##### Project Document

Usecase Specification Document

14 조

201602066 정수빈

201602033 윤혜원

201602044 이승재

지도교수: 김영국 교수님 (서명)

문서 버전 관리

|  |  |  |  |
| --- | --- | --- | --- |
| Rev# | Date | Affected Section | Author |
| 1 | 2019-04-01 | 유스케이스 명세서 1차 | 윤혜원 |
| 2 | 2019-04-04 | 유스케이스 명세서 2차 | 정수빈 |
| 3 | 2019-04-05 | 유스케이스 명세서 3차 | 이승재 |
|  |  |  |  |

목차

[1. 소개 6](#_Toc5283296)

[1.1. 목적 6](#_Toc5283297)

[2. 유스케이스 다이어그램 7](#_Toc5283298)

[2.1. User System 다이어그램 7](#_Toc5283299)

[2.2. Server System 다이어그램 8](#_Toc5283300)

[2.3. DoorLock System 다이어그램 9](#_Toc5283301)

[3. 유스케이스 설명서 10](#_Toc5283302)

[3.1. 회원가입 10](#_Toc5283303)

[3.1.1. 행위자 10](#_Toc5283304)

[3.1.2. 사전 조건 10](#_Toc5283305)

[3.1.3. 이벤트 흐름 10](#_Toc5283306)

[3.1.4. 후행 조건 11](#_Toc5283307)

[3.2. 로그인 12](#_Toc5283308)

[3.2.1. 행위자 12](#_Toc5283309)

[3.2.2. 사전조건 12](#_Toc5283310)

[3.2.3. 이벤트 흐름 12](#_Toc5283311)

[3.2.4. 후행 조건 12](#_Toc5283312)

[3.3. QR코드 발급 13](#_Toc5283313)

[3.3.1. 행위자 13](#_Toc5283314)

[3.3.2. 사전조건 13](#_Toc5283315)

[3.3.3. 이벤트 흐름 13](#_Toc5283316)

[3.3.4. 후행조건 13](#_Toc5283317)

[3.4. 경고알림 14](#_Toc5283318)

[3.4.1. 행위자 14](#_Toc5283319)

[3.4.2. 사전조건 14](#_Toc5283320)

[3.4.3. 이벤트 흐름 14](#_Toc5283321)

[3.4.4. 후행조건 14](#_Toc5283322)

[3.5. 공유하기 15](#_Toc5283323)

[3.5.1. 행위자 15](#_Toc5283324)

[3.5.2. 사전조건 15](#_Toc5283325)

[3.5.3. 이벤트 흐름 15](#_Toc5283326)

[3.5.4. 후행조건 15](#_Toc5283327)

[3.6. 위치확인 16](#_Toc5283328)

[3.6.1. 행위자 16](#_Toc5283329)

[3.6.2. 사전조건 16](#_Toc5283330)

[3.6.3. 이벤트 조건 16](#_Toc5283331)

[3.6.4. 후행조건 16](#_Toc5283332)

[3.7. 데이터 암호화 17](#_Toc5283333)

[3.7.1. 행위자 17](#_Toc5283334)

[3.7.2. 사전조건 17](#_Toc5283335)

[3.7.3. 이벤트흐름 17](#_Toc5283336)

[3.7.4. 후행조건 17](#_Toc5283337)

[3.8. 잠금장치제어 18](#_Toc5283338)

[3.8.1. 행위자 18](#_Toc5283339)

[3.8.2. 사전조건 18](#_Toc5283340)

[3.8.3. 이벤트흐름 18](#_Toc5283341)

[3.8.4. 후행조건 18](#_Toc5283342)

[3.9. 도어락 QR코드 데이터관리 19](#_Toc5283343)

[3.9.1. 행위자 19](#_Toc5283344)

[3.9.2. 사전조건 19](#_Toc5283345)

[3.9.3. 이벤트흐름 19](#_Toc5283346)

[3.9.4. 후행조건 19](#_Toc5283347)

[3.10. QR코드인식 20](#_Toc5283348)

[3.10.1. 행위자 20](#_Toc5283349)

[3.10.2. 사전조건 20](#_Toc5283350)

[3.10.3. 이벤트흐름 20](#_Toc5283351)

[3.10.4. 후행조건 20](#_Toc5283352)

[3.11. 도어락등록 21](#_Toc5283353)

[3.11.1. 행위자 21](#_Toc5283354)

[3.11.2. 사전조건 21](#_Toc5283355)

[3.11.3. 이벤트흐름 21](#_Toc5283356)

[3.11.4. 후행조건 21](#_Toc5283357)

그림 목차

FIGURE 1 – USE-CASE DIAGRAM OF USER SYSTEM …………………………………………………………………7

FIGURE 2 – USE-CASE DIAGRAM OF SERVER SYSTEM …………………………………………………………….8

FIGURE 3 – USE-CASE DIAGRAM OF DOORLOCK SYSTEM ………………………………………………………9

# 소개

## 목적

이 문서는 카카오톡과 유사한 기능을 하는 채팅 프로그램의 기능을 명세하고 있다. 요구사항을 상세하게 표현하기 위한 유스케이스 다이어그램과 각 유스케이스에 대한 명세서를 포함하고 있다.

# 유스케이스 다이어그램

## User System 다이어그램

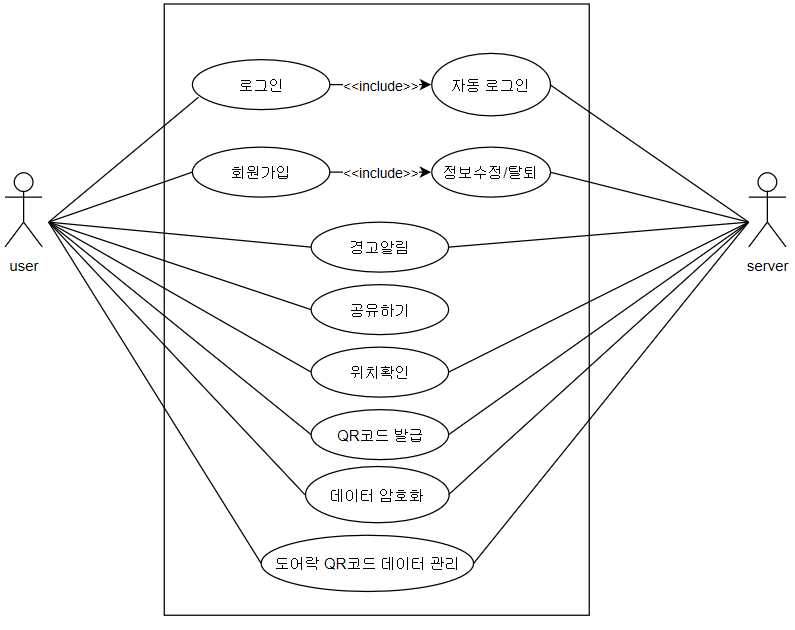
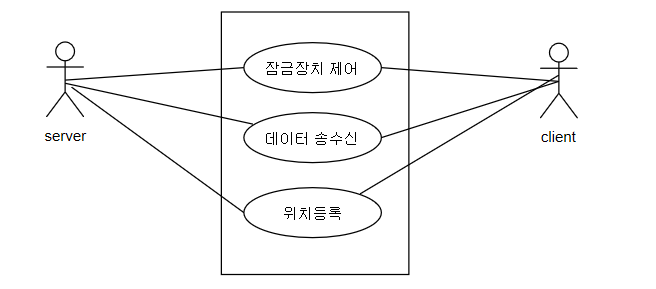


Figure 1 – Use-Case Diagram of User System

## Server System 다이어그램



**Figure 2 – Use-Case Diagram of Server System**

## DoorLock System 다이어그램

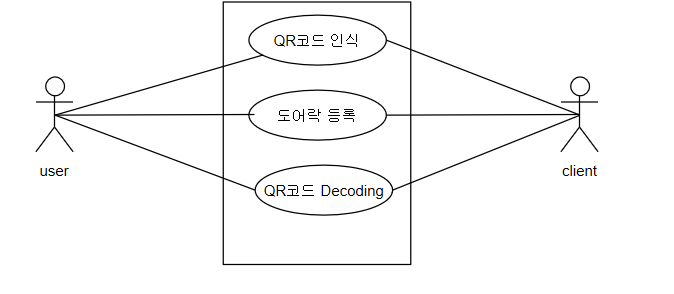


Figure 3 – Use-Case Diagram of DoorLock System

# 유스케이스 설명서

## 회원가입

### 행위자

User (Initiator), Server( supporter)

### 사전 조건

* 서버에 사용자 정보가 저장될 데이터베이스가 구축되어 있어야 한다.

### 이벤트 흐름

#### 진입 조건

* 시작화면에서 회원가입 화면으로 이동한다.

#### 기본흐름

1. 사용자는 서버에 인증번호를 요청한다.
2. 서버는 사용자에게 SMS로 인증번호를 전송한다.
3. 사용자는 전송된 인증 번호를 사용자 인증 페이지에 입력하고 인증을 요청한다.
4. 서버는 사용자를 확인하고, 사용자를 등록한다.
5. 서버는 사용자 등록을 완료한 뒤, 사용자에게 이름을 요청한다.
6. 사용자는 자신의 이름과 메일, 아이디와 비밀번호를 입력하고, 회원가입을 요청한다.
7. 서버는 사용자의 정보를 저장한 뒤, 등록에 성공하였음을 알린다.

#### 종료 조건

* 회원가입 완료 시 메인 화면으로 이동한다.

#### 대체 조건

* 중복된 사용자일 경우 회원가입을 할 수 없다,

### 후행 조건

* 서버에 사용자의 데이터가 성공적으로 저장된다.

## 로그인

### 행위자

User (Initiator), Server(supporter)

### 사전조건

* 서버에 로그인 하고자 하는 사용자의 정보가 저장되어 있어야 한다.

### 이벤트 흐름

#### 진입 조건

* Application을 실행한다.

#### 기본흐름

1. 사용자는 아이디와 비밀번호를 입력한다.
2. 로그인 버튼을 눌러 입력한 아이디와 비밀번호를 서버에 전송한다.
3. 서버는 사용자를 확인하고 로그인 처리한다.
4. 로그인에 성공할 경우, 자동로그인 체크박스가 되어있다면 sharedpreference으로 로그인 정보를 저장한다.

#### 종료 조건

* Activity를 종료한다.

#### 대체 흐름

* 서버에 저장되어 있지 않은 정보를 가진 사용자의 경우, 유저 정보가 존재하지 않는다는 알림창이 뜬다.
* sharedpreference에 자동로그인 정보가 있으면, 어플리케이션 시작시 저장된 값으로 로그인한다.

### 후행 조건

* 로그인을 완료하고 메인화면으로 이동한다.

## QR코드 발급

### 행위자

User(initiator), Server(supporter)

### 사전조건

* 도어락이 등록되어 있어야 한다.

### 이벤트 흐름

#### 진입 조건

* 사용자는 아이디와 비밀번호를 통해 서버에 로그인을 한다.

#### 기본 흐름

1. 도어락 목록에 등록된 도어락 중 잠금 해제를 원하는 도어락을 선택한다.
2. 선택된 도어락의 QR코드 발급 버튼을 누른다.
3. 사용자는 was에 QR코드 발급을 요청한다.
4. Was는 데이터를 암호화, json형태로 어플리케이션에 보내준다.
5. 어플리케이션은 json을 파싱한다.
6. 파싱된 데이터를 QR Code Generator api를 이용해서 QR코드형태로 보여준다.

#### 종료 조건

* 화면을 이동하거나 어플을 종료시킨다.

#### 대체 흐름

* 요청을 받을 수 없으면 에러를 출력한다.

### 후행조건

* Application 상의 QR코드를 출력해 준다.

## 경고알림

### 행위자

User(supporters), server(initiator)

### 사전조건

* 사용자가 잘못된 QR코드를 2회 이상 입력했었다.

### 이벤트 흐름

#### 진입 조건

* 사용자가 발급받은 QR코드를 도어락 카메라에 인식시킨다.

#### 기본 흐름

1. Client는 decoding한 data를 was에 보낸다
2. Was는 올바른 값인지 판단하고, 잘못된 값을 3회 이상 입력했는지 판단한다.
3. 도어락에 등록된 사용자들에게 경고를 보낸다
4. Fcm을 통해 푸쉬 알람을 보낸다.

#### 종료 조건

#### 대체 흐름

### 후행ㄴ조건

* 어플리케이션은 푸쉬알람을 통해 경고 알림을 받는다.

## 공유하기

### 행위자

User(initiator)

### 사전조건

* 사용자는 로그인이 되어있어야 한다.
* 받아야 될 대상 정보를 알아야한다..

### 이벤트 흐름

#### 진입 조건

* 사용자는 잠금 해제를 원하는 도어락을 선택한 후, 공유하기 버튼을 누른다.

#### 기본 흐름

1. 공유를 원하는 대상의 이름과 전화번호를 입력한다.
2. 공유 버튼을 통해 원하는 공유 방법과 공유 대상을 선택한다.
3. 선택한 공유 방법을 통해 공유 대상에게 이미지 파일로 QR코드가 전송된다.

#### 종료 조건

* 종료되어 원래의 액티비티로 돌아온다.

#### 대체 흐름

### 후행조건

* 공유 받을 대상에게 QR코드가 전송된다.

## 위치확인

### 행위자

User(initiator), Server(supporter)

### 사전조건

* 서버에 사용자와 도어락이 등록되어 있어야 하고, 사용자는 로그인을 완료해아 한다.

### 이벤트 조건

#### 진입 조건

* 위치확인 할 도어락을 선택한다.

#### 기본 흐름

1. Was에 저장된 도어락의 위치를 받아온다.
2. 어플리케이션은 서버에서 받아온 위치를 보여준다.

#### 종료 조건

* 위치 확인이 완료되면 메인화면으로 이동한다.

#### 대체 흐름

* 잘못된 위치정보를 받아올경우 에러 메시지를 출력한다.

### 후행조건

* 어플리케이션에 도어락의 위치를 보여준다.

## 데이터 암호화

### 행위자

Server(initiator), User(supporter)

### 사전조건

* 발급을 받을 도어락에 사용자가 등록되어 있어야 한다.

### 이벤트흐름

#### 진입 조건

* 어플리케이션으로부터 도어락 키를 요청받는다.

#### 기본 흐름

1. 도어락 키(data)를 생성한다.
2. 평문 블록을 블록 암호 알고리즘으로 암호화한다.

#### 종료 조건

* json형태로 암호화된 data를 출력한다.

#### 대체 흐름

* 트렌잭션에 문제가 생길 경우 에러메세지를 출력하고 rollback한다.

### 후행조건

* 서버에 암호화된 데이터가 저장된다.

## 잠금장치제어

### 행위자

Server(initiator), Client(supporter)

### 사전조건

* 발급된 QR코드에 대한 데이터가 서버에 저장되어 있어야한다.

### 이벤트흐름

#### 진입 조건

* 카메라에 QR코드를 인식시킨다.

#### 기본 흐름

1. Was에 QR에서 decode된 데이터를 전송한다.
2. Was는 암호화 블록을 블록 암호 알고리즘으로 복호화한다.
3. 암호화된 데이터와 복호화된 데이터를 비교한다.

#### 종료 조건

* 문이 열린 경우 종료된다.

#### 대체 흐름

* N회 이상 잘못 입력 시 경고 알람을 띄워준다.

### 후행조건

* 도어락의 잠금이 해제된다.

## 도어락 QR코드 데이터관리

### 행위자

Server(initiator), User(supporter)

### 사전조건

* QR코드가 발급되어야 한다.

### 이벤트흐름

#### 진입 조건

* QR코드 데이터 관리 쓰레드를 호출한다.

#### 기본 흐름

1. 데이터 베이스에 접근한다.
2. 서버는 일정 시간이 지나면 QR코드데이터를 삭제한다.

#### 종료 조건

* 쓰레드가 종료된다.

#### 대체 흐름

* 잠금이 해제된 경우엔 쓰레드에 상관 없이 QR코드 정보를 삭제한다.

### 후행조건

* 사용자 입출입 기록이 저장된다.

## QR코드인식

### 행위자

User(initiator), Client(supporter)

### 사전조건

* Application에 사용자에게 발급된 QR코드가 존재해야 한다.

### 이벤트흐름

#### 진입 조건

* 사용자가 카메라에 QR코드를 인식시킨다.

#### 기본 흐름

1. OpenCV로 실시간 이미지를 프로세싱한다.
2. Zxing Library를 이용하여 QR코드를 Decoding한다.
3. was에 docoding한 데이터를 전송한다.

#### 종료 조건

* 인식하려는 사용자가 없을 때 QR코드 인식이 종료된다.

#### 대체 흐름

* 잘못된 QR코드를 인식시켰을 경우 경고 알림을 보낸다.

### 후행조건

* 서버에 데이터를 보낸다.

## 도어락등록

### 행위자

User(initiator), Client(supporter)

### 사전조건

* 서버에 도어락과 사용자에 대한 정보가 저장되어 있어야 한다.

### 이벤트흐름

#### 진입 조건

* 도어락 추가 버튼을 클릭해서 액티비티를 이동한다.

#### 기본 흐름

1. 사용자로부터 등록하려는 도어락의 위치와 고유번호를 입력받는다.
2. Was에 데이터를 저장한다.
3. Application에 adaptor를 추가한다.

#### 종료 조건

* 도어락 등록 후 메인화면으로 이동한다.

#### 대체 흐름

* 이미 등록한 도어락, 고유번호가 입력되지 않은 도어락의 경우 에러 메시지를 출력해 준다.

### 후행조건

* 어플리케이션에서 QR코드를 발급할 수 있는 권한이 사용자에게 주어진다.