을 앱에 쓰게 되면 서버는 이를 받아들이고 매니저는 서버에 접속해서 이 내용을 확인할 수 있다. 만약 리포트 버튼을 누르지 않는다

면 경고 누적 횟수를 보여준다.

## 4. State machine diagram

아래는 'Helmet Safer'의 State machine Diagram을 구현한 그림이다.



## 5. Implementation requirements

운영 체제: Windows 10 Home 64비트 (10.0, 빌드 19044)

언어: 한국어 (국가별 설정: 한국어)

시스템 제조업체: LG Electronics

시스템 모델: 17Z90N-VA76K

BIOS: C2ZE0250 X64

프로세서: Intel(R) Core(TM) i7-1065G7 CPU @ 1.30GHz (8 CPUs), ~1.5GHz

메모리: 16384MB RAM

페이지 파일: 22081MB 사용됨, 10396MB 사용 가능

DirectX 버전: DirectX 12

디바이스

이름: Intel(R) Iris(R) Plus Graphics

제조업체: Intel Corporation

칩 유형: Intel(R) Iris(R) Graphics Family

DAC 유형: Internal

디바이스 유형: 전체 디스플레이 장치

전체 메모리 근사값: 8174 MB

디스플레이 메모리(VRAM): 128 MB

공유 메모리: 8046 MB

현재 디스플레이 모드: 2560 x 1600 (32 bit) (60Hz)

모니터: Generic PnP Monitor

이 노트북 환경에서 아무런 문제 없이 잘 동작하였다

소프트웨어 요구 사항으로는 윈도를 사용할 시 7 이상을 사용해야 할 것이고 exe 파일에 직접 접근하지 않으면 JDK를 사용하여 열어야 할 것이다. 하드웨어적 요구 사항으로는 RAM 1GB 이상, HDD / SSD : 10 GB 이상 CPU는 Intel Pentium IV 이상 정도만 되면 큰 무리 없이 동작할 것이다.

## 6. Glossary

Terms	Descriptions
애용	좋아하여 애착을 가지고 자주 사용함.
활성화	사회나 조직 등의 기능이 활발함. 또는 그러한 기능을 활발하게 함.
방안	일을 처리하거나 해결하여 나갈 방법이나 계획.
OTP	사용자 인증을 위하여 로그인할 때마다 매번 새로 생성되는 비밀번호. 이때 비밀번호는 무작위로 생성되기 때문에 비교적 안전한 인증 방식의 하나이다.
State machine diagram	객체 lifetime동안의 변환될 수 있는 모든 상태를 정의해 둔 diagram
report	보고하다.(영단어 뜻) / 조사나 연구, 실험 따위의 결과에 관한 글이나 문서
warning	경고, 주의
initialize	초기 내용을 설정하다

## 7. References

- 강의자료 : Structural Modeling I,,II, Behavior Modeling I, II
- 로고의 헬멧과 날개, Interface prototype의 그림 <https://www.flaticon.com/kr/>
- Glossary <https://ko.dict.naver.com/#/main>, <https://en.dict.naver.com/#/main>