아파트 공동 출입문 출입관리 시스템

작성자 : 한민수

Revision History

|  |  |  |
| --- | --- | --- |
| Version | Date | Summary |
| 1 | 2022.06.20 | * Introduce 작성 * Project Overview 작성 * System Overview 작성 * Architectural Driver 작성 |
| 2 | 2022.07.10 | * Top Level Design Description 작성 |
| 3 |  | * Architecture Evaluation 작성 |

# Introduction

본 문서는 아파트 공동 출입문 출입관리 시스템 소프트웨어 설계 문서로, Architectural Driver, Top Level Design, Component Level Design, Architecture Evaluation을 포함한다.

# Project Overview

## Project Background

아파트 공동 출입문 시스템은 아파트 단지 내 거주자의 요구에 맞는 보안 솔루션을 제공함으로써 안전한 생활과 편리한 주거공간을 구현하기 위한 시스템이다.

최근 코로나19로 인하여 공동 출입문에서 비밀번호를 직접 손가락으로 입력하는데 불안감을 느낀 입주민들이 증가하였다. 이에 비접촉식 출입인증에 대한 수요가 발생하였으며,

또한 배달, 택배 등 비대면 전달방식이 증가함에 따라 출입인증 보안 강화를 원하는 입주민들도 증가하였다.

아파트 단지 내 효율적인 출입 제어를 통한 범죄예방을 위해 출입관리 시스템을 도입하고자 한다.

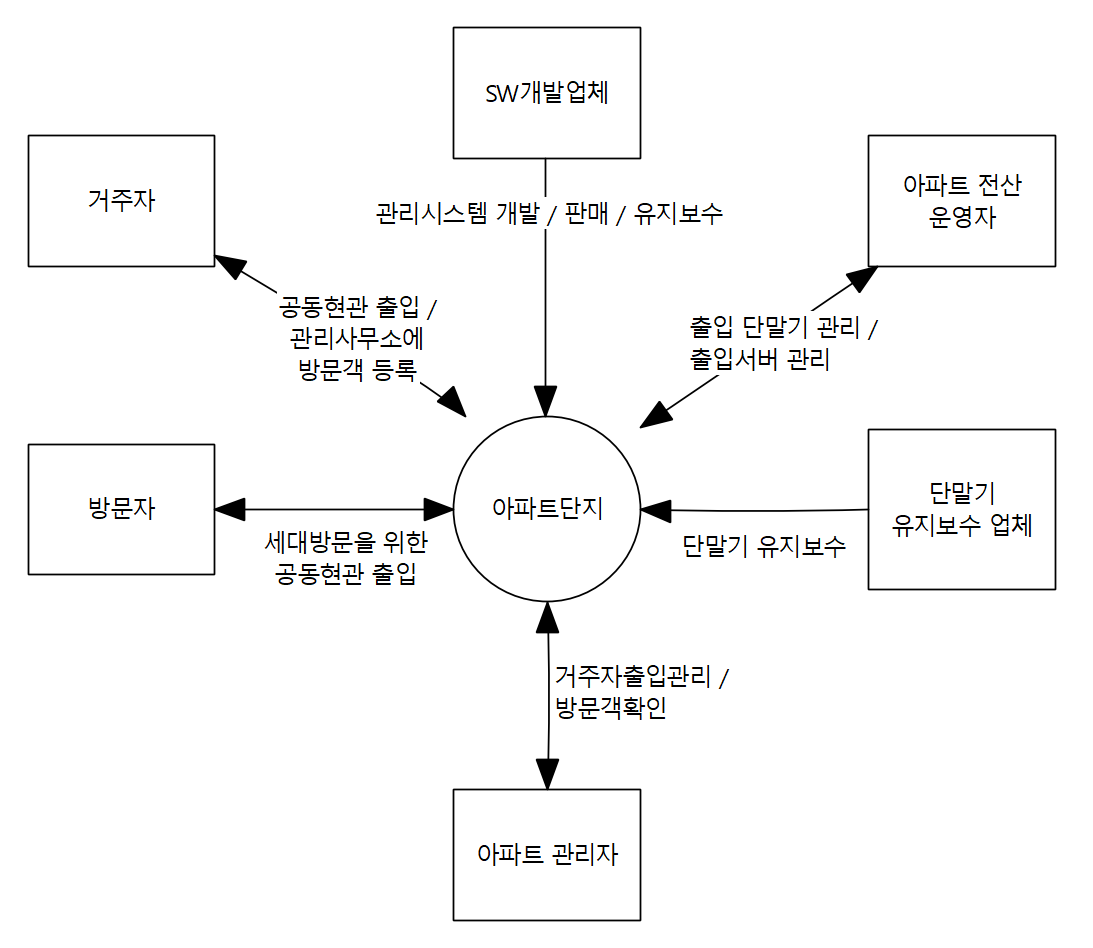
기존의 출입관리 시스템은 반드시 최초 아파트 신축 당시 함께 도입해야 하지만, 본 시스템은 신축 또는 구축 등 어떠한 아파트 환견에서도 도입이 가능하다.



새롭게 구축되는 공동 출입문 출입 관리 시스템에서는 아래 핵심기능을 목표로 개발된다.

* 상시 출입하는 거주자의 경우 얼굴인식 기능을 통해 출입이 가능하다.
* 방문자의 경우 방문세대에서 발급하는 임시 QR코드를 통해 출입이 가능하다.
* 기존에 사용중이던 방식(ex>비밀번호)도 혼용이 가능하다.
* 아파트 관리자를 통해 편리하게 출입정보의 등록, 수정, 삭제가 가능하다.
* 침입시도가 검출되는 경우 아파트 관리자를 통해 관할 경찰서로의 신고가 가능하다.

## Business Context Diagram



## Stakeholder List

|  |  |
| --- | --- |
| Stakeholder | Description |
| 거주자 | 아파트에 거주하고 있는 주민  비접촉식 인증방식을 통한 공동현관 출입을 원하고 있음  강력한 보안 시스템을 통해 출입이 관리되기를 원함  그러면서도 쉽고 빠른 인증절차를 통해 편리한 공동현관 출입이 되어야 함 |
| 아파트 관리자 | 거주자 관리, 얼굴 및 QR 등록담당  각 세대별로 얼굴인식에 사용될 데이터를 수집하고, 필요에 따라 세대에 할당될 QR코드를 발급하는 역할  해당 인증방법에 대해 직접 Database에 입력할 수 있어야 하며 추가,수정,삭제 권한을 가지고 있음 |
| 아파트 전산 운영자 | 단말기관리, 아파트 관리자 관리 (전체시스템의 운영자)  아파트 관리자가 사용하는 웹페이지를 관리하며, 인증서버에 대한 관리를 맡고 있음 |
| 방문자 | 택배, 배달등을 위해 세대현관까지 도달해야 하는 방문자  부재중일때를 대비하여 세대에서 방문 등록 후 발급받은 QR코드를 문자로 공유받아 공동현관을 출입할 수 있도록 함 |
| 단말기 유지보수업체 | 설치된 단말기의 유지보수업체  단말기를 통해 획득된 영상이나 QR코드를 시스템으로 전달받기 위해 Interface를 정의해야 함  또한 시스템 적용시 기존동작에 문제가 없어야 함 |
| SW개발업체 | 새로운 시스템을 개발하여 용인시 전체의 아파트에 제공하고자 함. |

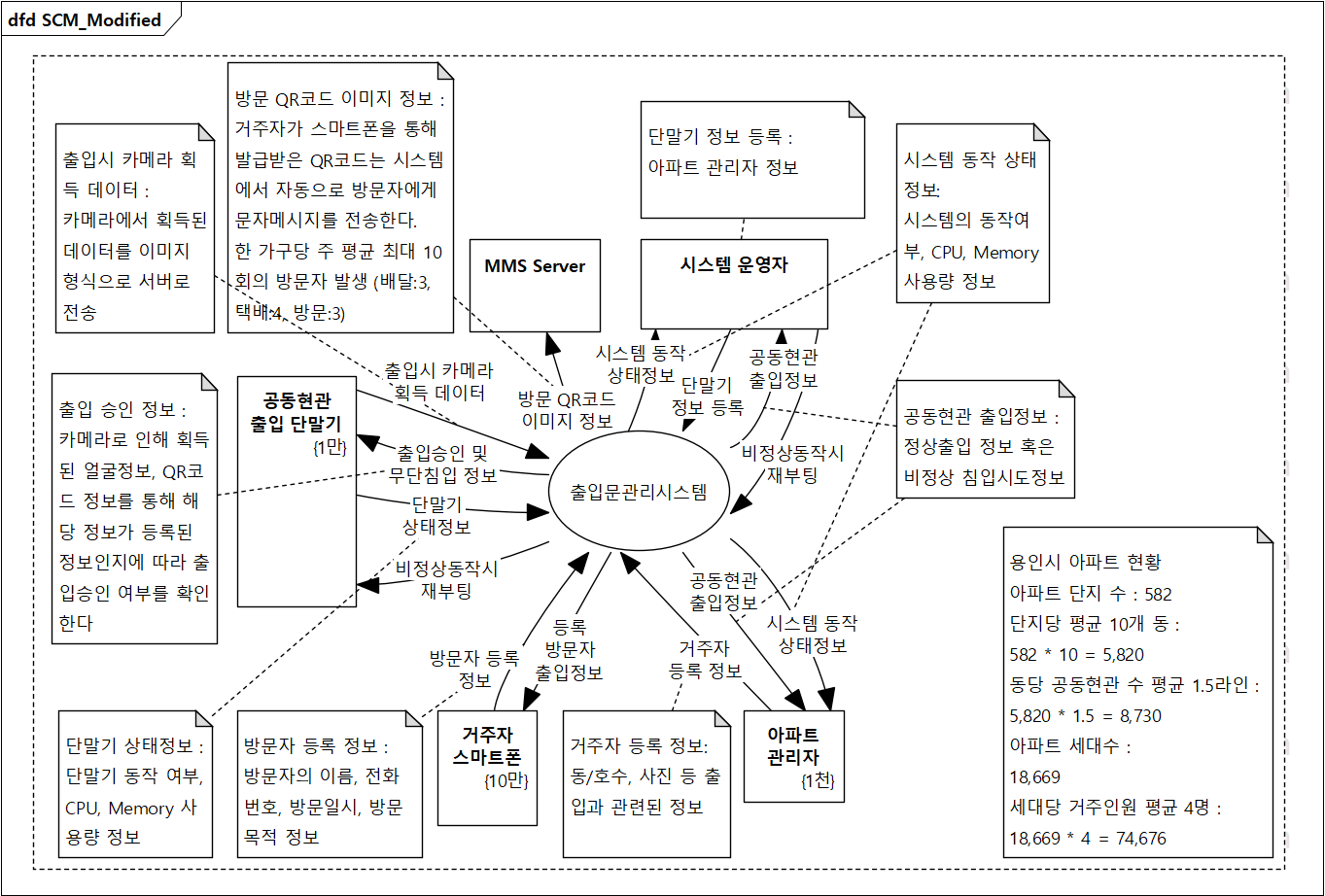
## Business Goal List

|  |  |  |  |
| --- | --- | --- | --- |
| Stakeholder | Business Goal | | |
| ID | Statement | I |
| 거주자 | BG-01 | 거주자는 빠르고 간편하게 출입하기 위하여 비 접촉식 출입을 원하고 있음. 비 접촉식 출입이 가능해 진다면 비밀번호를 누르는 기존(2초정도 소요)대비 1초가 더 빨라질 것으로 기대 | 상 |
| 아파트 관리자 | BG-02 | 아파트 관리자는 거주자와 방문자로부터 출입과 관련된 시스템 관리가 편리하게 진행되기를 바라며, 문제 발생시 문의건수가 줄어들기를 기대함 | 중 |
| 아파트 전산 운영자 | BG-03 | 아파트 관리자가 사용하는 웹페이지를 관리하며, 인증서버에 대한 관리를 맡고 있음  각 단말기의 동작상황과 서버의 동작상황을 관리자 페이지를 통해 한눈에 볼 수 있기를 기대하며, 아파트 별 단말기 등록, 수정, 삭제가 용이한 UI를 원하고 있음  인증서버에 문제가 발생시 즉각 조치가 되기를 기대하며, 침입 발생시 빠른 알림이 오기를 기대 | 중 |
| 방문자 | BG-04 | 미리 발급받은 QR코드를 통해 방문세대가 부재인 경우에도 시간을 빼앗기지 않고 출입이 가능하기를 기대 | 상 |
| 단말기 관리업체 | BG-06 | 각 단말기를 통해 획득된 얼굴인식 데이터 및 QR코드 데이터를 출입관리 시스템으로 전송하기 위한 프로토콜을 제공해 주는 역할을 해야하며, 해당 기능 적용시 기존 기능에 영향을 미치지 않기를 기대 | 중 |
| SW개발업체 | BG-07 | 새로운 시스템이 용인시 아파트 전체에 제공되기를 기대  용인시 아파트 현황  아파트 단지 수 : 582  단지당 평균 10개 동 ; 582 \* 10 = 5,820  동당 공동현관 수 평균 1.5라인 : 5,820 \* 1.5 = 8,730  아파트 세대수 : 18,669  세대당 거주인원 평균 4명 : 18,669 \* 4 = 74,676 | 중 |

\* I : Importance

# System Overview

## System Context Diagram



## External Entity List

|  |  |
| --- | --- |
| Name | Description |
| 거주자 스마트폰 | 유형 : 디바이스  핵심 기대 사항 : 스마트 폰을 통해 방문자를 등록하여, 방문자에게 QR코드가 전달되기를 기대한다. 등록한 방문자 출입시 해당 알림을 받을 수 있다.   * 방문자 등록 정보 : 방문자의 이름, 전화번호, 방문일시, 방문목적 정보)   숙련도 : 스마트폰 Application을 통해 로그인, 메뉴선택, 사용자 등록 등 일반적인 사용이 가능 |
| 아파트 관리자 | 유형 : PC  핵심 기대 사항 : 거주자가 얼굴인식을 통해 출입이 가능하도록 얼굴정보를 등록할 수 있어야 하며, 모든 출입정보를 확인할 수 있어야 한다. 출입문 관리 시스템의 동작정보를 확인할 수 있어야 한다.   * 시스템 동작 상태정보: 시스템의 동작여부, CPU, Memory 사용량 정보) * 공동현관 출입정보 : 정상출입 정보 혹은 비정상 침입시도정보   숙련도 : PC의 웹페이지에 접속할 수 있고, 원하는 동작을 메뉴를 통해 진행이 가능하다. |
| 시스템 운영 도구 | 유형 : PC  핵심 기대 사항 : 출입관리 시스템의 동작상태를 확인할 수 있으며,  숙련도 : 시스템의 전반적인 관리 지식이 있으며, 발생한 Error에 대하여 초동조치가 가능한 수준 |
| 문자전송시스템 | 유형 : 사용자  핵심 기대 사항 : 공동현관 출입에 문제가 없어야 하며, 스케줄상 정확한 시간에 출입하기 보다는, 해당 날짜에 1회 출입이 필요함  숙련도 : 스마트폰 문자를 통해 전달받은 QR코드를 통한 출입문 인증이 충분히 가능한 수준 |
| 공동현관 출입 단말기 | 유형 : Device  역할 : 카메라를 통해 획득된 영상을 시스템으로 전송하는 장치  특성 : 기존에 설치된 카메라가 존재하는 모든 공동현관 단말기   * 640x480 이상의 해상도를 가짐 * Network을 통해 시스템에 접근이 가능 * 시스템 인증시 현관 출입문 개방 가능 |

## External Interface List

|  |  |
| --- | --- |
| Name | Description |
| 출입시 카메라 획득 데이터 | 유형 :   * User Interface : 출입단말기 카메라 * System Interface : HTTPS   역할 : 출입을 위해 출입단말기 카메라로 입력되는 이미지 정보 전달  특성 :   * 크기 : 640 x 480 x 24bit \* 30 Frame/sec |
| 출입승인 및 무단침입 정보 | 유형 :   * System Interface : HTTPS, JSON   역할 : 출입관리시스템에서 인증절차를 통해 출입이 가능한지 여부를 판단하여 출입단말기로 전달  특성 :   * 1Kb 이내 JSON포멧 |
| 출입단말기 상태정보 | 유형 :   * User Interface : Web Application * System Interface : HTTPS, JSON   역할 : 출입단말기를 모니터링 하거나, 정상동작 여부를 판단하기 위해 출입관리시스템에서 출입단말기로 정보 요청  특성 :   * 10Kb 이내 JSON포멧 |
| 출입단말기 비정상동작시 재부팅 | 유형 :   * User Interface : Web Application * System Interface : HTTPS, JSON   역할 : 출입단말기의 상태정보에 따라 User는 재부팅 명령을 전달  특성 :   * 1Kb 이내 JSON포멧 |
| 방문자 등록 정보 | 유형 :   * User Interface : 스마트폰 Application * System Interface : HTTPS, JSON   역할 : 방문자를 등록하기 위하여 방문자의 정보를 시스템으로 전달  특성 :   * 1Kb 이내 JSON 포멧 * 방문자 등록정보 : 이름, 전화번호, 방문일시, 방문목적 |
| 등록 방문자 출입정보 | 유형 :   * User Interface : 스마트폰 Push Message * System Interface : Push Message Service   역할 : 방문자 출입시 해당 방문자를 등록한 거주자에게 출입알림을 전달  특성 :   * Push Notification |
| 거주자 등록정보 | 유형 :   * User Interface : Web Application * System Interface : HTTPS, JSON   역할 : 아파트 관리자는 거주자의 출입이 가능하도록 정보를 등록  특성 :   * 아파트 관리 Web에서 정보 입력 * 거주자 등록정보 : 동/호수, 사진 |
| 공동현관 출입정보 | 유형 :   * User Interface : Web Application * System Interface : HTTPS, JSON   역할 : 공동현관 출입정보 조회시 출입정보를 전달  특성 :   * 크기 : 한번에 1Mb까지 전달 * 출입정보 : 출입시간, 출입자 등록명 |
| 출입관리시스템 동작 상태정보 | 유형 :   * User Interface : Web Application * System Interface : System Timer, HTTPS, JSON   역할 : 시스템의 동작상태 조회할 수 있도록 정보 전달  특성 :   * 10Kb 이내 JSON 포멧 |
| 방문 QR코드 이미지정보 | 유형 :   * User Interface : 스마트폰 Application * System Interface : BMP Format, HTTPS   역할 : 거주자가 스마트폰을 이용해 방문자를 등록하면, 발급받은 QR코드를 이미지 형식으로  특성 :   * 이미지 사이즈 : 640 x 480 8bit 1 Frame |
| 단말기 정보 등록 | 유형 :   * User Interface : Web Application * System Interface : HTTPS, JSON   역할 : 시스템 운영자는 추가되는 단말기를 등록  특성 :   * 단말기 정보 : 아파트, 단말기 모델명, 단말기 Serial |

## System Feature List

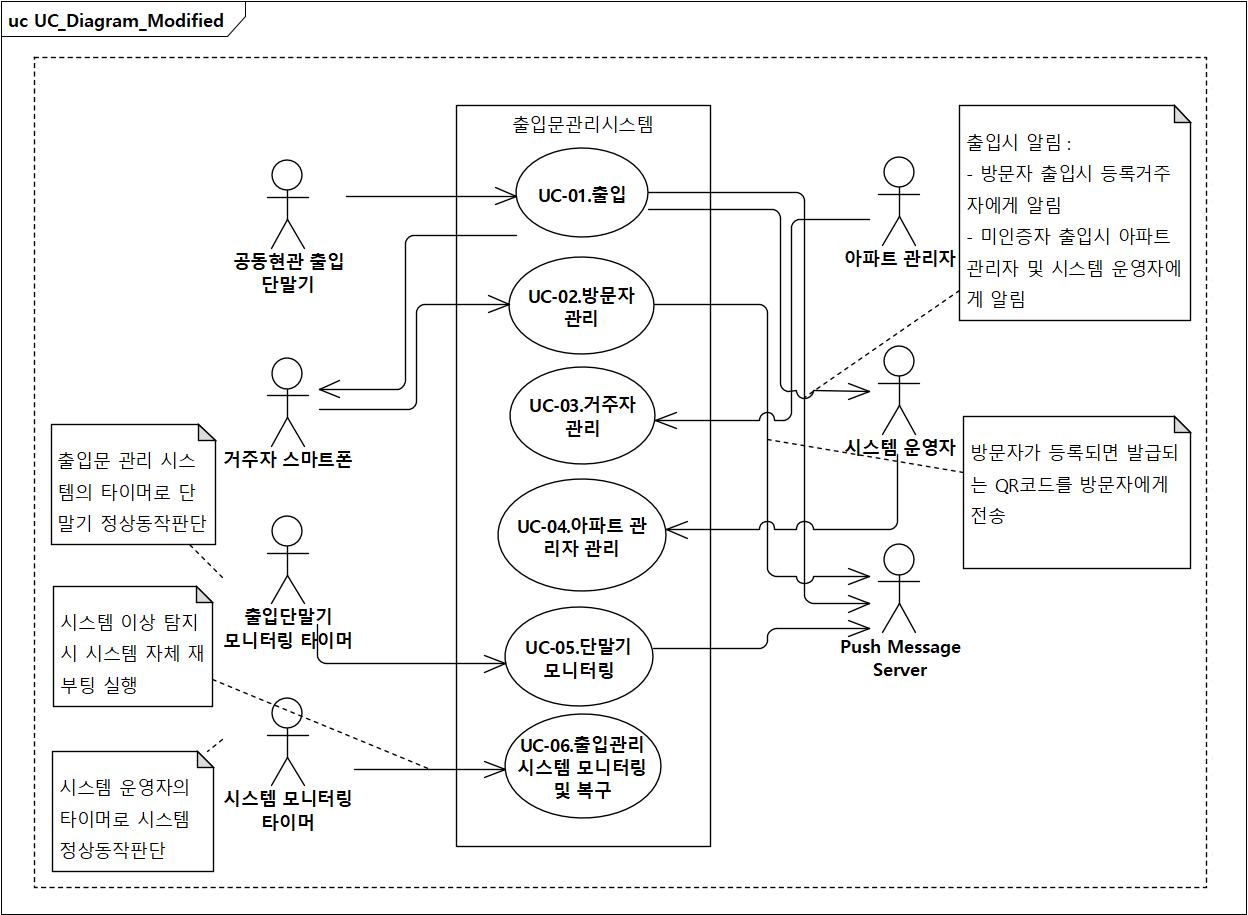
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Title | Description | I | Related  Business Goal ID |
| SF-01 | 얼굴인식 | 시스템은 출입단말기로부터 전송받은 카메라 영상 데이터를 기반으로 얼굴을 인식하는 기능을 제공한다. | 상 | BG-01 |
| SF-02 | 방문자 QR코드 발급 및 전송 | 시스템은 QR코드를 발급하여 사진형태로 방문자에게 문자메시지로 전송하는 기능을 제공한다. | 중 | BG-04 |
| SF-03 | 거주자 등록 | 시스템은 거주자의 정보를 전달받아 데이터베이스에 등록하는 기능을 제공한다. | 중 | BG-02 |
| SF-04 | 단말기 모니터링 | 시스템은 단말기의 동작상태를 모니터링하고, 장애를 인지할 수 있는 기능을 제공한다. | 상 | BG-02 |
| SF-05 | 출입관리 시스템 모니터링 및 복구 | 시스템은 장애상황을 인지를 통해 빠르게 복수할 수 있는 기능을 제공한다. | 상 | BG-03 |

\* I: Importance

# Architectural Driver

## Use Case Model

### Use Case Diagram



### Actor List

|  |  |
| --- | --- |
| Name | Description |
| 공동현관 출입 단말기 | 공동현관에 설치된 출입 단말기. 거주자 및 방문자의 출입시 획득된 영상을 시스템으로 전송하여 출입인증 절차를 시작한다.  시스템 관점에서의 Actor 역할:   * 출입자가 공동현관 출입단말기를 통해 출입을 시도 * 출입단말기 카메라에 입력된 정보를 출입시스템으로 전송 |
| 거주자 스마트폰 | 아파트에 거주하는 거주자의 스마트폰. 방문자를 등록할 수 있고, 등록한 방문자가 출입단말기를 통해 QR코드를 입력하면 알림을 받을 수 있다.  시스템 관점에서의 Actor 역할:   * 방문자를 등록하여 QR코드를 발급받을 수 있으며, 해당 QR코드는 방문자에게 MMS 형식으로 전송된다. * 등록한 방문자의 QR코드를 출입단말기에 인식하는 경우 해당 출입에 대하여 알림을 받을 수 있다. * 방문자 정보 : 이름, 전화번호, 방문일시, 방문목적 |
| 출입단말기 모니터링 타이머 | 시스템 내부에 존재하는 출입단말기 모니터링 타이머로, 단말기의 정상동작 여부를 주기적으로 확인한다.  시스템 관점에서의 Actor 역할:   * 시스템 동작 중 상시 동작하는 타이머 * 주기적으로 출입단말기의 정상동작 확인 * 비정상동작 감지 시 복구명령 가능 |
| 시스템 모니터링 타이머 | 시스템 운영자 내부에 존재하는 타이머. 출입단말기시스템의 정상동작 여부를 주기적으로 확인한다.  시스템 관점에서의 Actor 역할:   * 출입시 전달받은 인증 데이터를 통해 출입허가 * 출입이 허가되는 사용자에 대한 입력, 수정, 삭제기능 제공 |
| 아파트 관리자 | 아파트 관리자의 관리도구. 해당 관리 도구를 통해 거주자 정보를 등록할 수 있다.  시스템 관점에서의 Actor 역할:   * 거주자에 대한 정보 등록, 조회, 수정, 삭제 권한 * 거주자에 대한 정보 : 동/호수, 사진 |
| 시스템 운영자 | 출입관리시스템을 운영하는 운영자. 아파트 단위로 단말기 정보를 등록할 수 있다.  시스템 관점에서의 Actor 역할:   * 단말기의 등록, 조회, 수정, 삭제 권한 * 단말기 정보 : 아파트, 단말기 모델명, Serial 정보 |
| Push Message Server | 거주자 스마트폰에 의해 방문자가 등록되고 QR코드가 발급되는 경우, 해당 QR코드를 방문자에게 전송하는 역할  시스템 관점에서의 Actor 역할:   * 거주자 스마트폰으로 방문자의 등록이 완료되면 발급되는 QR코드를 방문자에게 전송 |

### Use Case List

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Title | Description | Priority | | System  Feature ID |
| I | D |
| UC-01 | 출입 | 공동현관 출입단말기로부터 영상을 전송받아 출입인증을 수행한다. 영상을 분석하여 사람얼굴 데이터인 경우 얼굴인식을 통해 등록여부를 판단한다. QR코드인 경우 해당 데이터를 decryption하여 등록된 방문자인지를 판단한다.  등록여부 확인 시 미등록으로 판단되면 해당 인증은 실패하며, 침입으로 판단하여 침입내역을 Push Message Server를 통해 아파트 관리자와 시스템 운영자에게 통보한다. | 상 | 상 | SF-01 |
| UC-02 | 방문자 관리 | 거주자의 스마트폰으로부터 방문자 등록을 요청받는다. 방문자의 이름, 전화번호, 방문일시, 방문목적을 입력받아 해당 내용으로 QR코드를 발급하고, Push Message Server를 통해 방문자에게 QR코드를 전송한다. | 상 | 상 | SF-05 |
| UC-03 | 거주자 관리 | 아파트 관리자는 거주자를 등록할 수 있다. 거주자로부터 동/호수, 사진을 전달받아 관리콘솔을 통해 해당 내용을 입력하면, 얼굴인식을 통한 인증절차에서 입력된 얼굴을 통해 인증이 진행된다. | 상 | 하 | SF-02 |
| UC-04 | 단말기 모니터링 | 출입문 관리 시스템은 모니터링 타이머를 통해 단말기의 상태를 모니터링 한다. 단말기 상태를 주기적으로 모니터링 한 뒤, 문제가 발생되는 경우 Push Message Server를 통해 해당 내용을 아파트 관리자에게 통보한다. | 중 | 하 | SF-02 |
| UC-05 | 출입관리시스템 모니터링 및 복구 | 시스템 운영자는 시스템 모니터링 타이머를 통해 시스템의 상태를 모니터링 한다. 시스템 상태를 주기적으로 모니터링 한 뒤, 문제가 발생되는 경우 시스템에 복구 명령을 전달하여 시스템의 문제가 해결되도록 한다. | 중 | 중 | SF-01 |

\* I: Importance, D: Difficulty

### UC-01 출입

#### Scenario List

|  |  |  |
| --- | --- | --- |
| Scenario Title | Kind | Description |
| 거주자 출입 | 기본 | 등록된 거주자가 공동현관 단말기에 얼굴을 인식한다. |
| 방문자 출입 | 선택 | QR코드를 발급받은 방문자가 공동현관 단말기에 QR코드를 인식한다. |
| 미등록자 출입 | 선택 | 얼굴이 등록되지 않은 사람이 출입을 시도한다. |
| 카메라 오작동 | 예외 | 얼굴인식에 사용되는 카메라의 오작동으로 영상이 전달되지 않아 인증서버와의 통신이 시도되지 않는다. |

#### 거주자 출입(기본)

#### Pre condition

|  |  |
| --- | --- |
| Title | Description |
| 출입단말기 대기상태 | 출입단말기가 정상적으로 동작하며, 출입대기 상태이다. |

#### Post condition

|  |  |
| --- | --- |
| Title | Description |
| 출입문 개방 | 출입문이 열린다. |

#### Flow of events

|  |  |
| --- | --- |
| Step No. | Description |
| 1 | 거주자가 출입을 위해 공동현관 출입 단말기 앞에 도착한다. |
| 2 | 출입 단말기는 카메라의 영상정보를 출입관리시스템으로 전송한다. |
| 3 | 출입관리시스템은 전달받은 영상정보에서 얼굴정보를 확인한다. |
| 4 | 출입관리시스템은 얼굴정보의 등록여부를 판단한다. |
| 5 | 출입관리시스템은 등록된 사용자임을 확인하고 출입단말기로 출입허가 명령을 보낸다. |

#### 방문자 출입(선택)

#### Pre condition

|  |  |
| --- | --- |
| Title | Description |
| 출입단말기 대기상태 | 출입단말기가 정상적으로 동작하며, 출입대기 상태이다. |

#### Post condition

|  |  |
| --- | --- |
| Title | Description |
| 출입문 개방 | 출입문이 열린다. |

#### Flow of events

|  |  |
| --- | --- |
| Step No. | Description |
| 1 | 방문자가 출입을 위해 공동현관 출입 단말기 카메라에 QR코드를 인식한다. |
| 2 | 출입 단말기는 카메라의 영상정보를 출입관리시스템으로 전송한다. |
| 3 | 출입관리시스템은 전달받은 영상정보에서 QR코드를 확인한다. |
| 4 | 출입관리시스템은 QR코드정보의 등록여부를 판단한다. |
| 5 | 출입관리시스템은 등록된 사용자임을 확인하고 출입단말기로 출입허가 명령을 보낸다. |

#### 미등록자 출입(선택)

#### Pre condition

|  |  |
| --- | --- |
| Title | Description |
| 출입단말기 대기상태 | 출입단말기가 정상적으로 동작하며, 출입대기 상태이다. |

#### Post condition

|  |  |
| --- | --- |
| Title | Description |
| 침입탐지 알림 | 아파트 관리자에게 침입시도가 있음을 알린다. |

#### Flow of events

|  |  |
| --- | --- |
| Step No. | Description |
| 1 | 미등록자가 출입을 위해 공동현관 출입 단말기 앞에 도착한다. |
| 2 | 출입 단말기는 카메라의 영상정보를 출입관리시스템으로 전송한다. |
| 3 | 출입관리시스템은 전달받은 영상정보에서 얼굴정보를 확인한다. |
| 4 | 출입관리시스템은 얼굴정보의 등록여부를 판단한다. |
| 5 | 출입관리시스템은 미등록된 사용자임을 확인하고 출입단말기로 출입불가 명령을 보낸다. |
| 6 | 출입관리시스템은 아파트 관리자에게 침입이 탐지되었음을 Push Message로 전송한다. |

### UC-02 방문자 관리

#### Scenario List

|  |  |  |
| --- | --- | --- |
| Scenario Title | Kind | Description |
| 방문자 QR코드 전송완료 | 기본 | 거주자가 거주자 스마트폰 어플리케이션을 통해 방문자를 등록하면, 방문용 QR코드가 방문자에게 문자메시지로 전달된다. |
| 방문자 QR코드 전송 실패 | 예외 | 거주자가 거주자 스마트폰 어플리케이션을 통해 방문자를 등록하였으나, 방문용 QR코드가 방문자에게 문자메시지로 전달에 실패한다. |

#### 방문자 QR코드 전송(기본)

#### Pre condition

|  |  |
| --- | --- |
| Title | Description |
| 거주자 스마트폰 어플리케이션 설치 | 거주자 스마트폰에 방문자 QR신청을 위한 어플리케이션이 설치되어 있다. |
| 방문자 스마트폰 문자메시지 수신가능 | 방문자는 스마트폰으로 문자메시지를 수신할 수 있는 환경 |

#### Post condition

|  |  |
| --- | --- |
| Title | Description |
| 방문자 QR코드 문자메시지 확인 | 방문자는 스마트폰에 문자메시지를 통해 QR코드를 전송받은다. |

#### Flow of events

|  |  |
| --- | --- |
| Step No. | Description |
| 1 | 거주자는 스마트폰 어플리케이션을 통해 방문자 정보(방문시간/목적/전화번호)를 입력한다. |
| 2 | 출입관리시스템은 전달받은 방문자정보를 통해 QR코드 이미지를 생성한다. |
| 3 | 출입관리시스템은 QR코드 이미지를 방문자에게 문자메시지로 전달한다. |

### UC-03 거주자 관리

#### Scenario List

|  |  |  |
| --- | --- | --- |
| Scenario Title | Kind | Description |
| 거주자 등록 | 기본 | 아파트 관리자는 공동현관 출입을 위해 출입관리 시스템에 거주자 정보(동/호수, 얼굴사진)를 입력한다. |
| 거주자 등록실패 | 예외 | 출입관리시스템의 비정상 동작으로 인하여, 거주자 등록페이지에 접속할 수 없는 상태. |

#### 거주자 등록(기본)

#### Pre condition

|  |  |
| --- | --- |
| Title | Description |
| 아파트 관리자용 콘솔 로그인 | 아파트 관리자는 웹브라우저를 통해 아파트 관리페이지에 정상적으로 접근하여 로그인이 완료되어 있어야 한다. |
| 거주자 등록페이지 정상동작 상태 | 아파트 관리를 위한 웹페이지가 정상적으로 동작하는 상태 |

#### Post condition

|  |  |
| --- | --- |
| Title | Description |
| 거주자 등록완료 | 거주자의 얼굴이 출입관리시스템에 등록되면 등록된 거주자는 얼굴인식을 통해 출입이 가능하다. |

#### Flow of events

|  |  |
| --- | --- |
| Step No. | Description |
| 1 | 아파트 관리자는 아파트 관리자 콘솔에서 거주자 등록 페이지에 접근한다. |
| 2 | 아파트 관리자는 거주자 등록페이지에서 거주자 정보(동/호수, 얼굴사진)를 입력한다. |
| 3 | 출입관리시스템은 전달받은 거주자 정보를 출입가능 사용자로 등록한다. |

### UC-04 단말기 모니터링

#### Scenario List

|  |  |  |
| --- | --- | --- |
| Scenario Title | Kind | Description |
| 단말기 모니터링 | 기본 | 출입관리시스템은 단말기의 정상동작 상태를 확인하고, 이상 |
| 단말기 장애 감지 | 기본 | 출입관리시스템이 단말기의 장애를 감지하면 아파트 관리자에게 통보한다. |

#### 단말기 모니터링(기본)

#### Pre condition

|  |  |
| --- | --- |
| Title | Description |
| 출입관리시스템 모니터링 컴포넌트 정상동작 | 출입관리시스템의 모니터링 관련 컴포넌트가 정상동작 상황이어야 한다. |

#### Post condition

|  |  |
| --- | --- |
| Title | Description |
| 정상동작 상황 확인 | 출입관리시스템에서 모니터링한 결과로 출입단말기가 정상동작임을 알 수 있다. |

#### Flow of events

|  |  |
| --- | --- |
| Step No. | Description |
| 1 | 출입단말시스템의 모니터링 컴포넌트가 출입단말기의 정상동작 여부를 확인한다. |

#### 단말기 장애 감지(기본)

#### Pre condition

|  |  |
| --- | --- |
| Title | Description |
| 출입관리시스템 모니터링 컴포넌트 정상동작 | 출입관리시스템의 모니터링 관련 컴포넌트가 정상동작 상황이어야 한다. |

#### Post condition

|  |  |
| --- | --- |
| Title | Description |
| 장애탐지 후 아파트 관리자에게 통보 | 출입관리시스템에서 모니터링한 결과 출입단말기의 장애가 판단되면 아파트 관리자에게 알림을 보낸다. |

#### Flow of events

|  |  |
| --- | --- |
| Step No. | Description |
| 1 | 출입단말시스템의 모니터링 컴포넌트가 출입단말기의 정상동작 여부를 확인한다. |
| 2 | 출입단말기의 장애가 탐지되면 아파트 관리자에게 통보한다. |

### UC-05 출입관리시스템 모니터링 및 복구

#### Scenario List

|  |  |  |
| --- | --- | --- |
| Scenario Title | Kind | Description |
| 출입관리시스템 모니터링 | 기본 | 출입관리시스템의 동작상태를 모니터링한다. |
| 출입관리시스템 장애감지 | 기본 | 출입관리시스템의 장애를 감지하면 요청을 수행하는 서버를 대체서버로 바꾸고 기존서버에 재시작 명령을 전송한다. |

#### 출입관리시스템 모니터링(기본)

#### Pre condition

|  |  |
| --- | --- |
| Title | Description |
| 시스템 모니터링 타이머 컴포넌트 정상동작 | 시스템 모니터링 타이머 컴포넌트가 정상동작 상황이어야 한다. |

#### Post condition

|  |  |
| --- | --- |
| Title | Description |
| 정상동작 상황 확인 | 시스템 모니터링 타이머는 echo명령을 수신하여 정상동작을 확인한다. |

#### Flow of events

|  |  |
| --- | --- |
| Step No. | Description |
| 1 | 시스템 모니터링 타이머는 출입관리시스템의 정상동작 여부를 확인하기 위한 Ping Message를 송신한다. |
| 2 | 출입관리시스템은 ping 명령을 수신하여 정상동작인 경우 echo를 회신한다. |

#### 출입관리시스템 장애 감지(기본)

#### Pre condition

|  |  |
| --- | --- |
| Title | Description |
| 시스템 모니터링 타이머 컴포넌트 정상동작 | 시스템 모니터링 타이머 컴포넌트가 정상동작 상황이어야 한다. |

#### Post condition

|  |  |
| --- | --- |
| Title | Description |
| 정상동작 상황 확인 | 시스템 모니터링 타이머는 echo명령을 설정된 시간동안 수신하지 못하는 경우 장애상태로 판단하고 대체서버를 사용한다. 장애가 발생한 서버는 재시작한다. |

#### Flow of events

|  |  |
| --- | --- |
| Step No. | Description |
| 1 | 시스템 모니터링 타이머는 출입관리시스템의 정상동작 여부를 확인하기 위한 ping명령를 송신한다. |
| 2 | 시스템 모니터링 타이머는 ping명령을 송신한 후 일정시간 대기한다. |
| 3 | 시스템 모니터링 타이머는 일정시간 echo명령을 받지 못한 경우 서버의 장애로 판단한다. |
| 4 | 시스템 모니터링 타이머는 장애로 판단된 서버는 재시작을 명령하며, 대체서버로 역할을 변경한다. |

## Quality Attribute Scenario

### QA Scenario List

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Title | Type | Priority | | Related  Use Case | System  Feature  ID |
| I | D |
| QA-01 | 출입 인가 처리 정확성 | Accuracy | 상 | 상 | UC-01 | SF-01 |
| QA-02 | 침입 시도 통보 시간 | Performance | 상 | 중 | UC-01 | SF-01 |
| QA-03 | 시스템 이상 발생시 대체가능 서버 활용 | Availability | 중 | 상 | UC-05 | SF-05 |
| QA-04 | 단말기 이상 모니터링 | Availability | 중 | 상 | UC-04 | SF-04 |
| QA-05 | 새로운 출입 단말기 지원 | Maintainability | 상 | 중 | UC-01 | - |

### QA-01 출입 인가 처리 정확성

|  |  |
| --- | --- |
| QA Type | Accuracy |
| Description | 출입관리시스템은 정상동작 상태에서 인식된 데이터와 등록된 데이터를 비교하여 동일한 얼굴인 경우 99.9%로 인증 허가를 발급한다. |
| Source of Stimulus | 출입단말기   * 출입단말기 최대 10만대 |
| Stimulus | 출입단말기로부터 얼굴데이터를 수신   * 등록가능한 얼굴정보 최대 1000만개 * 영상유형 : 사진 * 영상 해상도 : 출입단말기 모델에 따라 SD, HD, FHD 등의 다양한해상도로 제공될 수 있음 * 영상품질 : 다양한 촬영환경에 따라 품질이 달라질 수 있음 |
| Artifact | 공동현관 출입 관리 시스템 |
| Environment | 출입관리시스템 정상동작 상태   * CPU 사용량 70%, Memory 사용량 60% 이하 * 최대 동시 인증요청 1000개 |
| Response | 전달받은 얼굴 데이터의 출입가능 여부 |
| Response Measure | 등록된 얼굴에 대한 출입허가 정확도 99.9% 이상 |

### QA-02 침입 시도 통보 시간

|  |  |
| --- | --- |
| QA Type | Performance |
| Description | 출입관리시스템은 정상동작 상태에서 인식된 얼굴데이터가 미등록으로 판단되는 경우 침입으로 판단하여 아파트 관리자에게 통보한다. |
| Source of Stimulus | 출입단말기 |
| Stimulus | 서버에 등록되지 않은 사용자의 카메라 얼굴인식 |
| Artifact | 출입관리시스템 |
| Environment | 출입단말기   * 얼굴인식이 원활하게 가능한 카메라 상태 * 출입문이 닫혀있는 상태 * 카메라 시야에 얼굴이 잘 들어오는 상태   출입관리시스템   * CPU 70%이하, Memory 70%이하의 정상동작 상태 * 동시 인증요청 1000개 이하인 상태   Push Message   * Push Message는 FCM Server의 환경에 따라 전송 시간이 달라질 수 있음 |
| Response | 인식된 얼굴이 미등록 상태임을 아파트관리자와 운영자에게 Push Message형태로 전송 |
| Response Measure | 출입단말기에서 미등록자의 얼굴이 인식된 후 1초내에 아파트관리자와 운영자에게 Push Message 전달 완료 |

### QA-03 시스템 이상 발생시 대체가능 서버 활용

|  |  |
| --- | --- |
| QA Type | Availability |
| Description | 출입관리시스템의 이상 발생시 이중화 된 서버가 해당 명령을 수행할 수 있도록 한다. |
| Source of Stimulus | 시스템 모니터링 컴포넌트 |
| Stimulus | 출입관리시스템으로 주기적인 ping/echo 통신시도   * ping의 주기는 Performance를 고려하여 10초단위로 함 * echo의 최대 지연시간은 1000ms으로 함 * 시스템 모니터링 컴포넌트는 로드밸런스 기능과 같이 동작함 |
| Artifact | 출입관리시스템 |
| Environment | 시스템 모니터링 컴포넌트 정상동작   * 요청이 없는 상태인 경우 10초단위로 ping/echo 동작 * 요청이 있는 경우 로드밸런서는 시스템의 가용여부를 먼저 판단하고 있으므로, 장애인지시 복구작업이 바로 시작됨 |
| Response | 장애가 발생한 서버는 즉시 복구작업에 들어가며, 출입관리시스템의 가용성을 위하여 다음에 들어온 요청부터는 대체서버를 통해 동작됨 |
| Response Measure | 장애가 발생한 서버는 10초이내에 탐지되어야 하며, 장애발생 시점으로부터 20초 이내에 복구되어야 한다. 서버 장애 발생 즉시 대체서버로 운영이 가능하여야 한다. |

### QA-04 단말기 이상 모니터링

|  |  |
| --- | --- |
| QA Type | Availability |
| Description | 출입관리시스템은 출입단말기의 정상동작 상태를 모니터링하여 장애가 탐지되면 아파트 관리자에게 통보한다. |
| Source of Stimulus | 출입관리시스템 |
| Stimulus | 출입관리시스템의 모니터링   * 단말기 환경에 따라 ping/echo방식 혹은 heartbeat방식으로 모니터링 시도 * ping/echo방식에서 ping의 주기는 Performance를 고려하여 10초 단위로 설정 * echo의 최대 지연시간은 1000ms으로 함 * heartbeat방식의 단말기는 단말기 설정에 맞게 주기를 전달받음 * heartbeat 설정주기에서 3초 이상 지연이 발생하는 경우 장애발생으로 판단   Push Message   * Push Message는 FCM Server의 환경에 따라 전송 시간이 달라질 수 있음 |
| Artifact | 출입단말기 |
| Environment | 출입관리시스템   * CPU 70%이하, Memory 70%이하의 정상동작 상태 * 동시 인증요청 1000개 이하인 상태 |
| Response | 출입단말기의 장애를 탐지하여, 아파트 관리자에게 장애상황을 Push Message 형식으로 전송 |
| Response Measure | ping/echo방식을 지원하는 단말기에서는 장애발생시 11초내에 장애상황을 판단하여야 하며, 발생시점으로부터 15초내에 아파트 운영자에게 Push Message가 전달되어야 한다.  heartbeat방식을 지원하는 단말기에서는 heartbeat 주기에 따라 장애판단시간이 달라질 수 있으며, 장애탐지 이후 4초내에 아파트 운영자에게 Push Message가 전달되어야 한다. |

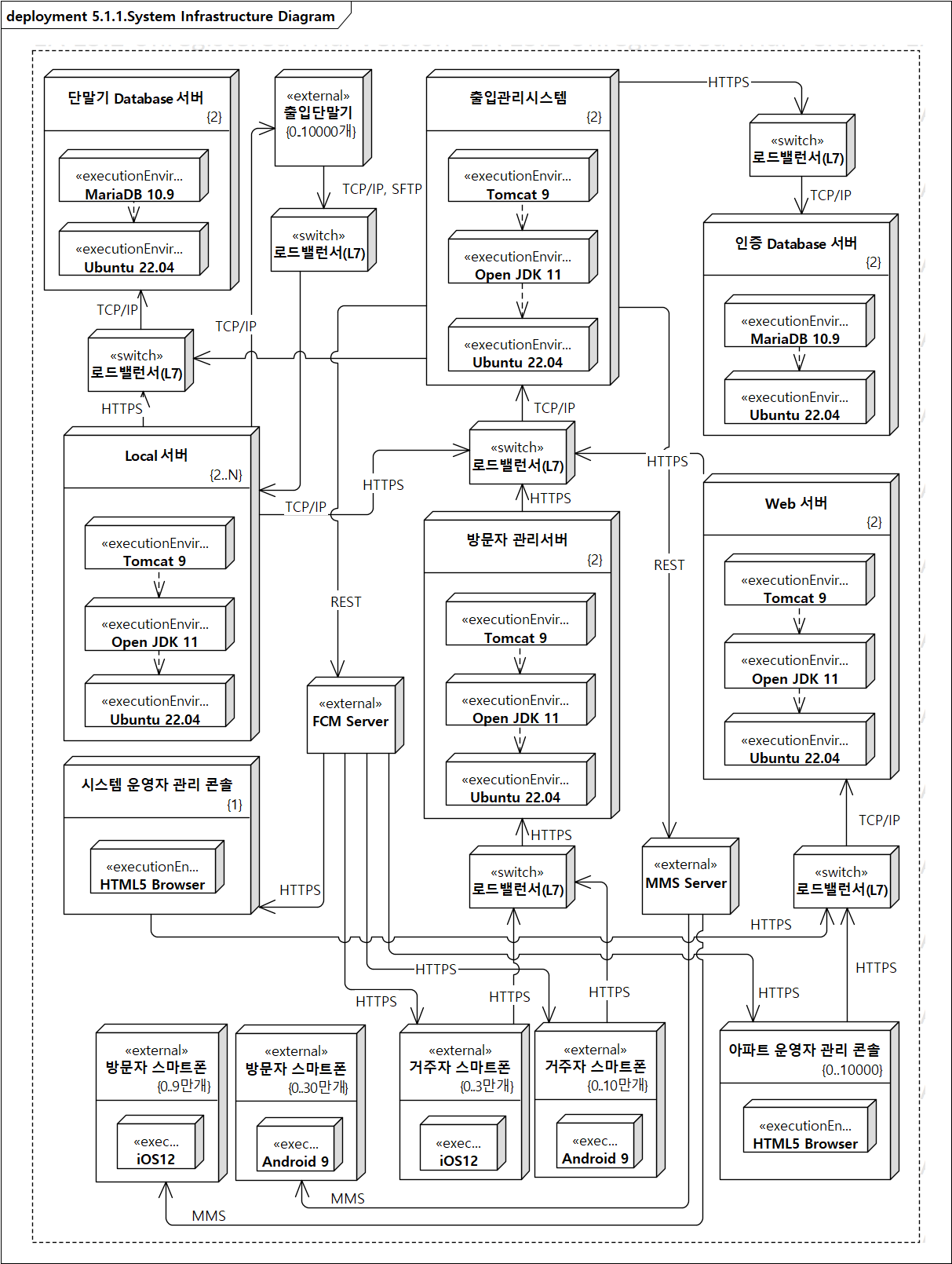
### QA-05 새로운 출입 단말기 지원

|  |  |
| --- | --- |
| QA Type | Maintainability |
| Description | 새로운 출입단말기가 기존의 인터페이스에 맞지 않는 경우 변경점을 최소화하여 빠르게 대응할 수 있어야 한다. |
| Source of Stimulus | 시스템의 도입되지 않은 아파트의 관리자 |
| Stimulus | 출입관리시스템을 도입하고자 함   * 새로운 아파트의 출입단말기는 기존 시스템으로 동작이 되지 않음 * 동작이 가능하도록 출입관리시스템 수정 및 도입이 필요함 |
| Artifact | 출입관리시스템 개발/테스터 |
| Environment | 출입관리시스템 개발환경   * 개발 리소스를 확보할 수 있는 상태 * 각 컴포넌트를 수정 및 빌드, 배포할 수 있는 상태 * 새 시스템을 통해 테스트가 가능한 상태 |
| Response | 출입관리시스템이 새로운 아파트의 출입단말기 환경에서 정상동작 |
| Response Measure | 개발, 테스트, 통합, 배포, 운영 등의 정상동작 확인까지 1MM 이내로 완료 |

# Top Level Design Description

## System Infrastructure View

### System Infrastructure Diagram



### Node Specification

|  |  |
| --- | --- |
| Name | Description |
| 출입단말기 | 역할 : 아파트 공동현관에 존재하는 외부 Device   * 인증을 통해 출입문을 개방하거나, 제한할 수 있음 * 출입관리시스템이 얼굴인증을 하기 위한 얼굴정보를 카메라를 통해 전송 |
| 출입관리시스템 | 역할 : Local 서버, Web 서버, 방문자 관리서버로부터 Database접근 요청을 처리   * HW 사양: 최대 10만 유저의 요청을 안정적으로 처리할 수 있는 성능을 보장하기 위함 * CPU: 3.4 GHz / 32 processors * Memory: 64GB * Network Bandwidth: 10Gbps * Multiplicity: 2개 (가용성을 위해 2개의 서버로 부하를 분산하며, 장애발생시를 대비함) |
| Local 서버 | 역할 : 출입단말기로부터 영상데이터를 받아 얼굴인식을 처리하며, 특징점 데이터로 변환한 뒤 출입관리시스템에 인증을 요청한다.   * HW 사양: 최대 10만 유저의 요청을 안정적으로 처리할 수 있는 성능을 보장하기 위함 * CPU: 3.4 GHz / 32 processors * Memory: 64GB * Network Bandwidth: 10Gbps   Multiplicity: 2개 (가용성을 위해 최소 2개부터 CPU와 Memory 상황에 맞추어 scalable하게 instance가 증가될 수 있도록 auto-scaling 적용됨) |
| 인증 Database 서버 | 역할 : 출입관리시스템으로부터 등록할 사용자의 데이터를 받아 저장   * HW 사양 : 2.5GHz Intel Processor * Memory: 64GB * Network Bandwidth: 10Gbps * Multiplicity: 2대 (가용성을 높이기 위해 이중화) |
| Web 서버 | 역할 : 아파트관리자가 거주자 정보를 등록할 수 있는 Web Service를 제공하는 서버   * HW 사양 : 2.5GHz Intel Processor * Memory: 64GB * Network Bandwidth: 10Gbps   Multiplicity: 2대 (가용성을 높이기 위해 이중화) |
| 단말 Database 서버 | 역할 : 출입관리시스템으로부터 단말기 정보 데이터를 받아 저장   * HW 사양 : 2.5GHz Intel Processor * Memory: 64GB * Network Bandwidth: 10Gbps   Multiplicity: 2대 (가용성을 높이기 위해 이중화) |
| 방문자 관리 서버 | 역할 : 거주자 스마트폰으로부터 방문자 정보를 등록할 수 있도록 Application과 Interaction 역할   * HW 사양 : 2.5GHz Intel Processor * Memory: 64GB * Network Bandwidth: 10Gbps   Multiplicity: 2대 (가용성을 높이기 위해 이중화) |
| 아파트 관리자 관리콘솔 | 역할 : 거주자 정보를 등록하기 위한 Web Browser   * HTML5를 지원하는 Web Browser |
| 시스템 운영자 관리 콘솔 | 역할 : 아파트 정보를 등록하기 위한 Web Browser   * HTML5를 지원하는 Web Browser |
| 로드밸런서 | 역할 : 부하 분산을 위해 사용하는 L7 Switch   * 부하 분산으로 가용성을 높이기 위해 Local Server, Web Server, App Server, 출입관리시스템 Server, Database 앞에 위치 * 서버 장애 발생시 로드밸런싱을 통해 가용한 서버로 요청을 보내며, 장애탐지된 서버는 즉시 복구를 시도 |

### Execution Environment Specification

|  |  |  |
| --- | --- | --- |
| Node | Name | Description |
| 아파트 운영자 관리콘솔  시스템 운영자 관리콘솔 | HTML5 Browser | HTTPS를 안정적으로 지원하기 위하여 HTML5를 정상지원하는 Web Browser를 사용 |
| 거주자 스마트폰  방문자 스마트폰 | Android 9 | Android 안정화 버전으로 HTTPS통신이 지원되는 최소 버전 |
| 거주자 스마트폰  방문자 스마트폰 | iOS 12 | iOS 안정화 버전으로 최신버전에서도 안정적으로 구동이 가능 |
| 출입관리시스템  Local 서버  방문자관리서버  인증 Database 서버  단말 Database 서버  Web 서버 | Ubuntu 22.04 | 배포 후 장기간 지원을 위하여 현재 최신버전 사용 |
| 출입관리시스템  Local 서버  방문자관리서버  Web 서버 | Open JDK 11 | Tomcat 9을 안정적으로 지원하기 위하여 사용 |
| 출입관리시스템  Local 서버  방문자관리서버  Web 서버 | Tomcat 9 | 웹 서버와 연동하여 실행할 수 있는 자바 환경 제공을 위해, 최신 안정화 버전 인 Tomcat 9.0으로 결정 |
| 인증 Database 서버  단말 Database 서버 | MariaDB 10.9 | 무료사용이 가능하며, 거주자 및 방문자 데이터는 유동성이 없는 것으로 판단하여 sql문법의 Database사용 |

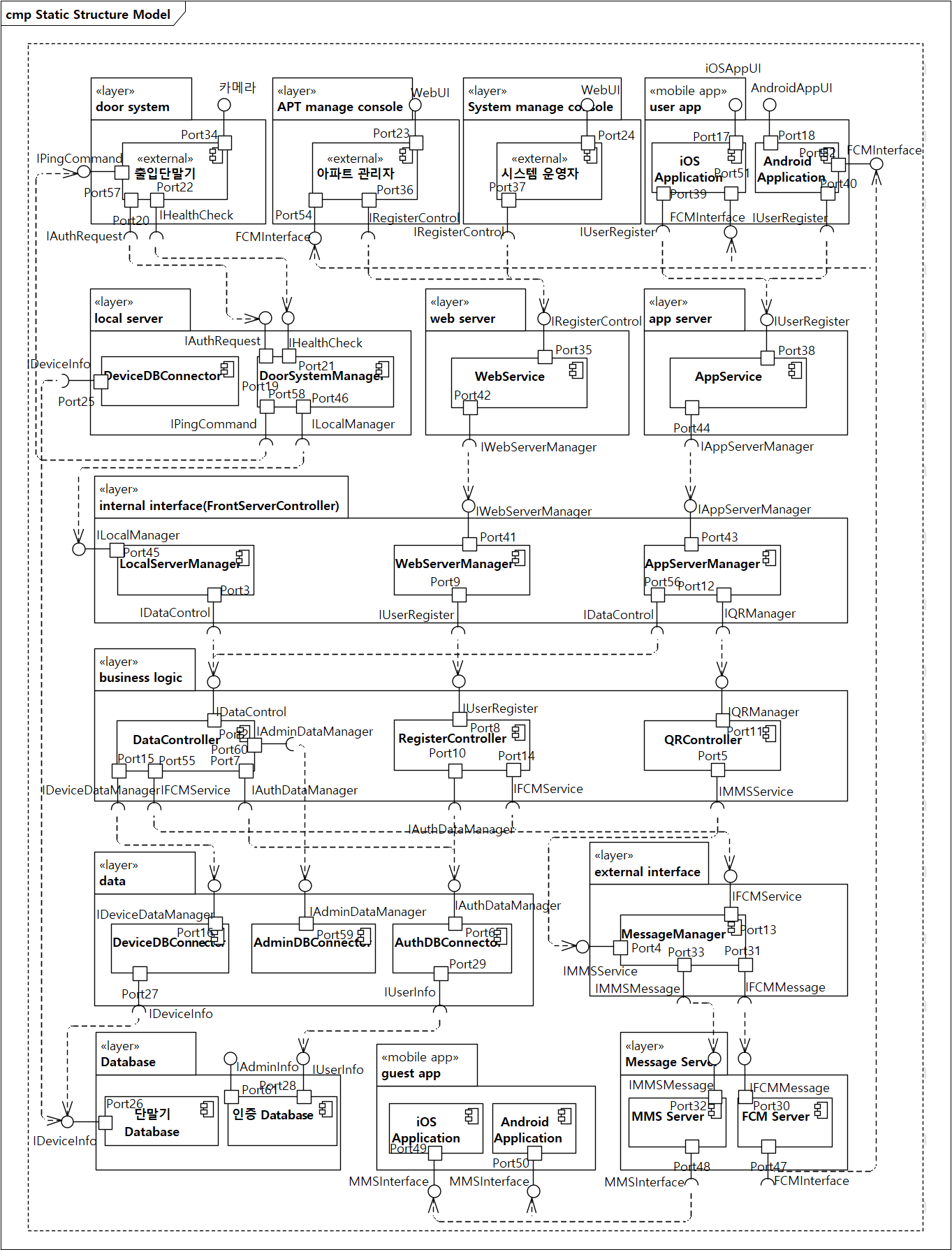
### Communication Path Specification

|  |  |
| --- | --- |
| Path | Description |
| 출입단말기 -> LocalServer | TCP/IP   * 출입단말기가 지원하는 카메라 영상 전송 프로토콜 * 출입단말기에서 획득한 카메라 영상을 Local 서버로 전송하기 위해 사용   SFTP   * 출입단말기가 지원이 가능한 경우 보안을 위해 SFTP 사용 |
| 각종서버 -> 로드밸런서  (Local 서버, 방문자 관리서버, Web 서버, 출입관리시스템) | HTTPS   * 보안품질 향상을 위하여 사용 |
| 아파트 관리자, 시스템 운영자 -> Web 서버 |
| 로드밸런서 -> 해당분산서버 | TCP/IP   * 로드밸런서에 들어오는 HTTPS보안의 요청에 대하여 부하를 줄이기 위하여 분산서버에 전송시에는 TCP/IP를 사용 |
| 각종서버 -> FCM, MMS Server | REST   * FCM Service, MMS Service에서 제공하는 REST형식의 API를 통해 통신 |

## Structure View

### Static Structure Model

#### Static Structure Diagram



#### Element List

<출입관리시스템>

|  |  |  |
| --- | --- | --- |
| Name | Responsibility | Relevant ADs |
| *internal interface (FrontServerController)* | *출입관리시스템 서버 내부에서 Local Server, Web Server, App Server의 요청을 받는 역할* | *-* |
| LocalServerManager | Local Server로부터 요청할 인증정보를 전달받아 인증을 요청하는 컴포넌트 | UC-01  QA-01  QA-02 |
| WebServerManager | Web Server로부터 등록할 아파트 거주자 정보를 입력받아 등록을 요청하는 컴포넌트 | UC-03  QA-01 |
| AppServerManager | App Server로부터 등록할 방문자 정보를 입력받아 출입관리시스템으로 QR발급 및 인증정보 등록을 요청하는 컴포넌트 | UC-02  QA-01 |
| *Business logic* | *출입관리시스템 서버 내부에서 입력된 데이터를 처리하는 역할* | *-* |
| DataController | 출입 인증을 요청받아 DBConnector로 전달하며, 인증실패를 판단하여 실패시 Message전송을 요청하는 컴포넌트 | UC-01  UC-03  QA-02 |
| RegisterController | 거주자정보 및 아파트관리자 정보를 입력받아 등록을 요청하는 컴포넌트 | UC-02  UC-03  QA-05 |
| QRController | 방문자 등록을 요청받아 QR코드를 생성하고, MMS로 전달하는 컴포넌트 | UC-02 |
| *Data* |  | *-* |
| DeviceDBConnector | 단말기 Database에 등록, 조회를 요청하는 컴포넌트 | UC-04  QA-04 |
| AuthDBConnnector | 인증 Database에 등록, 조회를 요청하는 컴포넌트 | UC-01  UC-02  UC-03  QA-01  QA-02 |
| *external interface* |  | *-* |
| MessageManager | FCM Server, MMS Server에 Message전송을 요청하는 컴포넌트 | UC-02  UC-04  QA-02  QA-04 |

<External Device>

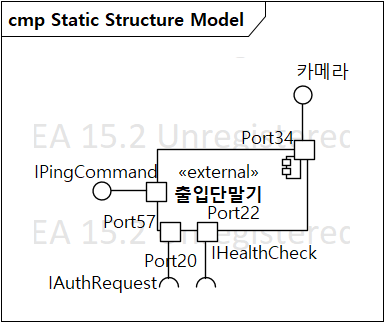
|  |  |  |
| --- | --- | --- |
| Name | Responsibility | Relevant ADs |
| *door system* | *출입단말기의 Interface layer* | *-* |
| 출입단말기 | 출입단말기 내부 컴포넌트로 영상을 전송하고, 단말기 가용성 체크를 위한 Interface 제공 | UC-01  UC-04  QA-01  QA-02  QA-04 |
| *APT manage console* | *아파트 거주자 관리를 위한 Web Page* | *-* |
| 아파트 관리자 | 거주자 등록을 위한 Web Page | QA-03 |
| *System manage console* | *아파트 관리를 위한 Web Page* | *-* |
| 시스템 운영자 | 아파트 등록을 위한 Web Page |  |
| *user app* | *거주자 스마트폰 Application* |  |
| iOS Application | iOS사용 스마트폰 Application | UC-02 |
| Android Application | Android OS사용 스마트폰 Application | UC-02 |
| *guest app* | *방문자 스마트폰* |  |
| iOS Application | MMS를 전송받기 위한 iOS사용 스마트폰 | UC-01 |
| Android Application | MMS를 전송받기 위한 Android OS사용 스마트폰 | UC-01 |

<server>

|  |  |  |
| --- | --- | --- |
| Name | Responsibility | Relevant ADs |
| *local server* | *출입단말기와 통신하는 서버* |  |
| DeviceDBConnector | 출입단말기의 가용여부를 판단하기 위하여 device정보를 읽어오는 컴포넌트 | UC-04, QA-04 |
| DoorSystemManager | 단말기로부터 영상정보를 수신하고, 단말기 가용성을 위한 Interface를 제공하는 컴포넌트 | UC-01  UC-04  QA-01  QA-04  QA-05 |
| *web server* | *아파트 관리자와 시스템 운영자에게 Web Service제공* | *-* |
| WebService | 아파트 관리자와 시스템 운영자가 거주자 등록 및 아파트 관리자 등록하기 위해 Web Service를 제공하는 컴포넌트 | UC-03 |
| *app server* | *거주자 스마트폰 Application을 위한 서버* | *-* |
| AppService | 거주자가 스마트폰 Application을 통해 방문자를 등록할 수 있도록 Service하는 서버 컴포넌트 | UC-02 |
| *Database* | *데이터 관리를 위한 Database Layer* | *-* |
| 단말기 Database | 출입단말기 정보를 가지고 있는 Database | UC-04  QA-04 |
| 인증 Database | 출입인증 정보를 가지고 있는 Database | UC-01  UC-02  UC-03  QA-01  QA-02 |
| *Message Server* | *각종 Message를 보내는 Layer* | *-* |
| MMS Server | 문자메시지를 전송하기 위한 서버 | UC-02 |
| FCM Server | Push Message를 전송하기 위한 서버 | UC-01  UC-04  QA-02  QA-04 |

### 출입단말기 - User interface

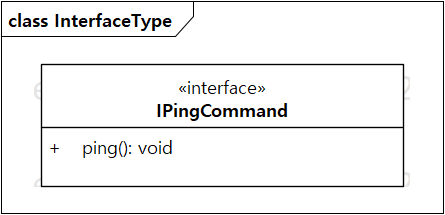
#### 출입단말기 Specification



#### Interface List

|  |  |  |
| --- | --- | --- |
| Name | Kind | Description |
| 카메라 | Provided | 카메라 정보를 받아오기 위한 인터페이스 |
| IPingCommand | Provided | 출입단말기 가용여부를 판단하기 위해 ping/echo방식을 지원하기 위한 인터페이스 |

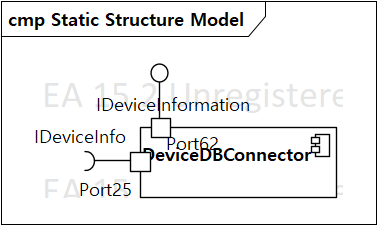
#### IPingCommand Interface Specification



|  |  |
| --- | --- |
| Operation | Responsibility |
| ping | 단말기에 ping message를 전달하기 위한 오퍼레이션 |

### local server – Business Layer

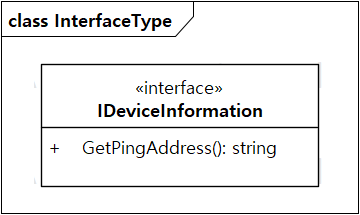
#### DeviceDBConnector Specification



#### Interface List

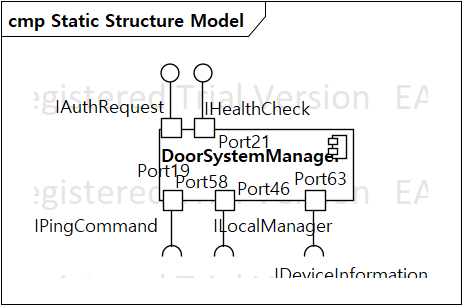
|  |  |  |
| --- | --- | --- |
| Name | Kind | Description |
| IDeviceInfomation | Provided | 출입단말기의 가용여부 판단을 위해 ping message를 전달할 주소를 제공하는 컴포넌트 |

#### IDeviceInfomation Interface Specification



|  |  |
| --- | --- |
| Operation | Responsibility |
| GetPingAddress | ping message를 전달해야할 출입단말기 address를 리턴하는 오퍼레이션 |

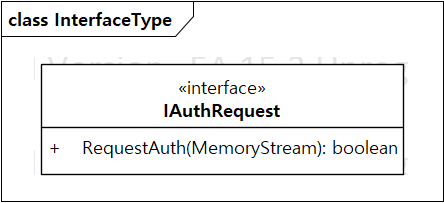
#### DoorSystemManager Specification



#### Interface List

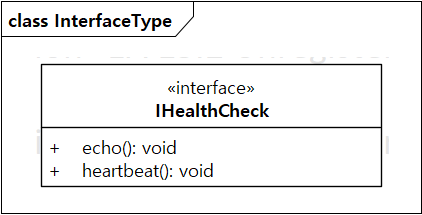
|  |  |  |
| --- | --- | --- |
| Name | Kind | Description |
| IAuthRequest | Provided | 출입단말기가 인증을 요청하기 위한 인터페이스 |
| IHealthCheck | Provided | 출입단말기로부터 heartbeat신호를 전달받기 위한 인터페이스 |

#### IAuthRequest Interface Specification



|  |  |
| --- | --- |
| Operation | Responsibility |
| RequestAuth | 출입단말기가 카메라 영상정보 인증정보를 요청하기 위한 오퍼레이션 |

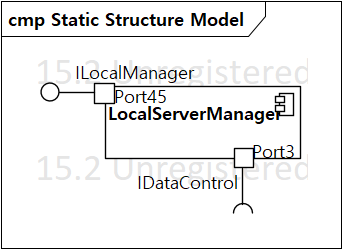
#### IHealthCheck Interface Specification



|  |  |
| --- | --- |
| Operation | Responsibility |
| echo | 출입단말기가 ping message를 받은경우 echo를 보내기 위한 오퍼레이션 |
| heartbeat | 출입단말기가 주기적으로 heartbeat신호를 전달하기 위한 오퍼레이션 |

### 출입관리시스템 – internal interface(FrontServerController)

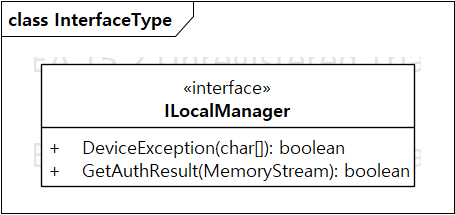
#### LocalServerManager Specification



#### Interface List

|  |  |  |
| --- | --- | --- |
| Name | Kind | Description |
| ILocalManager | Provided | LocalServerManager에서 인식된 얼굴정보의 데이터를 전송받아 Database로 인식요청을 보내기 위한 인터페이스 |

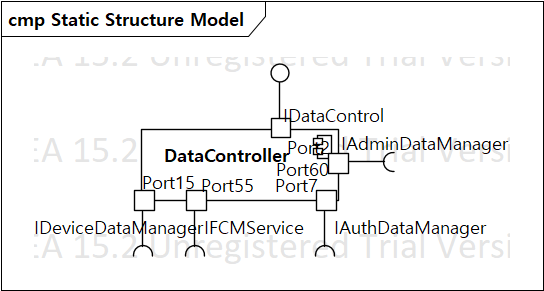
#### ILocalManager Interface Specification



|  |  |
| --- | --- |
| Operation | Responsibility |
| GetAuthResult | Database에서 인증결과를 리턴하는 오퍼레이션 |
| DevbiceException | 출입단말기에 장애발생시 처리를 위한 오퍼레이션 |

### 출입관리시스템 – business logic

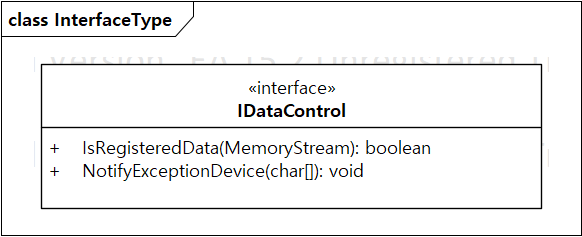
#### DataController Specification



#### Interface List

|  |  |  |
| --- | --- | --- |
| Name | Kind | Description |
| IDataControl | Provided | 방문자를 등록하고, 출입단말기 장애시 아파트관리자 정보를 읽어 메시지를 전달하는 인터페이스 |

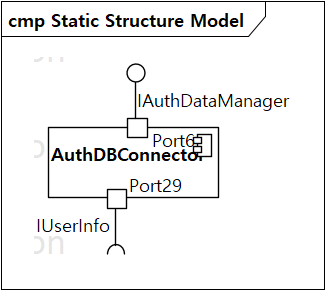
#### IDataControl Interface Specification



|  |  |
| --- | --- |
| Operation | Responsibility |
| IsRegisteredData | 전달받은 영상정보의 인증여부를 리턴하는 오퍼레이션 |
| NotifyExceptionDevice | 출입단말기의 장애발생 상황을 알리기 위한 오퍼레이션 |

### 출입관리시스템 – data

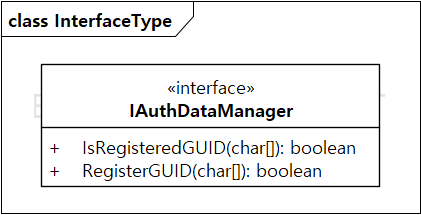
#### AuthDBConnector Specification



#### Interface List

|  |  |  |
| --- | --- | --- |
| Name | Kind | Description |
| IAuthDataManager | Provided | Database로부터 인증관련 동작을 하는 인터페이스 |

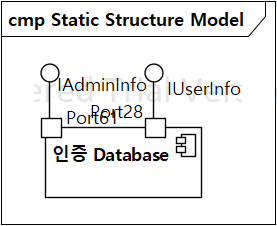
#### IAuthDataManager Interface Specification



|  |  |
| --- | --- |
| Operation | Responsibility |
| IsRegisteredGUID | 얼굴인식 및 QR코드의 해석결과인 guid가 등록되어있는지를 확인하는 오퍼레이션 |
| RegisterGUID | guid를 통해 거주자등록, 방문자등록을 동작하는 오퍼레이션 |

### Database – 단말기Database

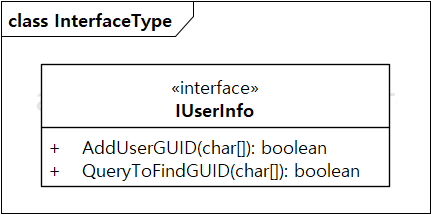
#### 인증 Database Specification



#### Interface List

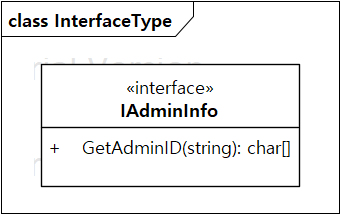
|  |  |  |
| --- | --- | --- |
| Name | Kind | Description |
| IUserInfo | Provided | 사용자 등록, 조회를 위한 인터페이스 |
| IAdminInfo | Provided | 아파트 운영자 정보를 등록, 조회하는 인터페이스 |

#### IUserInfo Interface Specification



|  |  |
| --- | --- |
| Operation | Responsibility |
| QueryToFindGUID | 특정 guid가 database에 존재하는지를 리턴하는 오퍼레이션 |
| AddUserGUID | 특정 guid를 database에 등록하기 위한 오퍼레이션 |

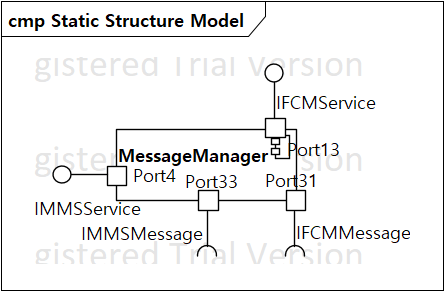
#### IAdminInfo Interface Specification



|  |  |
| --- | --- |
| Operation | Responsibility |
| GetAdminID | 아파트 id정보를 입력받아 아파트 관리자의 FCM ID를 리턴하는 오퍼레이션 |

### MessageManager – external interface

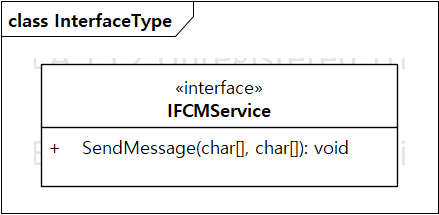
#### MessageManager Specification



#### Interface List

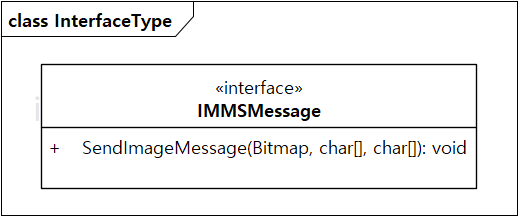
|  |  |  |
| --- | --- | --- |
| Name | Kind | Description |
| IFCMService | Provided | Push Message를 전송하기 위한 인터페이스 |
| IMMSService | Provided | MMS를 전송하기 위한 인터페이스 |

#### IFCMService Interface Specification



|  |  |
| --- | --- |
| Operation | Responsibility |
| SendMessage | FCM Message를 전송하는 오퍼레이션 |

#### IMMSService Interface Specification

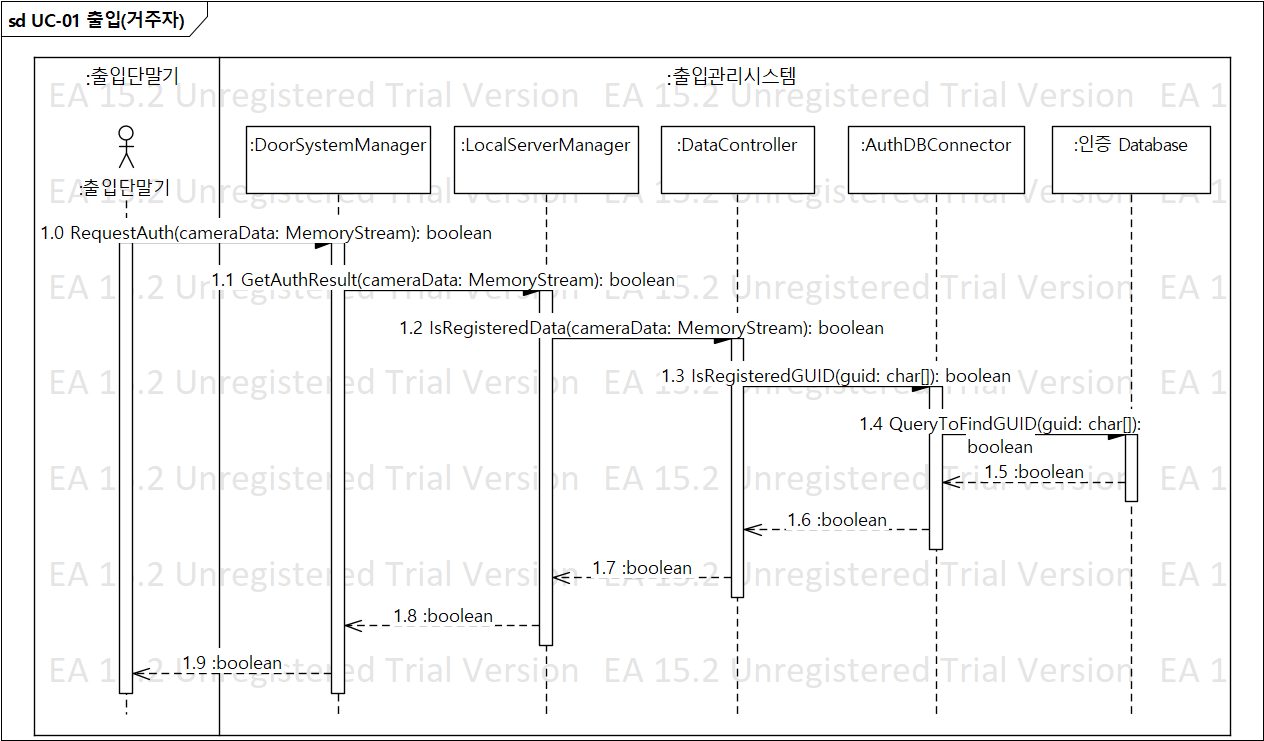


|  |  |
| --- | --- |
| Operation | Responsibility |
| SendMessage | MMS를 전송하는 오퍼레이션 |

## Behavior View

### UC-01 출입 Use Case Behavior Model

#### Behavior Diagram



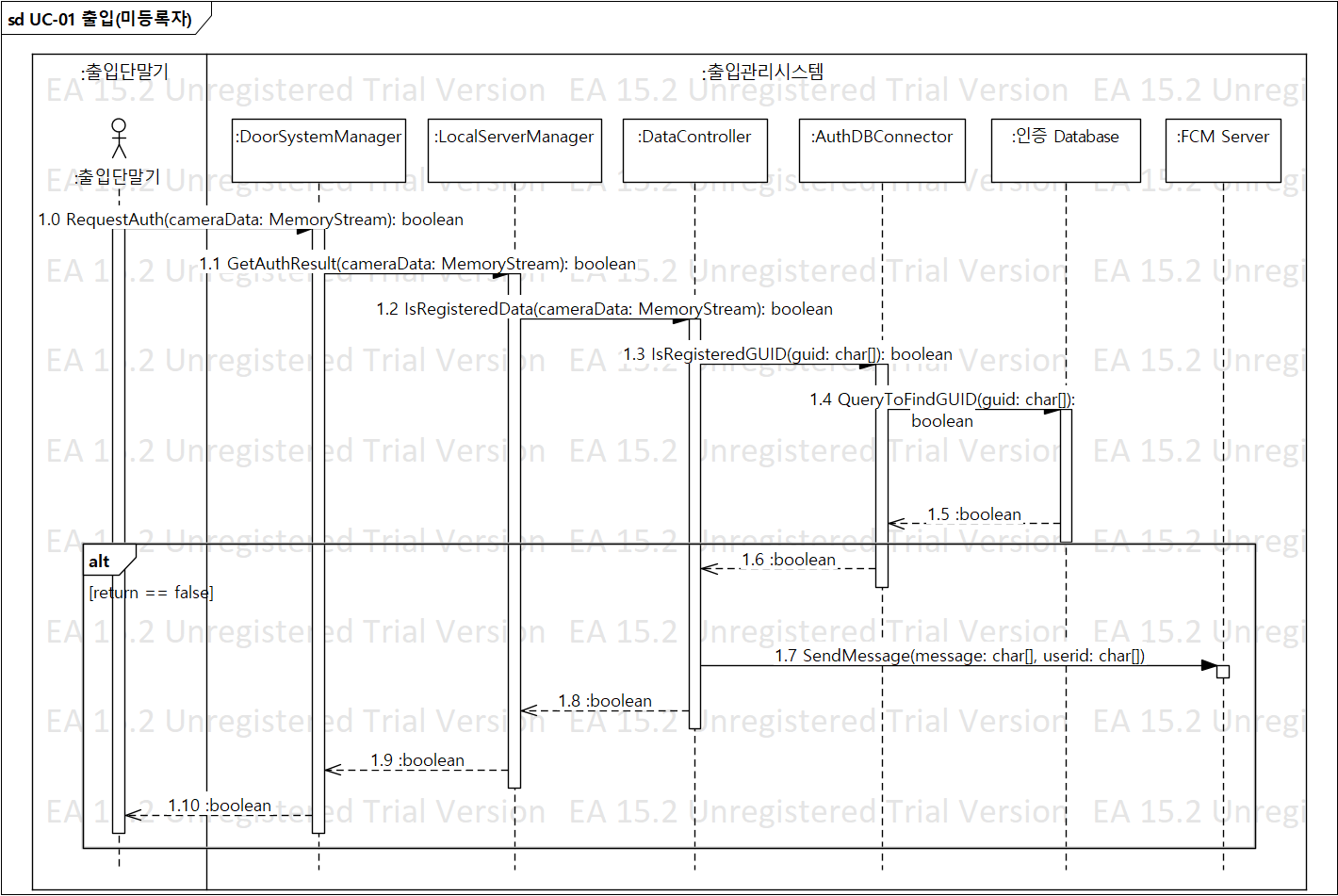
#### Behavior Description

거주자가 출입을 위해 출입단말기 앞에서 얼굴인식을 시도한다. 출입단말기의 카메라에 의해 획득된 영상정보는 출입관리시스템으로 전송되어 등록된 거주자 여부를 판단하고, 출입문 개방을 위해 출입단말기로 등록 여부를 리턴한다.

출입인증을 위한 데이터는 분산된 Local Server로 전달되며 Local Server는 해당 영상의 얼굴인식 매커니즘을 거쳐 GUID형태로 출입관리시스템의 Main 서버로 전송하며, 이를 기반으로 Database에 등록 여부를 판단하게 된다.

|  |  |
| --- | --- |
| UC Step | Behavior |
| 1. 거주자가 출입을 위해 공동현관 출입 단말기 앞에 도착한다. | 1.0 메시지를 통해 출입단말기 카메라의 영상을 출입관리시스템으로 전송한다. |
| 1. 출입 단말기는 카메라의 영상정보를 출입관리시스템으로 전송한다. |
| 1. 출입관리시스템은 전달받은 영상정보에서 얼굴정보를 확인한다. | 1.1, 1.2 메시지를 통해 영상정보에서 얼굴정보를 GUID형식으로 전환한다. |
| 1. 출입관리시스템은 얼굴정보의 등록여부를 판단한다. | 1.3, 1.4 메시지를 통해 등록된 얼굴인지를 Database를 통해 확인한다. |
| 1. 출입관리시스템은 등록된 사용자임을 확인하고 출입단말기로 출입허가 명령을 보낸다. | 1.5-1.9 메시지를 통해 인증정보를 전달받아 출입허가를 회신한다. |

#### Behavior Diagram

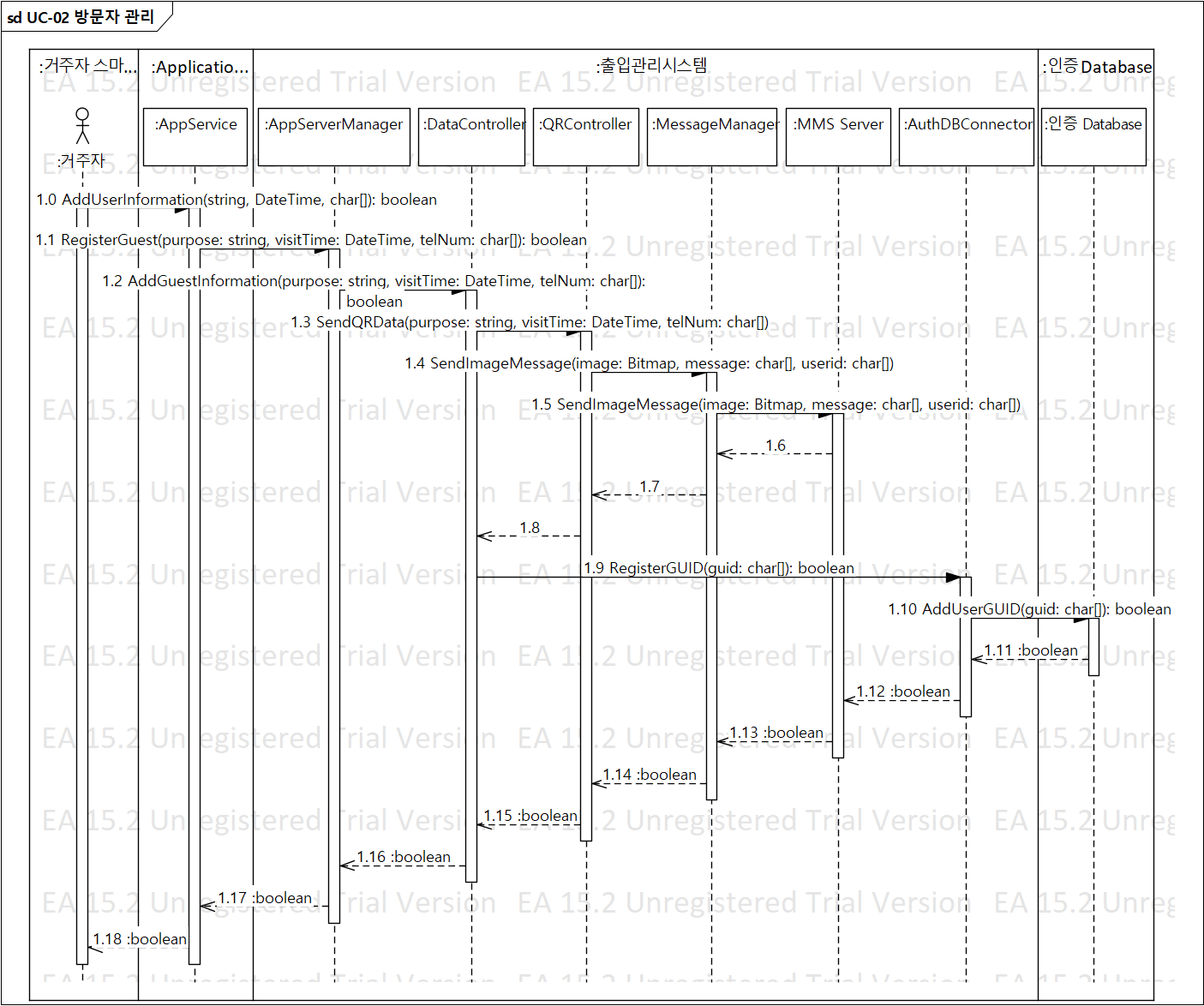


#### Behavior Description

|  |  |
| --- | --- |
| UC Step | Behavior |
| 1. 미등록자가 출입을 위해 공동현관 출입 단말기 앞에 도착한다. | 1.0 Message를 통해 출입단말기 카메라의 영상을 출입관리시스템으로 전송한다. |
| 1. 출입 단말기는 카메라의 영상정보를 출입관리시스템으로 전송한다. |
| 1. 출입관리시스템은 전달받은 영상정보에서 얼굴정보를 확인한다. | 1.1, 1.2 메시지를 통해 영상정보에서 얼굴정보를 GUID형식으로 전환한다. |
| 1. 출입관리시스템은 얼굴정보의 등록여부를 판단한다. | 1.3, 1.4 메시지를 통해 등록된 얼굴인지를 Database를 통해 확인한다. |
| 1. 출입관리시스템은 미등록된 사용자임을 확인하고 출입단말기로 출입불가 명령을 보낸다. | 1.8-1.10 메시지를 통해 인증정보를 전달받아 출입 가능여부를 회신한다. |
| 1. 출입관리시스템은 아파트 관리자에게 침입이 탐지되었음을 Push Message로 전송한다. | 1.7 메시지를 통해 Push Message를 전송한다. |

### UC-02 방문자 관리 Use Case Behavior Model

#### Behavior Diagram



#### Behavior Description

거주자는 스마트폰 어플리케이션을 통해 방문자를 등록할 수 있다.

스마트폰 어플리케이션에서 전달되는 정보는 App Server를 통해 처리가 진행되며, App Server는 출입관리시스템을 통해서 입력된 정보를 Database에 등록한다.

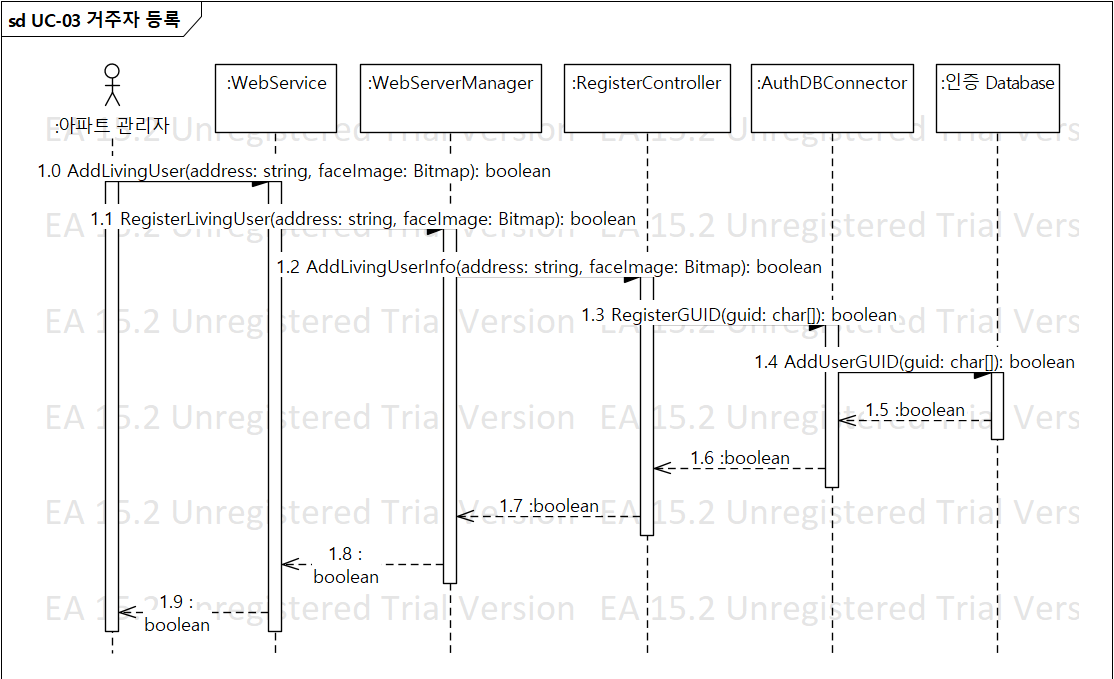
거주자가 스마트폰 어플리케이션을 통해 방문자를 등록하면, 해당 정보를 통해 QR코드를 생성하고 생성된 QR코드 이미지를 이미지 포멧으로 MMS를 전송한다.

또한 입력된 정보는 방문자 출입인증을 위해 인증 Database에 저장한다.

|  |  |
| --- | --- |
| UC Step | Behavior |
| 1. 거주자는 스마트폰 어플리케이션을 통해 방문자 정보(방문시간/목적/전화번호)를 입력한다. | 1.0, 1.1, 1.2 메시지를 통해 입력된 정보가 Web Server로 전송된다. |
| 1. 출입관리시스템은 전달받은 방문자정보를 통해 QR코드 이미지를 생성한다. | 1.3, 1.4 메시지를 통해 QR이미지가 생성된다. |
| 1. 출입관리시스템은 QR코드 이미지를 방문자에게 문자메시지로 전달한다. | 1.5 메시지를 통해 생성된 QR이미지가 전송된다. |

### UC-03 거주자 등록 Use Case Behavior Model

#### Behavior Diagram



#### Behavior Description

아파트 관리자는 해당 아파트에 거주하는 주민을 아파트 관리자 콘솔을 통해서 입력한다.

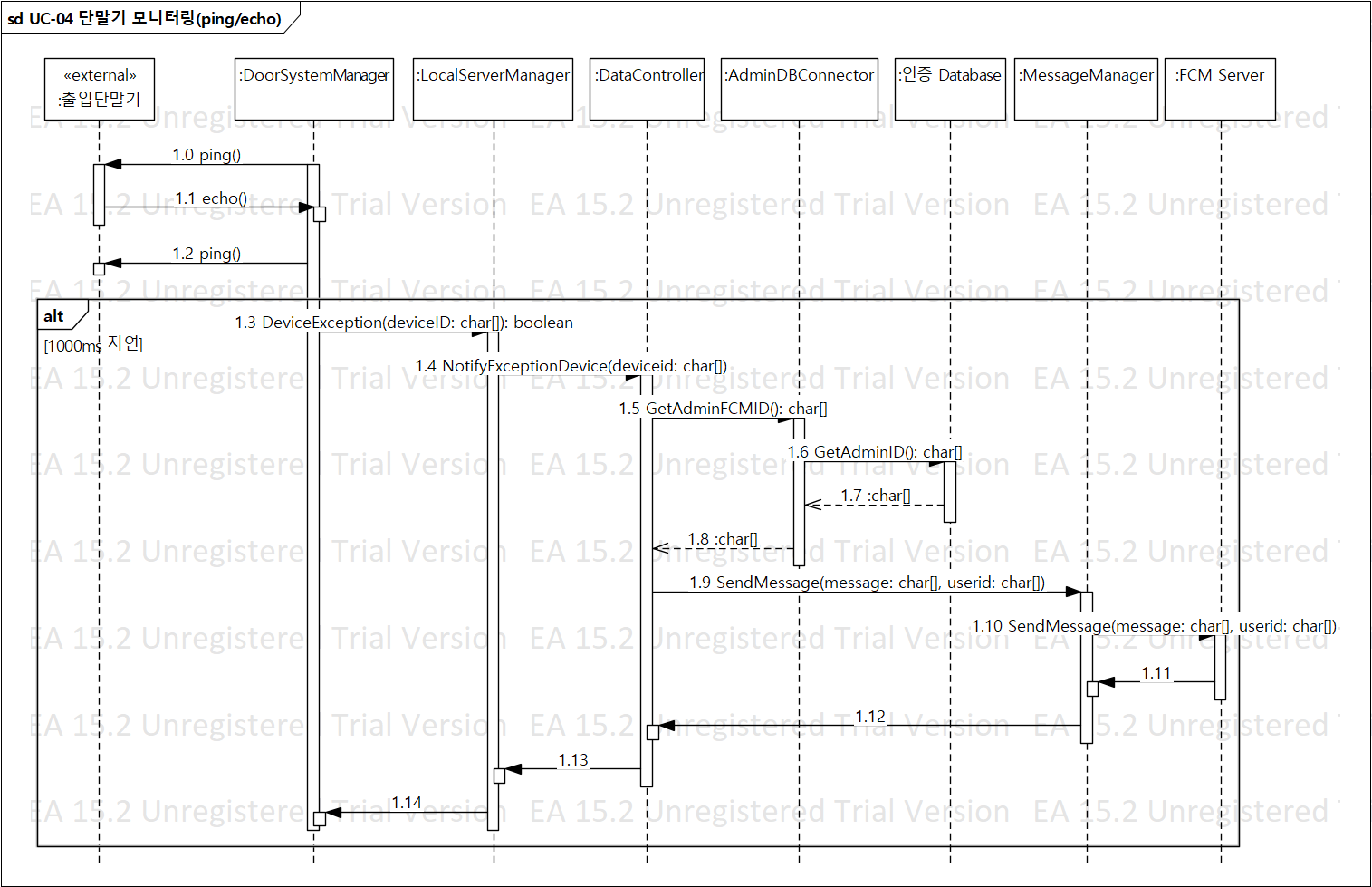
거주자로부터 동/호수, 얼굴사진 데이터를 받아 웹사이트를 통해 입력이 가능하다.

웹사이트의 등록페이지에서 거주자 정보를 입력하면, Web Server에서 처리가 진행되며, Web Server는 출입관리시스템을 통해 Database에 정보를 저장한다.

|  |  |
| --- | --- |
| UC Step | Behavior |
| 1. 아파트 관리자는 아파트 관리자 콘솔에서 거주자 등록 페이지에 접근한다. | 1.0, 1.1, 1.2, 1.3 메시지를 통해 입력된 거주자 정보는 Database에 입력을 요청한다. |
| 1. 아파트 관리자는 거주자 등록페이지에서 거주자 정보(동/호수, 얼굴사진)를 입력한다. |
| 1. 출입관리시스템은 전달받은 거주자 정보를 출입가능 사용자로 등록한다. | 1.4 메시지를 통해 Database에 거주자로 등록한다. |

### UC-04 단말기 모니터링 Use Case Behavior Model

#### Behavior Diagram



#### Behavior Description

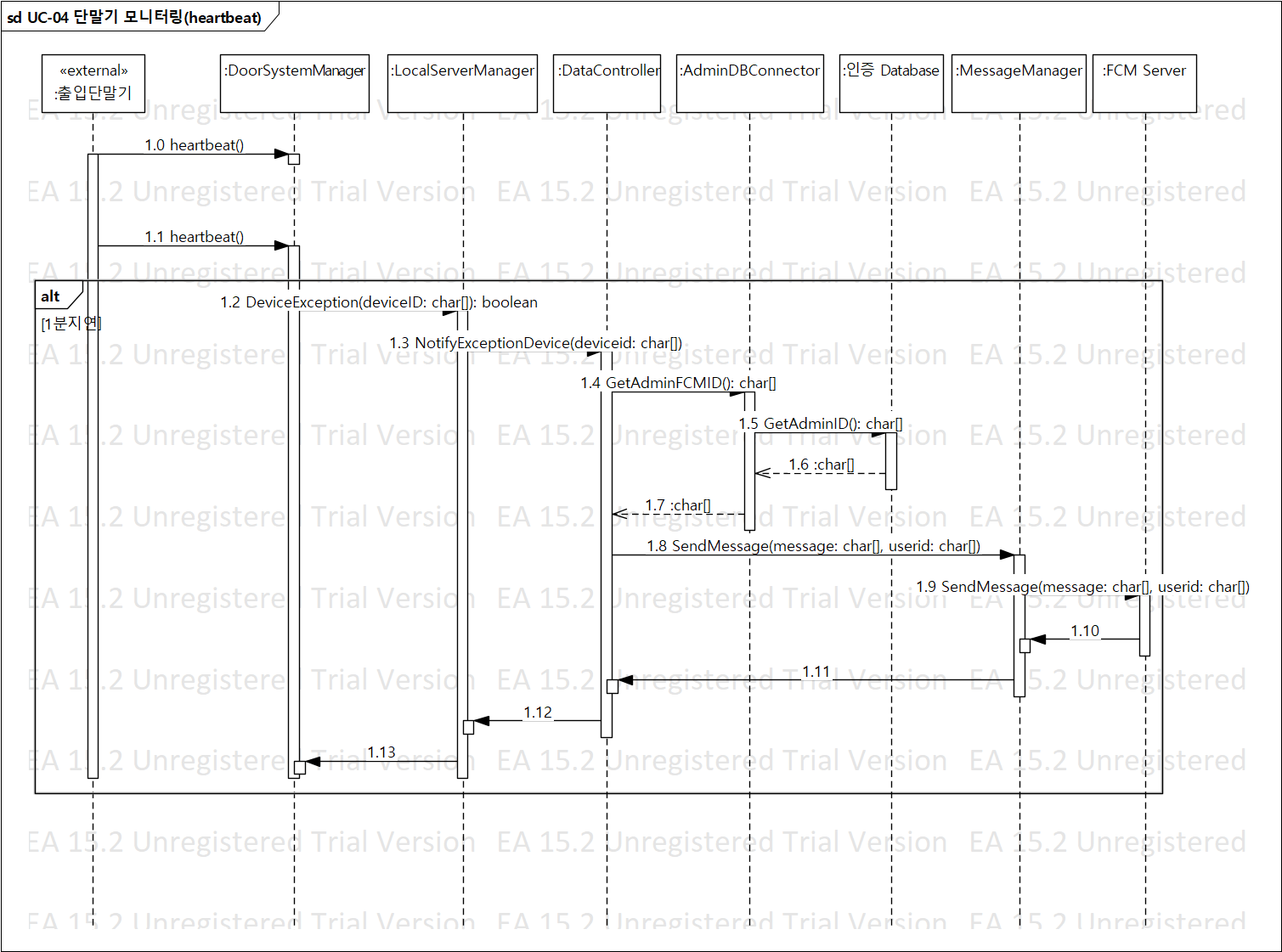
출입단말기는 제조사에 따라 지원하는 가용성 판단 인터페이스가 달라진다.

위 Sequence Diagram은 ping/echo방식의 출입단말기에 대한 가용성 판단 Sequence이다.

ping메시지를 호출한 뒤, 1초이상 echo신호를 받지 못하는 경우 단말기 장애로 판단하며, 단말기 장애가 발생하는 경우 아파트 관리자에게 push message를 전송하여 장애상황을 알려준다.

|  |  |
| --- | --- |
| UC Step | Behavior |
| 1. 출입단말시스템의 모니터링 컴포넌트가 출입단말기의 정상동작 여부를 확인한다. | 1.0, 1.1 메시지를 통해 출입단말기가 정상상태인지를 판단한다. |
| 1. 출입단말기의 장애가 탐지되면 아파트 관리자에게 통보한다. | 1.2 메시지를 보낸 뒤 echo신호가 전달되지 않는경우 1.3-1.10 메시지를 통해 아파트 관리자에게 Push Message를 전송한다. |

#### Behavior Diagram



#### Behavior Description

출입단말기는 제조사에 따라 지원하는 가용성 판단 인터페이스가 달라진다.

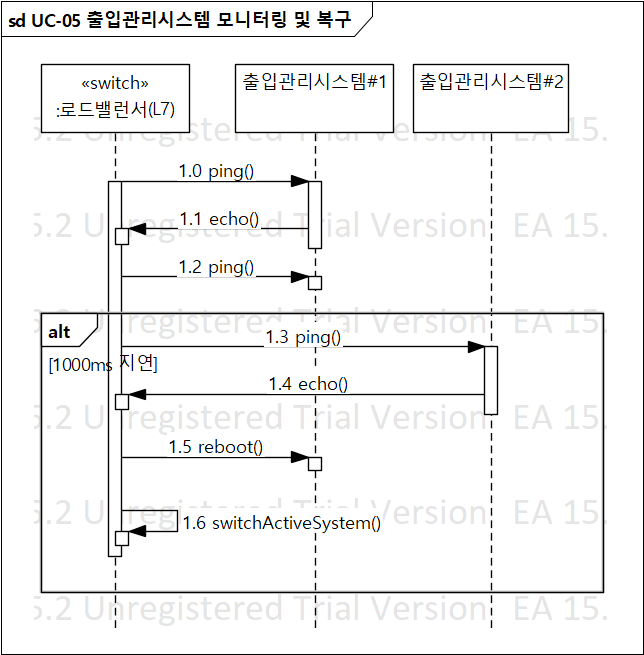
위 Sequence Diagram은 heartbeat방식의 출입단말기에 대한 가용성 판단 Sequence이다.

Heartbeat의 주기는 출입단말기 스펙에 따라 달라진다. 타겟단말기의 주기에서 1분이상 heartbeat신호를 받지 못하는 경우 단말기 장애로 판단하며, 단말기 장애가 발생하는 경우 아파트 관리자에게 push message를 전송하여 장애상황을 알려준다

|  |  |
| --- | --- |
| UC Step | Behavior |
| 1. 출입단말시스템의 모니터링 컴포넌트가 출입단말기의 정상동작 여부를 확인한다. | 1.0, 1.1 메시지를 통해 출입단말기가 정상상태인지를 판단한다. |
| 1. 출입단말기의 장애가 탐지되면 아파트 관리자에게 통보한다. | 일정시간동안 Heartbeat신호가 전달되지 않는경우 1.2-1.9 메시지를 통해 아파트 관리자에게 Push Message를 전송한다. |

### UC-05 출입관리시스템 모니터링 및 복구 Use Case Behavior Model

#### Behavior Diagram



#### Behavior Description

출입관리시스템을 구성하는 여러 종류의 서버는 가용성을 확보하기 위하여 2개 이상으로 이루어져 있으며 모두 로드밸런서를 가지고 있다.

로드밸런서는 AWS의 EC2를 사용하며, ping/echo 방식의 가용성 판단로직을 가지고 있다.

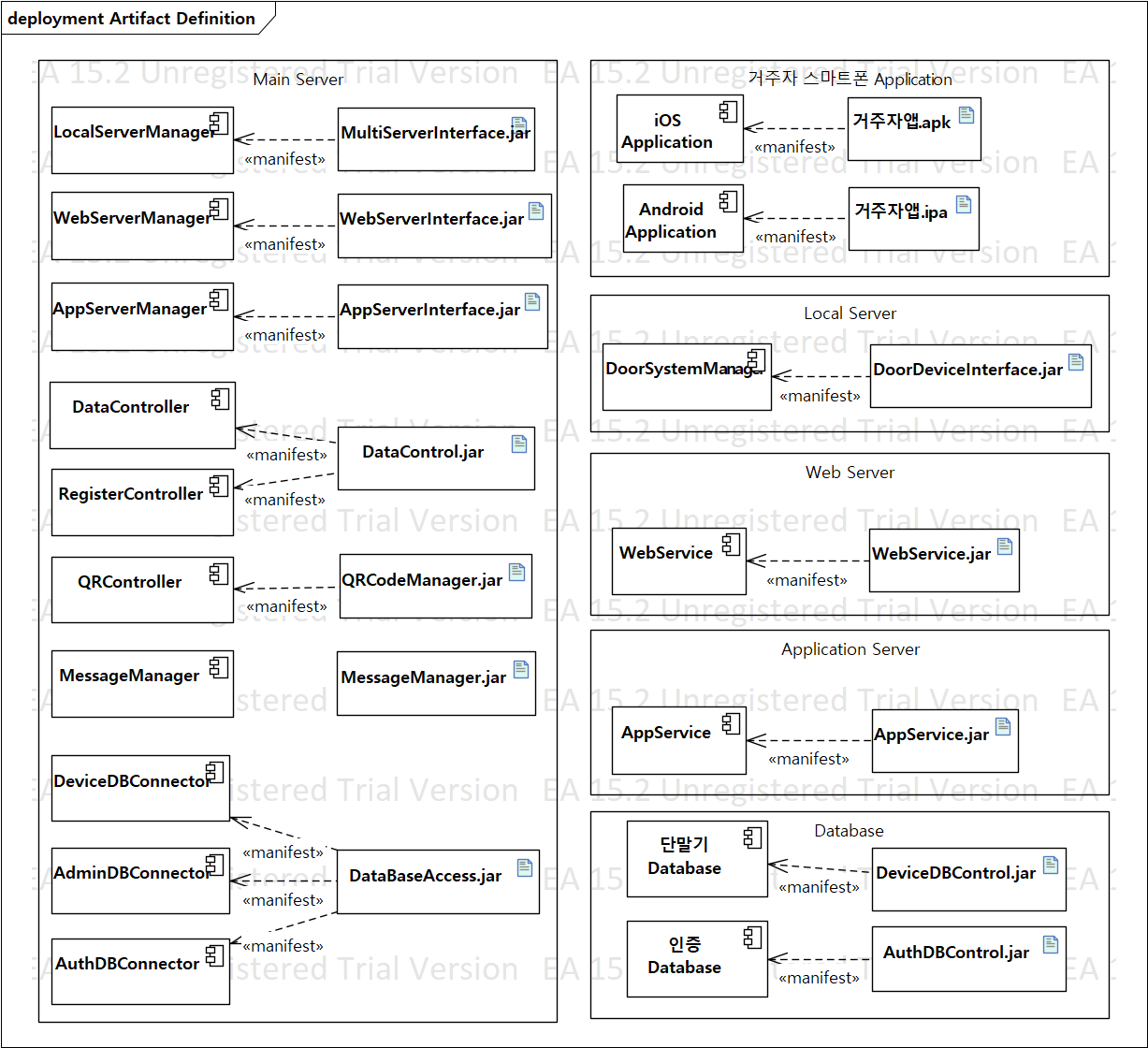
서버 장애가 탐지되면 대체서버로 역할을 이동함과 동시에 장애발생 서버는 재부팅을 진행한다.

|  |  |
| --- | --- |
| UC Step | Behavior |
| 1. 시스템 모니터링 타이머는 출입관리시스템의 정상동작 여부를 확인하기 위한 ping명령를 송신한다. | 1.0, 1.1 메시지를 통해 출입단말기가 정상상태인지를 판단한다. |
| 1. 시스템 모니터링 타이머는 ping명령을 송신한 후 일정시간 대기한다. |
| 1. 시스템 모니터링 타이머는 일정시간 echo명령을 받지 못한 경우 서버의 장애로 판단한다. | 1.2 메시지를 보낸 뒤 echo신호가 전달되지 않는경우 1.3-1.10 메시지를 통해 아파트 관리자에게 Push Message를 전송한다. |
| 1. 시스템 모니터링 타이머는 장애로 판단된 서버는 재시작을 명령하며, 대체서버로 역할을 변경한다. | 1.5 메시지를 통해 장애발생 서버에 재부팅을 명령하며, 1.6 메시지를 통해 대체서버로 역할을 변경한다. |

## Deployment View

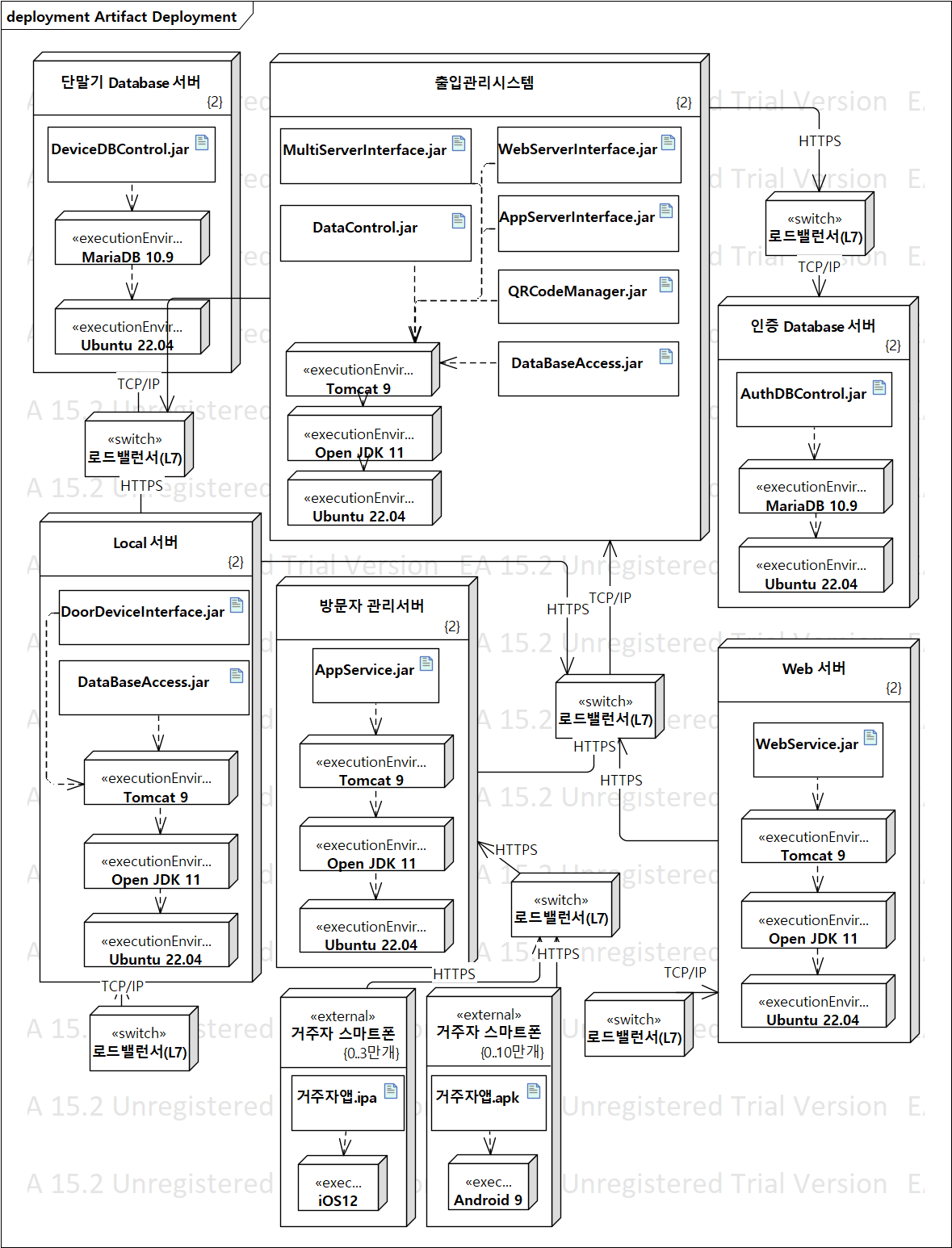
### Artifact Definition Model

#### Artifact Definition Diagram



### Artifact Deployment Model

#### Artifact Deployment Diagram



## Documenting Design Decisions

### Design Decision List

|  |  |  |
| --- | --- | --- |
| ID | Title | 목표 QAs |
| DD-01 | 얼굴인식 정확도 확보를 위한 알고리즘 다양화(정확도, 변경용이성) | QA-01(출입 인가 처리 정확성) |
| DD-02 | 출입관리시스템의 처리부하를 막고 안정성 확보를 위한 서버 구성 방식 (성능, 가용성) | QA-02(침입 시도 통보 시간)  QA-03(시스템 모니터링 및 복구) |
| DD-03 | 단말기 정상동작 판단을 위한 모니터링 방안(가용성) | QA-04(단말기 이상 모니터링) |
| DD-04 | 출입단말기 연동방식 (유지보수성) | QA-05(새로운 출입 단말기 지원) |

### DD-01 얼굴인식 정확도 확보를 위한 알고리즘 다양화(정확도, 변경용이성)

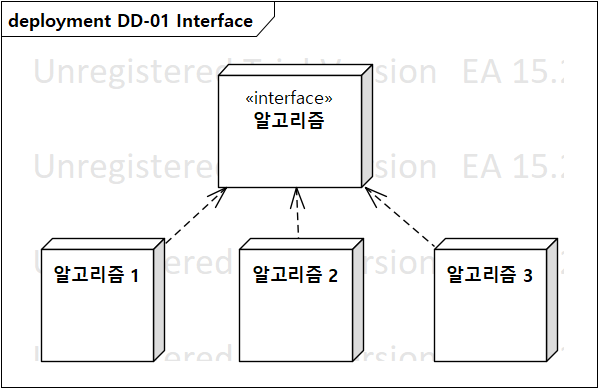
#### Design Goal

얼굴인식의 인식률을 향상시키기 위해, 단일 알고리즘의 사용 보다는 여러가지 알고리즘의 결과를 조합하는 방법을 채택하였다.

이를 위해 알고리즘 모듈의 크기를 최소화 하여, 단일책임을 부여하도록 설계하였으며 알고리즘 로직을 인터페이스화 하는 작업을 진행한다.

#### Design Approach List

#### Design Approach #1 Description: 알고리즘 모듈 크기 감소 및 인터페이스화



각각의 알고리즘은 interface를 상속받아 작성되며, interface를 사용하는 주체에서는 필요에 따라 각 알고리즘으로 의존성을 주입하여 사용이 가능하다.

#### Design Approach #2 Description: 3가지 알고리즘 결과 조합

얼굴인식 결과에 대하여 세가지의 알고리즘으로 인식 결과를 도출한다.

세가지 알고리즘이 모두 다른 결과를 도출하는 경우를 대비하여 정확도에 따라 우선순위를 정한다. 결과값이 가장 높은 얼굴이 인식 결과가 된다.

#### Design Approach #3 Description: 알고리즘 모듈 지연바인딩

알고리즘을 수행하는 interface만 객체화 되어있고, 알고리즘이 수행되는 과정에서 상황에 따라 의존성이 주입되어 로직을 수행하는 방법이다. 영상의 품질 및 환경에 따라 적합한 알고리즘으로 얼굴인식을 진행할 수 있다.

#### Decision and Rationale

본 시스템에서는 #1, #2의 택틱을 채택하여 **알고리즘의 인터페이스 화를 진행한 후 3가지 알고리즘의 결과를 조합**하여 얼굴인식을 진행하도록 설계하였다.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Quality Attribute | | Analysis | 알고리즘 모듈 크기 감소 및 인터페이스화 | 3가지 알고리즘 결과 조합 | 알고리즘 모듈 지연바인딩 |
| ID | Title |
|  | 정확도 | Pros/Cons | (0)정확도에 영향을 미치지 않는다. | (++)얼굴인식을 여러 알고리즘의 결과를 조합하므로 정확도 향상이 기대됨 | (0)정확도에 영향을 미치지 않는다. |
|  | 성능 | Pros/Cons | (0)성능에 영향을 미치지 않는다. | (-)얼굴인식 로직에 대하여 3가지 알고리즘이 모두 실행되어야 하므로 성능이 저하될 수 있다. | (-)런타임에 의존성이 주입되므로, 성능이 저하된다. |
|  | 변경용이성 | Pros/Cons | (+)추후 추가적인 알고리즘 적용이 용이한 설계이다. | (0)영향없음 | (+)지연바인딩을 위한 설계로 의존성이 낮아지므로 추가적인 알고리즘 모듈 적용이 용이해진다. |

### DD-02 출입관리시스템의 처리부하를 막고 안정성 확보를 위한 서버 구성 방식 (성능, 가용성)

#### Design Goal

출입관리시스템은 시스템 구축을 통해 용인시 전체의 아파트에 적용될 수 있도록 설계해야 한다. 확장 가능성을 고려하여 출입단말기를 대응할 서버가 유동적으로 늘어날 수 있는 방향으로 설계를 할 필요가 있다.

단일서버만으로 늘어나는 출입단말기의 부하를 견디기는 어려울 것으로 예상하였다. 따라서 AWS의 EC2를 로드밸런서로 도입하는 것으로 결정하였다.

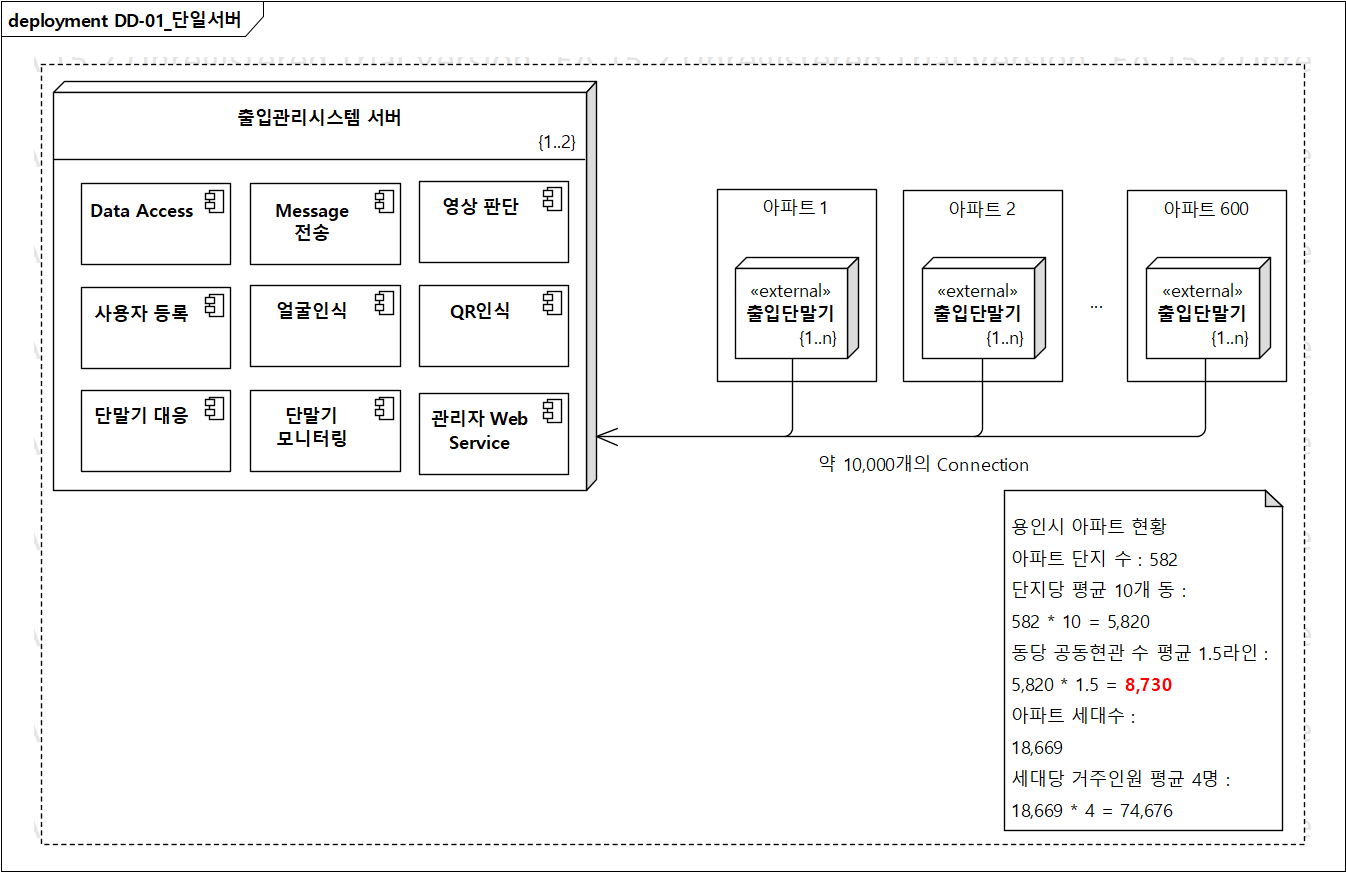
AWS EC2를 사용하면 하드웨어에 선 투자할 필요가 없어 더 빠르게 애플리케이션을 개발하고 배포할 수 있는 장점이 존재한다.

AWS EC2를 각 서버에 도입하여 1) 출입관련 프로세스의 성능 을 보장할 수 있으며, 2) 서버의 장애발생시 가용성을 확보 할 수 있는 장점이 있다.

또한 3) 출입관리시스템의 Main Server와 Multi Local Server를 분리함으로써, 서버에 부하가 있을만한 로직에 대하여 로드밸런싱을 하고, Main Server는 데이터 관련 로직만 수행하도록 하였다.

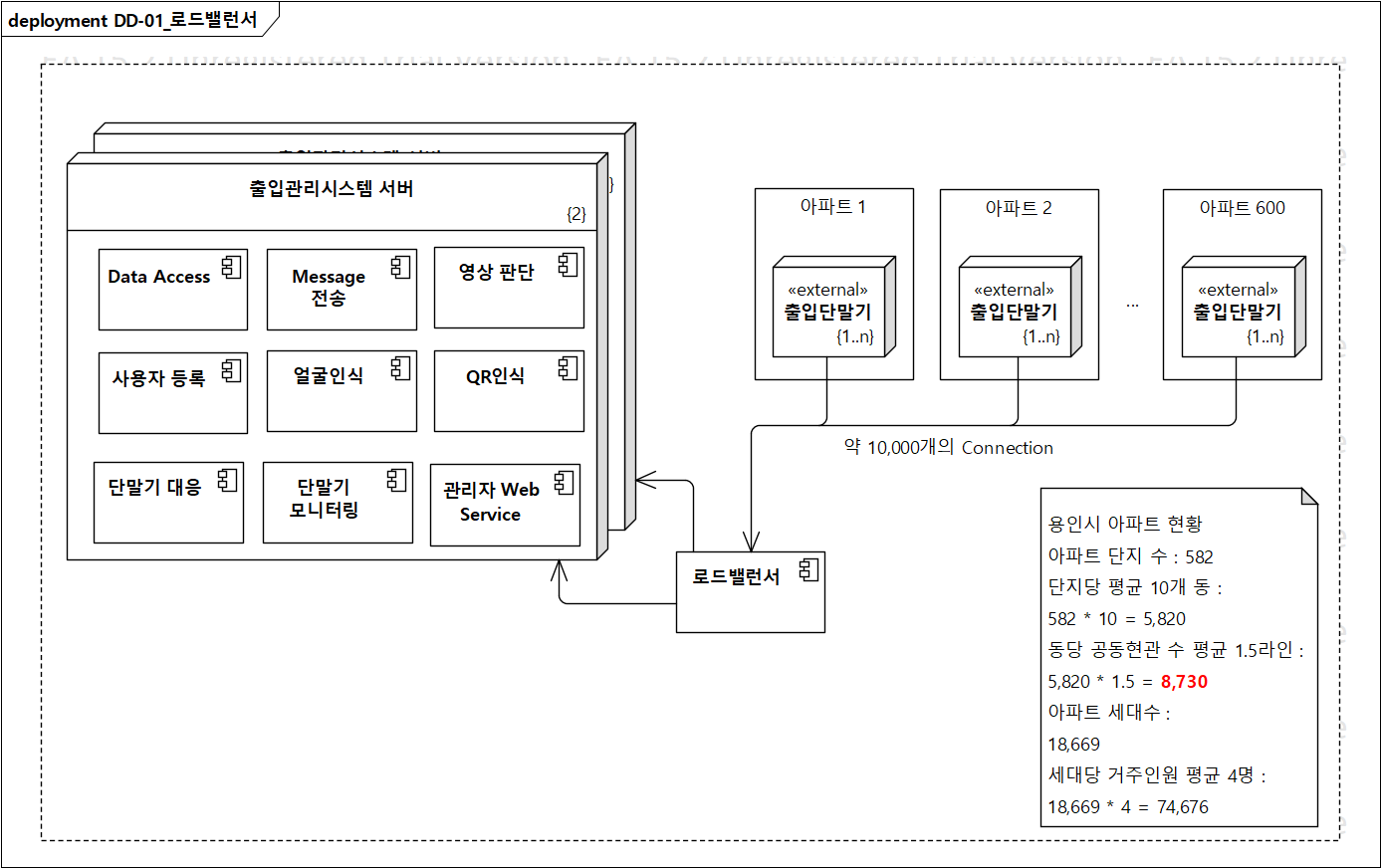
#### Design Approach List

#### Design Approach #1 Description: 단일서버



최초 설계된 하나의 출입관리시스템 서버를 이용해서 모든 아파트의 출입단말기를 제어하는 방식이다. 동시에 들어오는 요청에 대한 부하를 방지할 수 없으며, 서버 장애 발생에 취약한 구조를 가지고 있었다.

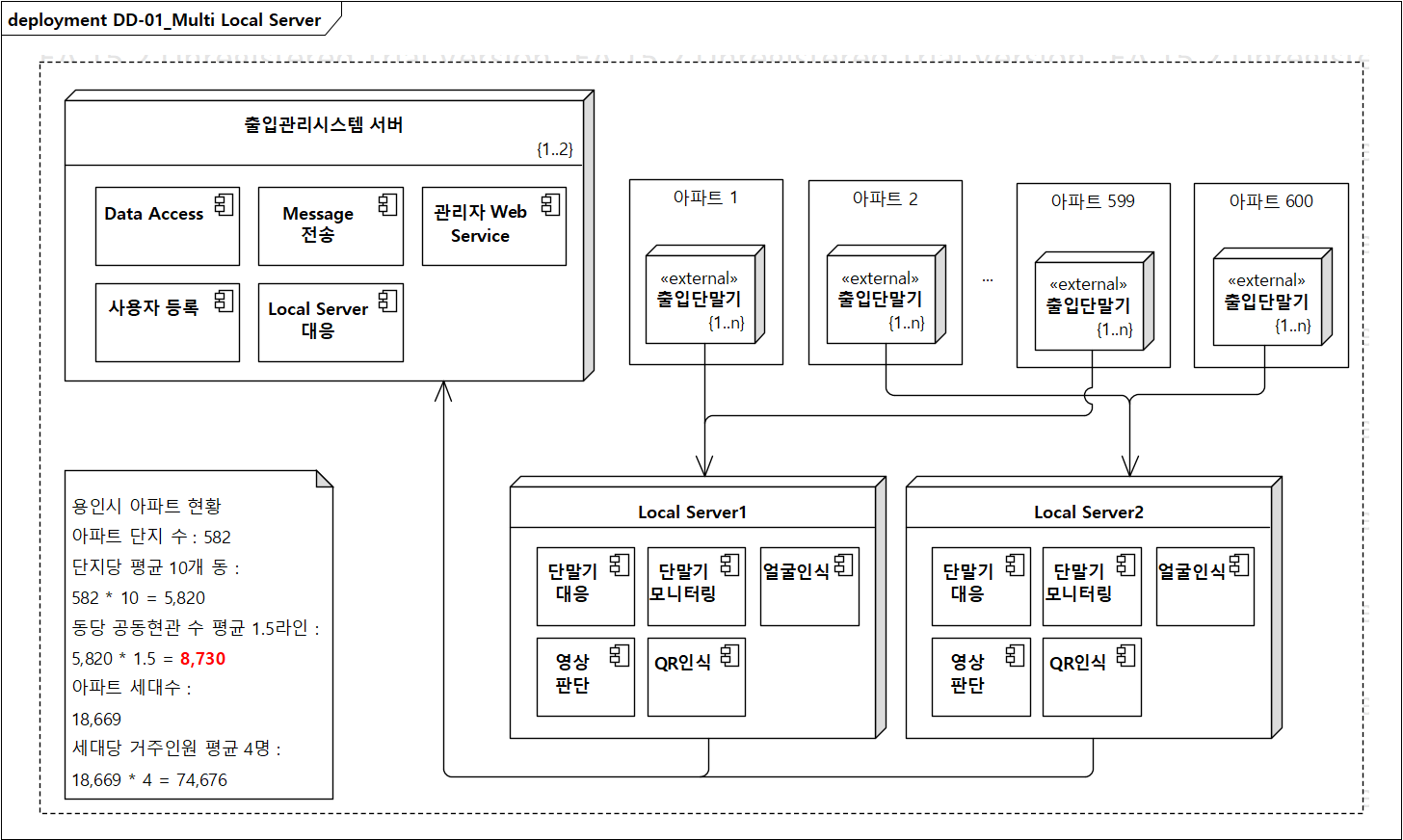
#### Design Approach #2 Description: 로드 밸런서



출입관리시스템 서버를 2개로 변경하고 로드밸런서를 중간에 배치한 모습이다.

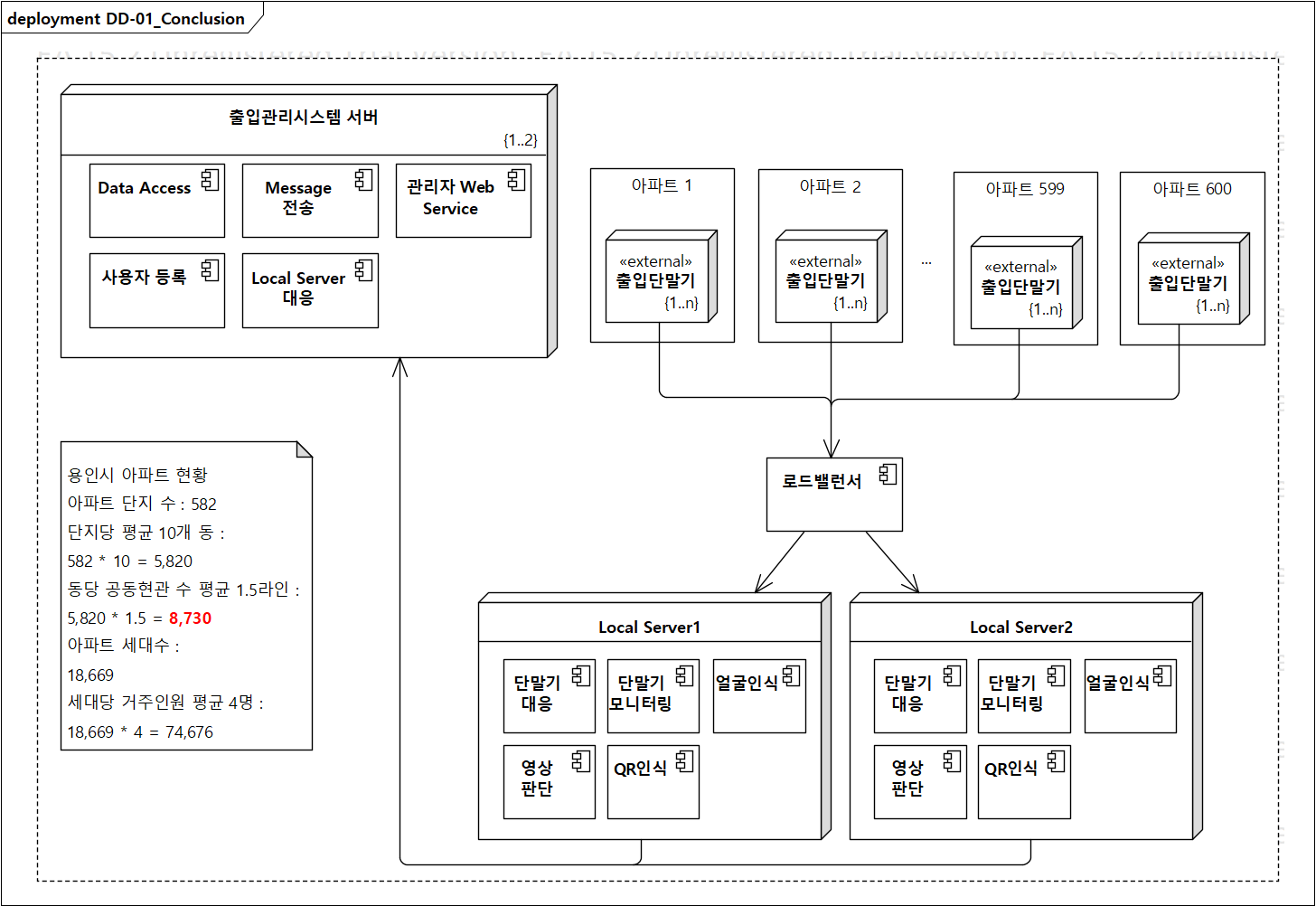
2개의 출입관리시스템으로 인하여 가용성은 확보할 수 있지만, 출입단말기가 증가할 경우 서버의 부하에 대한 대응은 되지 않는다.

#### Design Approach #3 Description: Multi Local Server



출입관리시스템과 로컬서버를 분리하여 설계한 모습이다. 서버에서 부하가 될만한 얼굴인식 관련 컴포넌트는 로컬서버에서 처리하도록 분리하여, 출입단말기가 증가할 때 로컬서버의 확장으로 해결할 수 있도록 하였다.

#### Decision and Rationale



결론적으로, 출입관리시스템의 Main Server와 Multi Local Server를 나누고 모든 Multi Local Server는 AWS EC2를 도입한 로드밸런서를 통해 부하를 분산할 수 있는 구조로 설계하였다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Quality Attribute | | Analysis | 로드밸런서 + Multi Local Server  **(Selected)** | 단일서버 |
| ID | Title |
| QA-02 | 침입 시도 통보 시간 | Pros/Cons | (+) 출입관리시스템의 부하가 줄어들면, 침입을 탐지하는 시간이 빨라져 통보시간이 빨라질 수 있다. | (-) 서버에 부하가 걸릴수록 얼굴인식에 걸리는 시간이 증가하여 침입을 탐지하고 통보하는 시간이 늘어난다. |
| QA-03 | 시스템 이상 발생시 대체가능 서버 활용 | Pros/Cons | (+) 로드밸런서가 시스템 이상을 감지할 수 있으며, 서버 재가동 및 대체서버로 전환이 용이하다. | (-) 시스템 이상 발생시 대체서버가 없으며, 이상발생을 감지하기 위한 별도의 서버가 필요하다. |
| QA-05 | 새로운 출입 단말기 지원 | Pros/Cons | (+) 로드밸런서로 인하여 새로운 출입단말기 지원시 서버의 증설 및 부하관리가 용이하다. | (-) 새로운 출입단말기가 지원될수록 서버에 부하가 늘어난다. |

### DD-04 단말기 정상동작 판단을 위한 모니터링 방안

#### Design Goal

출입관리시스템은 정상동작이 가능한 환경을 확인하기 위하여 각 단말기의 동작상태를 모니터링 한다.

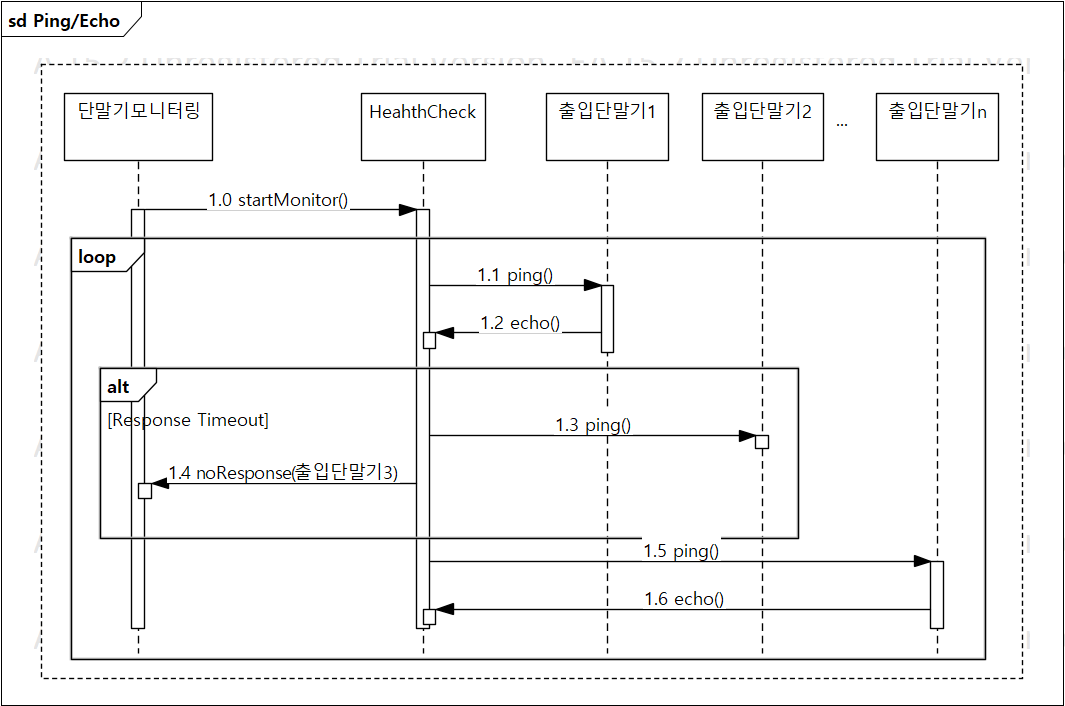
출입단말기는 모델에 따라 가용여부 판단을 위한 인터페이스가 다르게 구성되어 있을 수 있다.

출입관리시스템에서 출입단말기의 가용여부를 판단하기 위해서는 각 출입단말기가 제공하는 여러가지 방식의 테스트 방식을 모두 지원할 수 있어야 한다.

따라서 대표적인 Ping/Echo방식과 Heartbeat방식을 모두 지원하는 방법을 채택하였다.

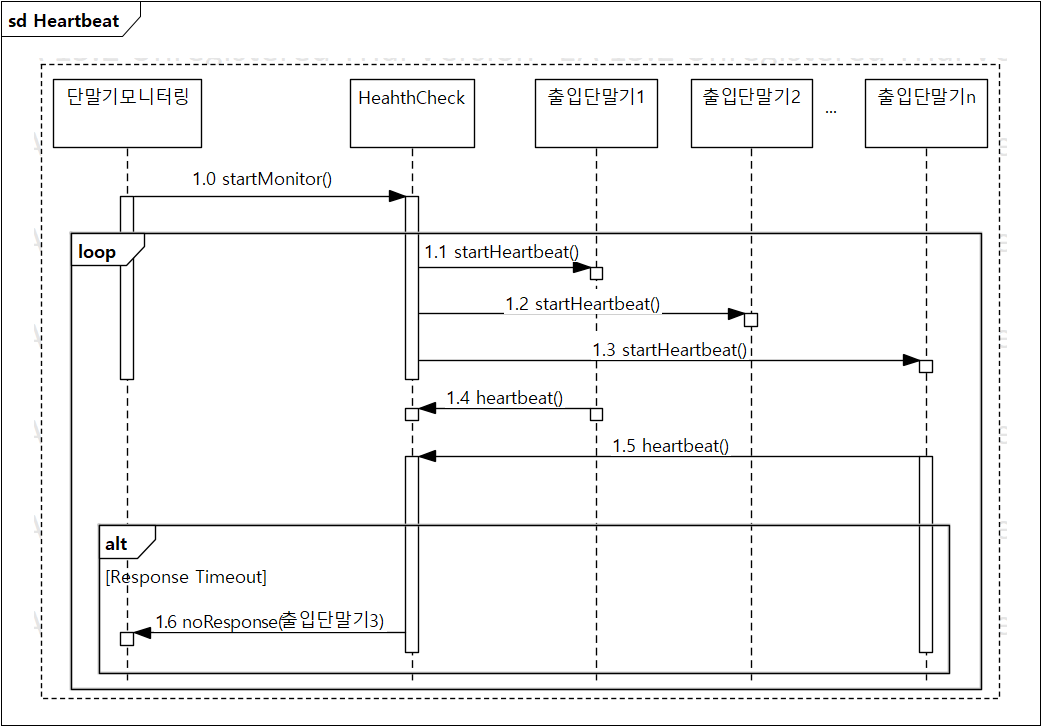
Ping/Echo 방식과 Heartbeat방식은 다음과 같은 특징을 가지고 있다.

#### Design Approach #1 Description: Ping/Echo



최초 설계된 하나의 출입관리시스템 서버를 이용해서 모든 아파트의 출입단말기를 제어하는 방식이다. 동시에 들어오는 요청에 대한 부하를 방지할 수 없으며, 서버 장애 발생에 취약한 구조를 가지고 있었다.

#### Design Approach #1 Description: Ping/Echo



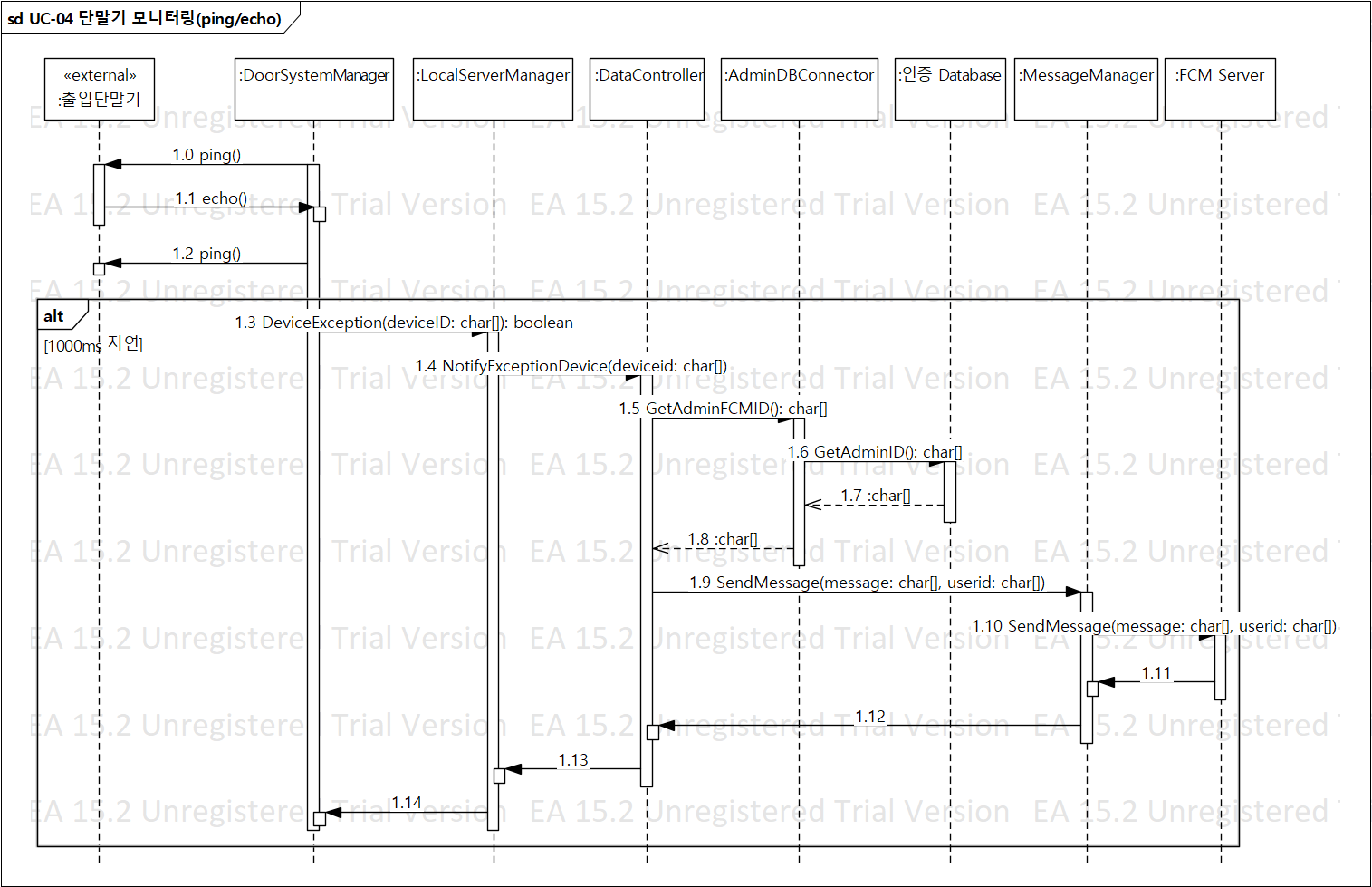
최초 설계된 하나의 출입관리시스템 서버를 이용해서 모든 아파트의 출입단말기를 제어하는 방식이다. 동시에 들어오는 요청에 대한 부하를 방지할 수 없으며, 서버 장애 발생에 취약한 구조를 가지고 있었다.

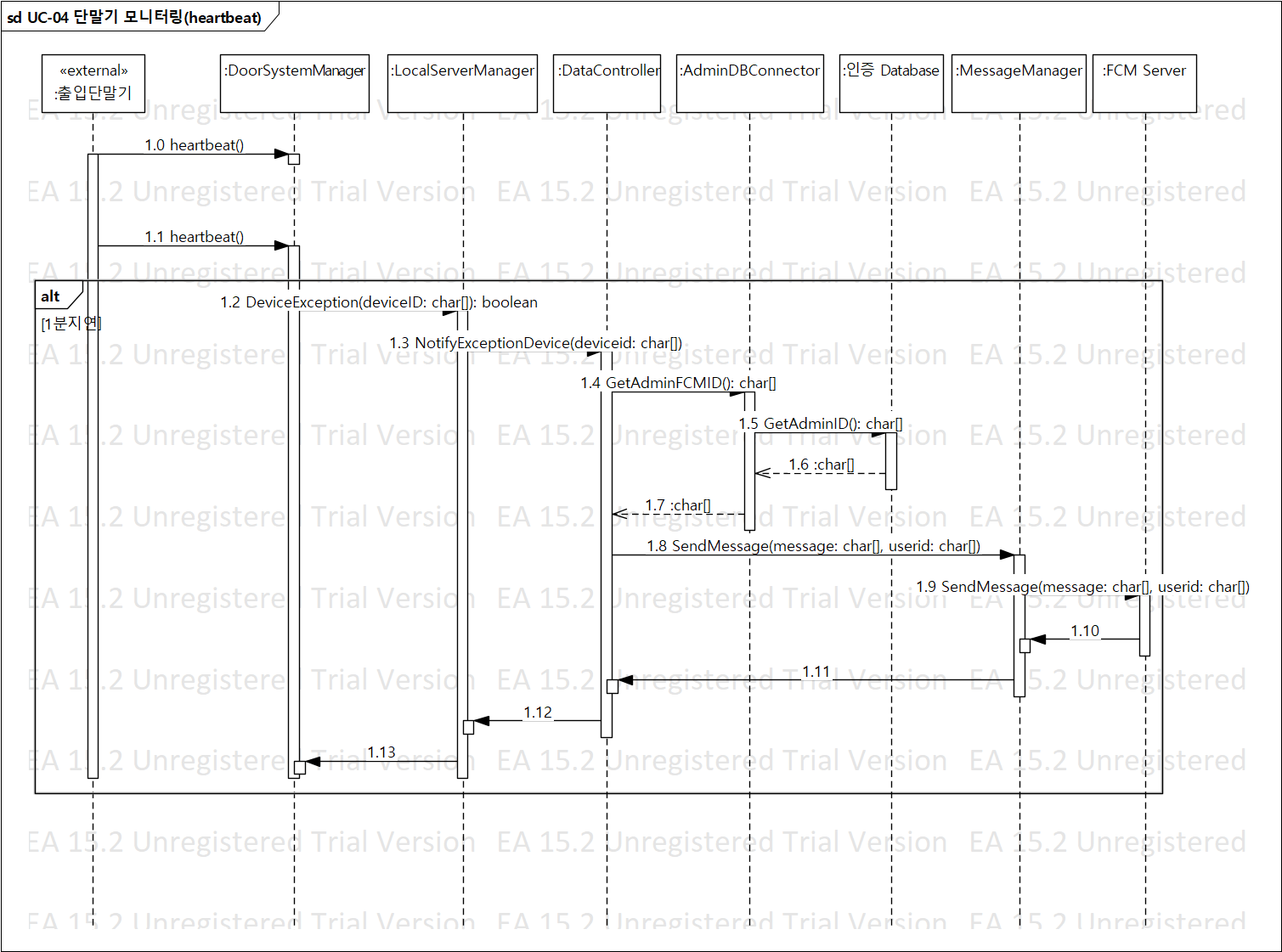
#### Design Approach #1 Description: 예외 발생

예외 발생 Tactic은 출입단말기의 Exception을 컨트롤하여 이상발생을 탐지하여야 하므로, External Device에 관련된 구현사항은 본 시스템에서는 고려대상이 되지 않았다.

#### Decision and Rationale

출입단말기의 가용여부를 판단하기 위하여, ping/echo방식과 heartbeat방식 두가지 모두 채택하였으며, 출입단말기의 장애 발생시 아파트 관리자에게 통보하는 방식으로 설계하였다.





|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Quality Attribute | | Analysis | Ping/echo  or  heartbeat | 예외발생 |
| ID | Title |
|  | 가용성 | Pros/Cons | (+) 출입단말기의 가용여부 판단 및 아파트 관리자 통보를 통해 External Device임에도 가용성을 향상시킨다. | 출입단말기 내부의 예외처리 사항으로 구현 불가 |
|  | 변경용이성 | Pros/Cons | (+) 두가지 방식의 출입단말기를 모두 지원함으로써, 추가적인 출입단말기 대응이 용이해 질 수 있음 | 출입단말기마다 예외처리 방식이 달라져 새로운 출입단말기를 대응할 수 없음 |
|  | 성능 | Pros/Cons | (-) 주기적인 신호에 의해 장애를 탐지하는 설계로써 탐지시간과 부하는 서로 반비례 관계임 | (0) 예외발생 방식을 사용할 수 있다면 성능 측면에서는 전혀 부하가 없음 |

### DD-04 출입단말기 연동방식 (유지보수성)

#### Design Goal

본 시스템의 설계는 External Device인 출입단말기의 호환성을 고려해야 한다.

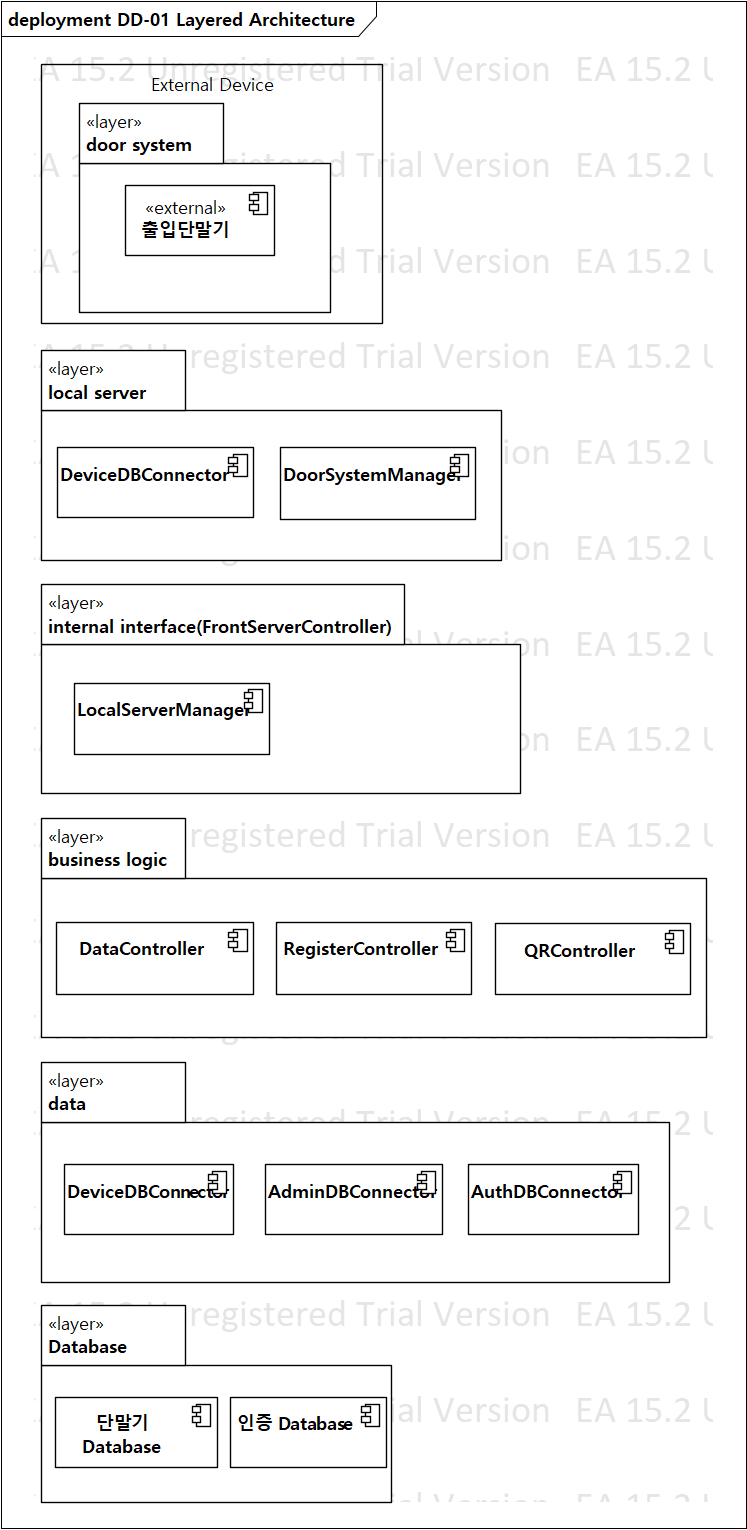
따라서 다양한 출입단말기에 대한 Interface를 지원해야 하며, DD-04에서도 같은 맥락으로 ping/echo, heartbeat방식의 가용성 판단기능을 모두 지원할 수 있도록 설계하였다.

전체적인 시스템의 설계에서는 Layered Architecture를 활용하여 출입단말기와 Interaction하는 부분을 최상위단 Layer로 분리하고, 출입단말기와 통신하는 컴포넌트를 분리하는 작업을 진행한다.

이와 같은 설계를 통해 추가적인 출입단말기의 기능에 대한 변경 컴포넌트를 최소화 하고, 변경점에 대하여 다른 컴포넌트에 영향성이 없도록 한다.

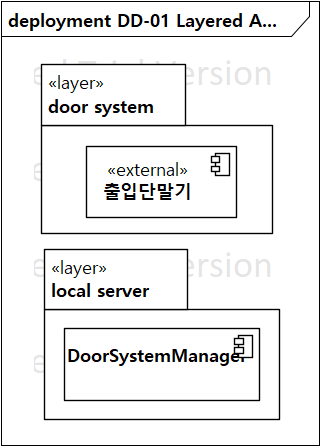
#### Design Approach List

#### Design Approach #1 Description: Layered Architecture



각각의 Layer에서는 상위레이어로의 의존성 관계가 존재하지 않도록 설계하여, 출입단말기 인터페이스의 변경은 DoorSystemManager에만 변경점이 발생하도록 구조화 하였다.

#### Design Approach #2 Description: 출입단말기 대응 컴포넌트 최소화



출입단말기와 통신하는 DoorSystemManager 컴포넌트는 내부 구성 클래스가 출입단말기와의 인터페이스로만 구성되어, 수정이 용이한 방향으로 설계되었다.

#### Decision and Rationale

다양한 출입단말기를 지원하기 위하여 위에서 언급된 Layered Architecture와 컴포넌트 최소화 전략을 모두 채택하였다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Quality Attribute | | Analysis | Layered Architecture | 출입단말기 대응 컴포넌트 최소화 |
| ID | Title |
|  | 변경용이성 | Pros/Cons | (+)각 컴포넌트들의 의존성이 분리되어 수정이 용이한 구조를 가진다. | (+)추가적인 출입단말기 인터페이스를 지원하기 위하여 수정이 발생하면, 최소한의 영향성을 가진다. |

# Component Level Design Description

## *Component 1* Design Description

### Overview

### Static Structure Diagram

### Element List

|  |  |
| --- | --- |
| Name | Responsibility |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

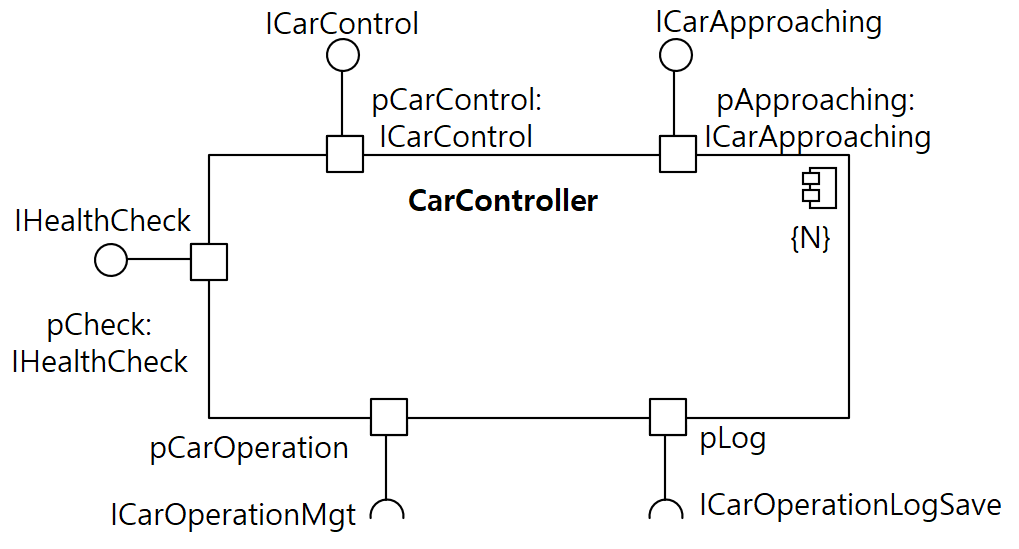
### Design Rationale

|  |  |  |
| --- | --- | --- |
| QA | Relevant Elements | Description |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## CarController Design Description

### Overview

<작성 방법>



**각 인터페이스를 Class diagram으로 표시**

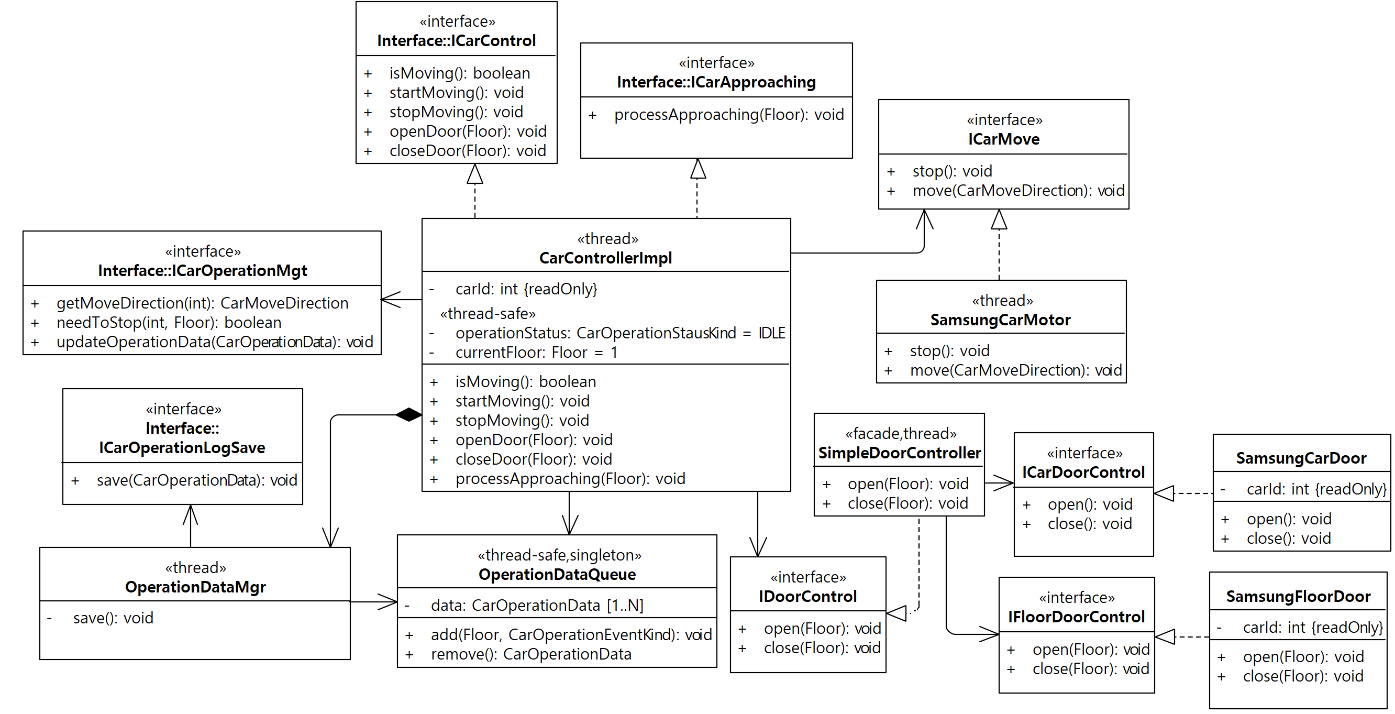
|  |  |
| --- | --- |
| 항목 | 설명 |
| 개요 | 이 컴포넌트의 역할을 한/두 줄로 간략하면서 명확하게 서술한다. |
| 컴포넌트 기능 요구사항 | 이 컴포넌트에 부여된 기능 요구사항을 서술한다. Provided Interface를 중심으로 구체적으로 설명한다. |
| 컴포넌트 품질 요구사항 | 이 컴포넌트가 만족해야 할 품질 요구사항을 서술한다. 시스템의 QA를 충족시키기 위하여 이 컴포넌트가 가져야할 QA(성능, 유지보수성)을 구체적으로 서술한다. |

### Static Structure Diagram

<작성 방법>

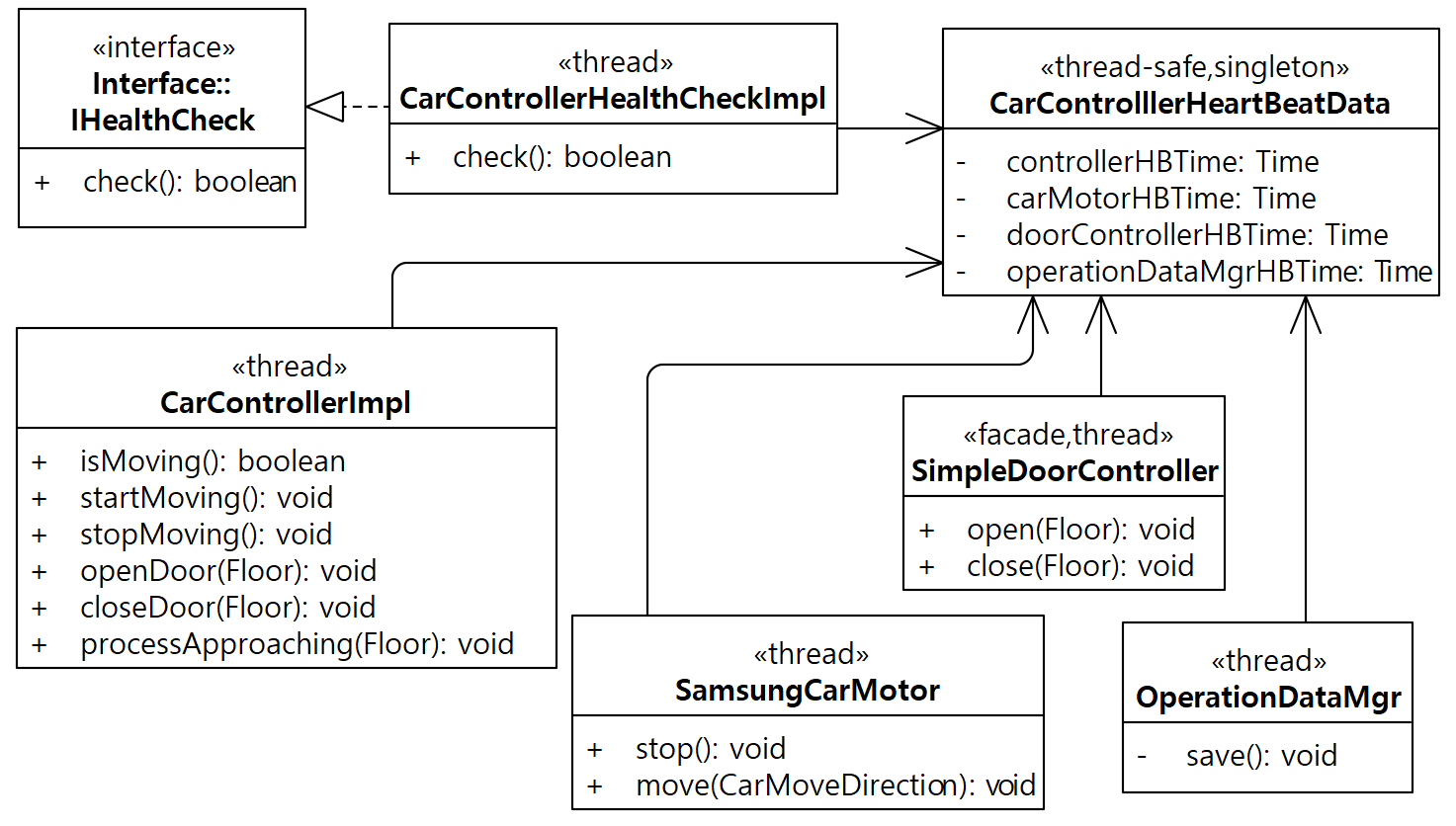
* 앞에서 명세한 기능 및 품질이 Static Structure Diagram을 통해서 구체화되었는가?
* Component를 구현하기 위하여 필요한 Class 및 Interface들이 모두 식별되었는가? 즉 Component에 주어진 기능 및 품질 요구사항을 충족시키기 위하여 필요한 설계가 수행되었는가?
  + 성능을 고려하면 multi-threading, thread-safe queue 등이 필요함
  + 유지보수성을 고려하면 응집도, SOLID 등을 적용해서 세분화 필요함
* Class 간의 관계(generalization, association, composition/aggregation, dependency)가 올바르고 타당한가?
* Class의 attribute 및 operation이 구체적으로 명시되었는가?
* SOLID 등의 설계 원칙을 준수하였는가?
* GoF 등의 Design Pattern이 올바르게 적용되었는가?
* 많은 수의 Class 및 Interface가 식별된 경우 Package Cohesion/Coupling을 고려하여 Packaging한다.

<ICarConrol, ICarApproaching 설계>



CarControllerImpl 클래스는 thread로서 ICarControl과 ICarApproaching을 구현한다. ICarControl 기능은 ……. 그리고 ICarApproaching 기능은…….

<IHealthCheck 설계>



설명

### Element List

<작성 방법>

* 위의 Class diagram에 표현된 모든 요소(package, class, interface)에 대한 역할을 설명한다.
* 각 요소가 Component에 부여된 요구사항(기능, 품질, 제약사항 등)에 어떻게 기여하는 지 측면에서 설명한다.

|  |  |
| --- | --- |
| Name | Responsibility |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

### Design Rationale

<작성 방법>

* Component 내부 설계를 통해서 구현/달성되는 QA에 대하여 그 설계 결과를 정당화한다.
* 이 Component와 관련된 QA에 대하여 이 QA 달성을 위하여 직접적으로 기여하는 design elements(class 및 interface)를 나열한다.
* 이들 관련된 design elements들이 QA 달성 측면에서 최적의 설계 인지에 대하여 정당화한다. 적용된 Pattern 및 Tactic이 있다면 해당 Pattern/Tactic을 구체적으로 제시한다. 그리고 고려될 수 있는 다른 설계 후보가 있었다면 그 설계 후보와 비교하여 정당화한다.

|  |  |  |
| --- | --- | --- |
| QA | Relevant Elements | Description |
| QA-01 |  |  |
| QA-03 | - | - |
| … |  |  |

## *Component n* Design Description

# Architecture Evaluation

## Traceability Summary

<작성 방법>

점검 기준>

* 모든 기능적 요구사항 즉 Use Case가 설계에 반영되었는가?
* 모든 품질 요구사항 즉 QA Scenario가 설계에 반영되었는가?
* 모든 제약 사항 특히 Technical Constraint가 설계에 반영되었는가?
* 각 AD와 관련된 Design Element를 모두 명시하였는가?
* Design Element를 명시할 때 해당되는 View와 함께 구체적으로 명시하였는가?
* AD의 특성에 따라서 타당한 View/Element가 명시되었는가?
* 이러한 추적 관계를 실제 해당 View에서 확인할 수 있는가?

|  |  |  |
| --- | --- | --- |
| Architectural Driver | | Design Elements |
| ID | Title |
| UC-01 |  | C1: …  C2: .. |
| UC-02 |  |  |
| … |  |  |
| QA-01 |  | Node1, Node2: load balancing  C1, C2: ….. |
| QA-02 |  |  |
| … |  |  |
| BC-01 |  |  |
| BC-02 |  |  |
| … |  |  |
| TC-01 |  |  |
| TC-02 |  |  |
| … |  |  |

## Use Case Traceability

### UC-01 Title

<작성 방법>

* 해당 기능(Use case)에 대한 설계 결과가 구체적으로 기술되었는가? 즉 기능에 대한 설계 요소(Component, class 등) 및 설계 방법 등이 구체적인가?
* 기술된 design decision을 바탕으로 해당 기능에 대한 설계의 적절성을 판단할 수 있는가?

|  |  |
| --- | --- |
| Design Element | Design Decision Description |
|  |  |
|  |  |

### UC-02 Title

## Quality Attribute Traceability

### QA-01 Title

<작성 방법>

* 해당 QA Scenario에 대한 설계 결과가 구체적으로 기술되었는가? 즉 QA Scenario에 대한 설계 요소(Component, class 등) 및 설계 방법 등이 구체적인가?
* 기술된 design decision을 바탕으로 해당 QA에 대한 설계의 적절성을 판단할 수 있는가?

|  |  |
| --- | --- |
| Design Element | Design Decision Description |
|  |  |
|  |  |

### QA-02 Title

## Constraint Traceability

### BC-01 Title

<작성 방법>

* 해당 constraint 대한 설계 결과가 구체적으로 기술되었는가? 즉 constraint에 대한 설계 요소(Node, Execution Environment, Communication Path, Component, Class 등) 및 설계 방법 등이 구체적인가?
* 기술된 design decision을 바탕으로 해당 Constraint가 설계에 적절하게 반영되었는 지를 판단할 수 있는가?

|  |  |
| --- | --- |
| Design Element | Design Decision Description |
|  |  |
|  |  |

### BC-02 Title

### TC-01 Title

### TC-02 Title