# interim feedback

| 장 | 절 | 배점 | 점수 | 평가 의견 |
| --- | --- | --- | --- | --- |
| Project Overview  (30) | Project Background | 5 | 5 | business context의 central entity에는 시스템을 제시하지 않습니다.  business 단계에서는 명시적으로 시스템이 언급되지 않습니다.  시스템 관점이 아니라 비즈니스 관점/수준에서의 각 이해관계자간의 정보/이해관계를 기술합니다.  각 이해관계자 입장에서의 시스템 수준이 아닌, 궁극적인 goal에 대하여 타당한 목표 수치를 정량적으로 제시 필요합니다. |
| Business Context Diagram | 5 | 3 |
| Stakeholder List | 5 | 5 |
| Business Goals | 15 | 10 |
| System Overview  (90) | System Context Diagram | 20 | 13 | 10만, 1만 등의 산출 근거 설명 필요  방문자와 직접 interaction하는지?  방문QR은 단말기를 통해서 시스템에 입수됨  인증서버? 이 기능도 시스템 범위에 포함됨  External entiity는 사용자, 장치, 시스템 유형으로서 업체는 부적절함  전송, 알림, 입력 등의 동작/행위가 아니라 전송되는 데이터 자체만을 표현해야 함  방문자 출입 시 거주자에 이 출입 사실이 통보되어야 함  Feature는 시스템의 단순 기능이 아니라 Business goal 달성에 기여하는 Stakeholder 수준에서 이해/평가될 수 있어야 함; 현재 제시된 Feature는 일부는 제공할 기능으로서 Stakeholder의 BG를 달성하는 데 직접적으로 기여한다고 판단되기 보다는 시스템이 구체적으로 제공할 기능으로 보입니다.  시스템을 주체로 하여 시스템의 특징/강점이 Stakeholder 관점에서 이해되도록 기술되어야 합니다. 즉, Title/Description이 이해관계자에게 설명/설득시킬 수 있는 문장이 되어야 합니다. 평이한/단순한 기능/특징 설명은 부족합니다. |
| External Entity List | 20 | 15 |
| External Interface List | 20 | 15 |
| System Feature List | 30 | 22 |
| Architectural Driver  (90) | Use Case Model | 40 | 27 | UC는 사용자 관점/수준에서의 독립적인 기능 단위입니다. 따라서 얼굴/QR 인식에 따른 처리(통보 포함)은 하나의 UC가 됨  UC diagram 전체적으로 보완 필요  UC 시나리오는 시스템 또는 Actor를 주어로 하는 능동태 문장이 되어야 합니다.  시나리오는 실제 Actor와 시스템간의 입/출력을 명확하게 표현해야 합니다.  예) 발급받은 QR코드는 방문자 정보에 입력된 연락처를 통해 공유된다. ==> 누가 하는지? 방문자에게 공유되는 방법은?  단말기 모니터링, 시스템 모니터링 복구 등의 기능은?  QA도 사용자 관점에서 정의 필요. 예) QA-01, 03 병합 필요  QA-02 얼굴인식까지의 성능이 아니라 출입 인가 판단까지 또는 출입단말기까지의 전송까지의 시간이 중요함 |
| Quality Attribute Scenario | 40 | 27 |
| Constraint | 10 | 10 |
|  | | 210 | 152 | 72.4% |

# prefinal feedback

| 장 | 절 | 배점 | 점수 | 평가 의견 |
| --- | --- | --- | --- | --- |
| Architectural Driver  (50) | Use Case Model | 20 | 15 |  |
| Quality Attribute Scenario | 25 | 20 |
| Constraint | 5 | 5 |
| Top Level Design Description  (160) | System Infrastructure View | 20 | 15 | comm. path multiplicity 제시 필요  스마트폰도 개발 범위에 포함되므로 <<external>> 부적절  인증 DB로의 L7 LB의 역할은?  각 노드에 deploy되는 컴포넌트들이 명확히 구분되도록 컴포넌트 다이어그램 작성이 필요합니다. 예) 노드 별로 컴포넌트 다이어그램 작성  각 노드 내부의 컴포넌트들에 대해서 layer 구조가 필요합니다.  패턴을 활용한 적절한 이름이 권장됩니다. 예) gateway, frontcontroller, DAC, ...  컴포넌트의 역할에 맞는 동작 및 이름 필요. 예) QRController가 SendQRData를 하는지..QR생성이 바람직  Seq diagram에서 각 컴포넌트가 deploy되는 노드를 swimlane으로 표현하면 좋겠습니다.  UC-01 거주자인지, 미등록자인지는 일단 인식 후에 결정되는 것임. 따라서 2개가 아니라 1개로 diagram 작성 필요  1.3 얼굴인식 뿐만 아니라, 해당 출입문에 대한 허가된 사용자 인도도 판단해야 함. 이러한 동작이 명확해야 함. IsRegisteredData는 모호함. DataController도 모호함  독립적인 변경/배포의 가능성이 있는 컴포넌트들은 개별적인 <<artifact>>가 권장됩니다.  즉, 개별적인 변경이 있을 수 있는 여러 개의 컴포넌트들을 하나의 <<artifact>>로 하는 것은 바람직하지 않습니다.  MariaDB 상에서 JAR가 실행가능한지요?  각 Design approach를 텍스트로 설명하는 것뿐만 아니라, 해당 approach에 따른 설계 결과를 적절한 그림/다이어그램으로 표현하는 것이 필요합니다.  Decision and Rationale에서는 목표로한 QA 뿐만 아니라 이 Design approach로 인해서 영향을 받을 수 있는 모든 QA에 대한 pros/cons 제시가 필요합니다.  알고리즘 모듈 크기 감소 및 인터페이스화 ==> 정확도와 무슨 관계가 있는지?  상호배타적인 design approach를 제시하고 이에 대한 장단점 분석을 통해서 최종적으로 선택하는 것입니다.  외부 시스템/장치에 대한 heartbeat vs ping/echo는 우리가 결정할 수 있는 것이 아니라 해당 장치와의 통신 방법에 의해서 결정됩니다. 즉 Design decision이라고 볼 수가 없습니다. |
| Structure View | 40 | 26 |
| Behavior View | 40 | 27 |
| Deployment View | 20 | 15 |
| Documenting Design Decisions | 40 | 27 |
|  | | 210 | 150 | 71.4% |

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.

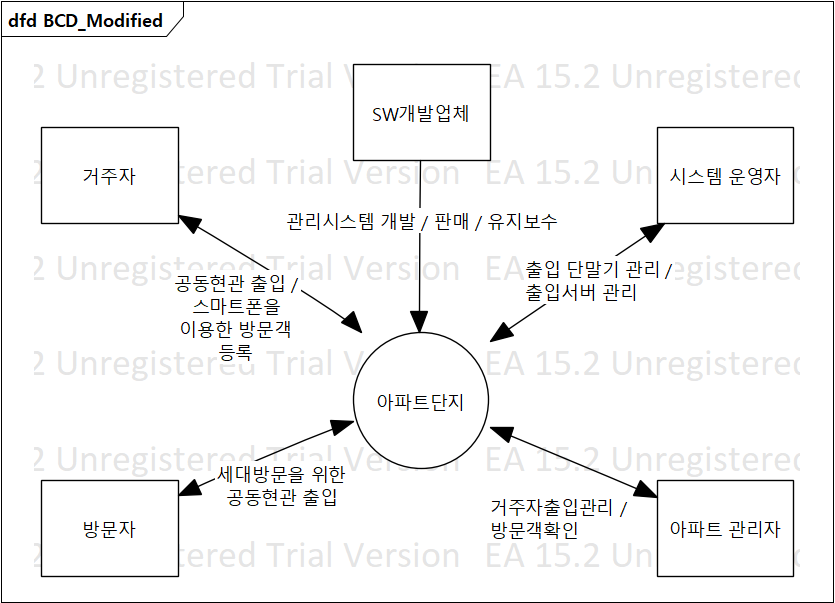
기존의 출입관리 시스템은 반드시 최초 아파트 신축 당시 함께 도입해야 하지만, 본 시스템은 신축 또는 구축 등 어떠한 아파트 환견에서도 도입이 가능하다.



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.
* 아파트 관리자를 통해 거주자의 등록, 수정, 삭제가 가능하다.
* 용인시 전체 아파트에 시스템이 도입된다.

## Business Context Diagram



## Stakeholder List

| Stakeholder | Description |
| --- | --- |
| 거주자 | 배경 : 아파트에 거주하고 있는 주민  관심사 :   * 비접촉식 인증방식을 통한 공동현관 출입을 원하고 있음 * 강력한 보안 시스템을 통해 출입이 관리되기를 원함 * 스마트폰을 이용해 해당 세대에 방문자를 등록하기를 원함 |
| 아파트 관리자 | 배경 : 거주자 정보를 등록할 수 있는 아파트 관리자   * 아파트 단지별로 1명씩 존재함 * 각 세대별로 얼굴인식에 사용될 데이터를 수집하고, 아파트 관리 콘솔을 통해 거주자 데이터를 입력할 수 있음 * 각 아파트마다 다양한 출입단말기를 가지고 있기 때문에, 해당 아파트에 시스템이 잘 설치되기를 원함 |
| 시스템 운영자 | 배경 : 전체 시스템 관리 및 아파트 관리자 관리   * 전체 시스템관리자 1명 * 시스템 관리 콘솔을 이용하여 아파트 정보를 등록 |
| 방문자 | 배경 : 택배, 배달등을 위해 세대현관까지 도달해야 하는 방문자   * 방문하고자 하는 세대가 부재중일때를 대비하여 세대에서 거주자가 방문 등록 후 발급받은 임시출입증을 통해 공동현관을 출입할 수 있어야 함 |
| SW개발업체 | 배경 : 시스템의 개발 담당업체   * 새로운 시스템을 개발하여 용인시 전체의 아파트에 제공하고자 함 * 시스템 장애 발생을 최소화할 것과 문제 발생시 즉각적인 기술지원을 보증함 |

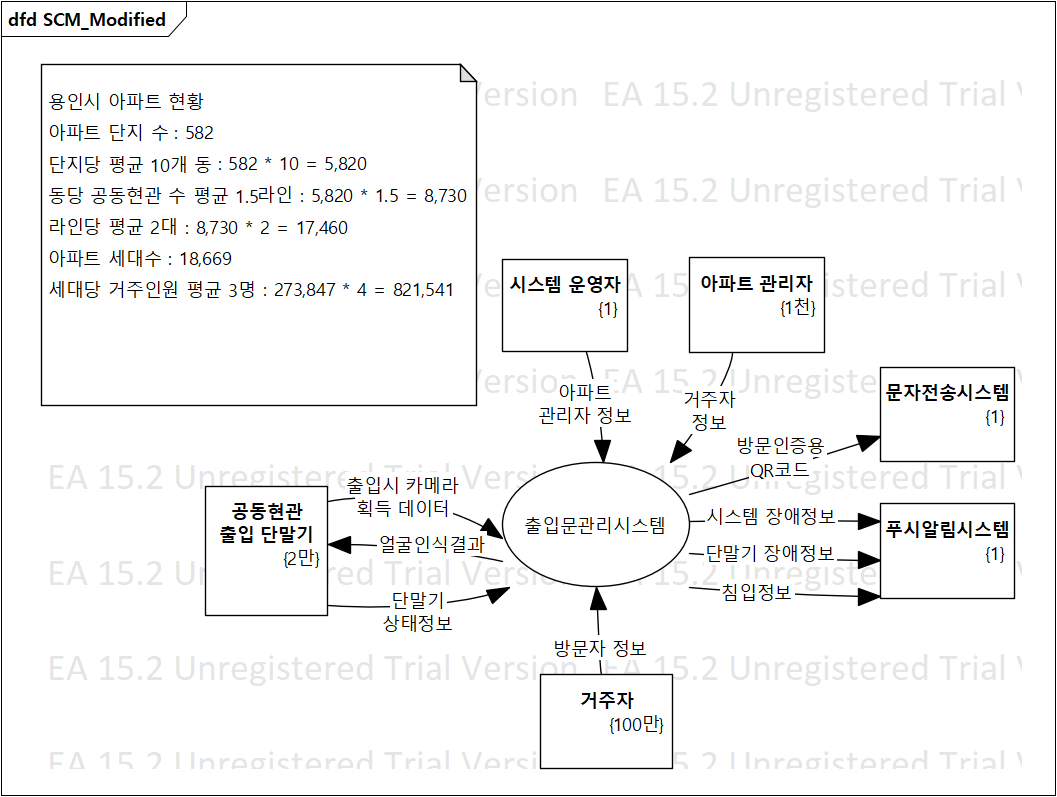
## Business Goal List

| Stakeholder | Business Goal | | |
| --- | --- | --- | --- |
| ID | Statement | I |
| 거주자 | BG-01 | 거주자는 얼굴인식을 통해 공동현관이 개방될 수 있기를 기대함  스마트폰을 통해 방문자를 등록할 수 있어야 함   * 기존 비밀번호 방식(약 5초)보다 빠르게 공동현관문이 개방 되었으면 좋겠음 * 스마트폰을 통해 세대에 방문하고자 하는 방문자를 등록하고, 등록과 함께 방문자에게 임시출입이 가능한 출입증이 1분 이내에 전송되었으면 좋겠음 | 상 |
| 아파트 관리자 | BG-02 | 아파트 관리자는 거주자 정보를 등록하여야 하며, 시스템 도입시 아파트 관리콘솔에서 기존과 동일하게 운영이 가능하기를 기대함   * 거주자 등록 등 아파트 관리콘솔에 별도 프로그램 설치 없이 관리가 가능하기를 원함 * 출입단말기 이상 발생시 5분이내에 알림을 받을 수 있기를 원함 * 공동현관에서 침입시도가 검출되는 경우 5초이내에 알림을 받기를 원함 | 상 |
| 시스템 운영자 | BG-03 | 시스템 운영자는 용인시 전체 아파트에 도입된 시스템을 관리하여야 함.   * 시스템 장애 발생시 1분이내에 알림을 받을 수 있기를 원함 * 시스템 장애 발생시 즉시 복구될 수 있는 조치가 필요함 * 공동현관에서 침입시도가 검출되는 경우 5초이내에 알림을 받기를 원함 | 상 |
| 방문자 | BG-04 | 방문자는 세대호출 없이도 공동현관 출입이 가능하기를 원함   * 미리 발급받은 임시출입증 통해 방문세대가 부재인 경우에도 공동현관 출입이 가능하기를 기대 | 중 |
| SW개발업체 | BG-05 | 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 * 라인당 평균 2대 : 8,730 \* 2 = 17,460 * 아파트 세대수 : 273,847 * 세대당 거주인원 평균 3명 : 273,847 \* 4 = 821,541   \*. 출처 : http://www.yongin.go.kr/user/web/coprnhousestus/BD\_selectCoprnHouseStus.do | 하 |

\* I : Importance

# System Overview

## System Context Diagram



## External Entity List

| Name | Description |
| --- | --- |
| 거주자 | 유형 : 사용자  핵심 기대 사항 : 거주자는 스마트 폰 앱을 통해 방문자를 등록할 수 있으며, 등록시 방문자에게 QR코드가 문자메시지로 전송된다.  (방문자 등록 정보 : 방문자의 이름, 전화번호, 방문일시, 방문목적 정보)  숙련도 : 스마트 폰 앱을 통해 로그인, 메뉴선택, 사용자 등록 등 일반적인 사용이 가능 |
| 아파트 관리자 | 유형 : 사용자  핵심 기대 사항 : 아파트 관리자는 거주자가 얼굴인식을 통해 출입이 가능하도록 얼굴정보를 등록할 수 있어야 하며, 침입시도가 검출되면 알림을 받는다.  출입단말기의 장애 발생시 알림을 받을 수 있다.   * 얼굴정보 등록 : 아파트 관리 PC의 Web Browser를 통해 거주자의 사진 및 동/호 정보를 등록한다. * 출입단말기의 장애 알림 : Web Browser에서 푸시알림을 통해 장애발생여부를 확인한다.   숙련도 : PC의 웹페이지에 접속할 수 있고, 원하는 동작을 메뉴를 통해 진행이 가능하다. |
| 시스템 운영자 | 유형 : 사용자  핵심 기대 사항 : 시스템 운영자는 출입관리 시스템에 아파트 정보를 등록하여 시스템의 기능을 사용할 수 있도록 한다. 침입시도가 검출되면 알림을 받는다.  시스템의 장애 발생시 알림을 받을 수 있다.   * 아파트 정보 등록 : 시스템 관리 PC의 Web Browser를 통해 아파트 관리자 정보, 출입단말기 정보를 등록한다. * 시스템의 장애 알림 : 출입관리시스템의 장애 발생시 Web Browser에서 푸시알림을 통해 알림을 받는다.   숙련도 : PC의 웹페이지에 접속할 수 있고, 원하는 동작을 메뉴를 통해 진행이 가능하다. |
| 문자전송시스템 | 유형 : 시스템  역할 : 문자전송시스템은 거주자가 스마트폰을 통해 등록한 출입용 QR코드를 방문자에게 전달하기 위해 사용하는 외부 시스템이다.  시스템의 사양 :   * QR Code 이미지가 포함된 MMS 발송 지원 * MMS 발송을 위한 REST API 지원 |
| 푸시알림시스템 | 유형 : 시스템  핵심 기대 사항 : 푸시알림시스템은 시스템에서 제공하는 Web을 이용하는 Application에 알림을 전송하기 위해 사용하는 외부 시스템이다.  시스템의 사양 :   * Web Browser에서 해당 푸시알림서비스 수신 가능 * 푸시 알림 발송을 위한 REST API 지원 |
| 공동현관 출입 단말기 | 유형 : 장치  역할 : 카메라를 통해 획득된 영상을 시스템으로 전송하는 장치  장치의 사양 : 기존에 설치된 카메라가 존재하는 모든 공동현관 단말기   * 해상도 : SD(640x480), HD(1280x720), FHD(1920x1080) * Frame Rate : 30fps * Network : Ethernet or Wifi * 출입문 제어기능 탑재 |

## External Interface List

| Name | Description |
| --- | --- |
| 출입시 카메라 획득 데이터 | 역할 : 출입을 위해 출입단말기 카메라로 입력되는 이미지 정보 전달  유형 :   * User Interface : 출입단말기 카메라 * System Interface : TCP   특성 :   * 크기 : SD(640 x 480), HD(1280 x 720), FHD(1920 x 1080) 지원 * Frame Rate : 30 Frame/sec * Format : PNG or Jpeg * 빈도 : 동시 최대 2만건(출입단말기 최대 2만대) |
| 얼굴인식결과 | 역할 : 출입관리시스템이 출입단말기로부터 영상을 받아 얼굴을 인식한 뒤 해당 얼굴정보를 통해 출입가능여부를 회신한다.  유형 :   * System Interface : HTTPS   특성 :   * 크기 : 1KB * Format : JSON * 빈도 : 동시 최대 2만건(출입단말기 최대 2만대) |
| 단말기 상태정보 | 역할 : 출입단말기의 가용여부를 확인하기 위하여 주기적으로 신호를 전송한다.  유형 :   * System Interface : HTTPS Protocol   특성 :   * 크기 : 1KB * Format : JSON * 빈도 : 단말기별 분당 1건, 시스템 기준 분당 2만건 |
| 아파트 관리자 정보 | 역할 : 아파트 관리자 등록을 위해 시스템 운영자가 시스템에게 출입단말기 정보를 전달한다.  유형 :   * System Interface : HTTPS Protocol   특성 :   * 아파트 이름, 주소, 단말기 정보 * 크기 : 1KB * Format : JSON * 빈도 : 하루 최대 1천건 (아파트관리자 최대 1천명) * 1회성 등록으로 실제 빈도 낮음 |
| 방문자 정보 | 역할 : 거주자가 스마트폰 앱을 통해 방문자 정보를 등록한다.  유형 :   * User Interface : Mobile Application * System Interface : HTTPS Protocol   특성 :   * 방문시간, 방문목적, 방문자 전화번호 * 크기 : 10KB * Format : JSON * 빈도 : 하루 최대 2만건 (평균 1가구당 1회 방문시) |
| 거주자 정보 | 역할 : 거주자 등록을 위해 아파트 관리자가 거주자 정보를 입력하면 출입관리시스템에 등록된다.  유형 :   * User Interface : Web Application * System Interface : HTTPS Protocol   특성 :   * 거주자 동/호수, 거주자 얼굴사진 * 크기 : 최대 10MB(FHD 1920x1080x32bit 사진 + JSON 정보) * 빈도 : 최대 100만건 (거주자 100만명) |
| 방문 인증용 QR코드 | 역할 : 거주자가 스마트폰 앱을 사용하여 방문자를 등록하는 경우 출입관리시스템에서 방문 인증용 QR코드가 발급되어 문자전송시스템으로 전달한다.  유형 :   * System Interface : REST API   특성 :   * QR코드 이미지 (BMP) * 크기 : 최대 30KB (JSON String 1KB이내) * 빈도 : 하루 최대 2만건 (평균 1가구당 1회 방문시) |
| 시스템 장애정보 | 역할 : 출입관리시스템에 장애가 탐지되면 시스템관리자에게 알림을 전송하기 위하여 푸시알림시스템으로 해당 내용을 전달한다.  유형 :   * System Interface : REST API   특성 :   * 크기 : 1KB * Format : JSON * 빈도 : 매우적음 |
| 단말기 장애정보 | 역할 : 출입단말기의 장애가 탐지되면 아파트관리자에게 알림을 전송하기 위하여 푸시알림시스템으로 해당 내용을 전달한다.  유형 :   * System Interface : REST API   특성 :   * 크기 : 1KB * Format : JSON * 빈도 : 매우적음 |
| 침입정보 | 역할 : 출입관리시스템에서 얼굴인식을 진행한 뒤 미등록 사용자임이 확인되면 아파트관리자에게 알림을 전송하기 위하여 푸시알림시스템으로 해당 내용을 전달한다.  유형 :   * System Interface : REST API   특성 :   * 크기 : 1KB * Format : JSON * 빈도 : 매우적음 |

## System Feature List

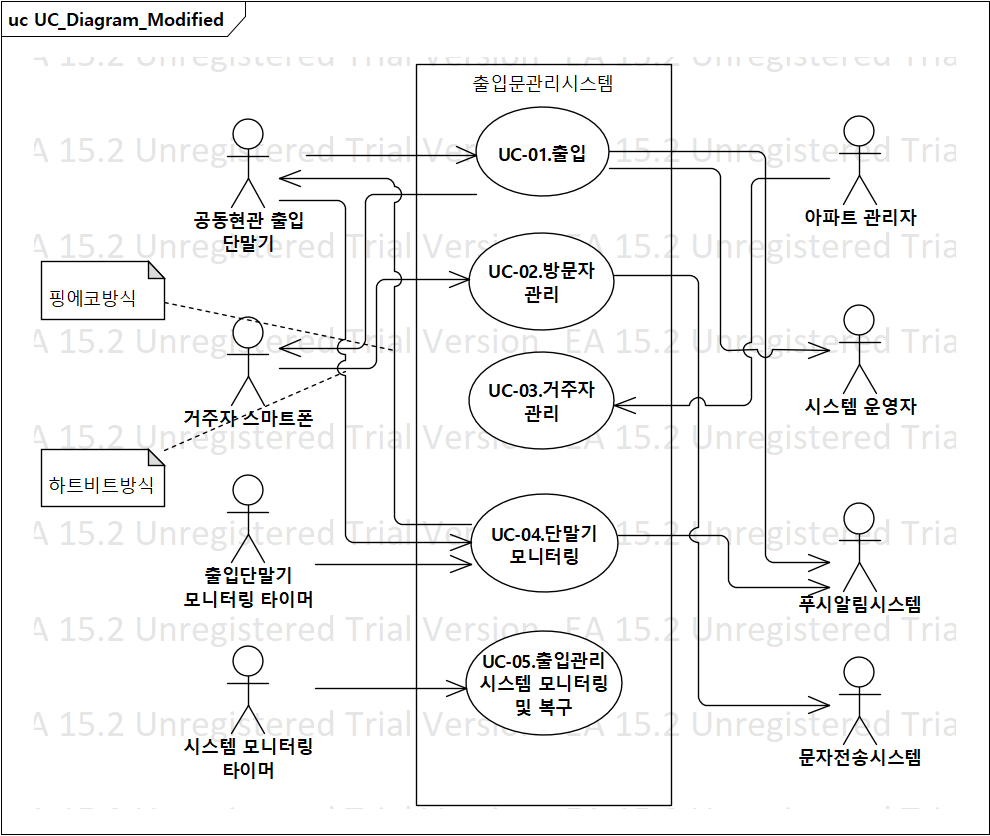
| ID | Title | Description | I | Related  Business Goal ID |
| --- | --- | --- | --- | --- |
| SF-01 | 편리한 공동 출입문 출입인증 서비스 제공 | 얼굴인식 및 QR코드를 통한 공동 출입문 출입을 지원함에 따라 비밀번호를 손으로 누르지 않는 비접촉 방식으로 출입이 가능하도록 기능을 제공하고자 한다. | 상 | BG-01  BG-04 |
| SF-02 | 간편한 방문자 신청 시스템 제공 | 거주자는 스마트폰 앱을 통해 자신이 원하는 시간에 방문자가 공동 출입문을 통과할 수 있도록 간편하게 신청할 수 있으며, 방문자는 문자메시지를 통해 QR코드를 전달받아 편리한 방문이 가능하도록 기능을 제공하고자 한다. | 중 | BG-01  BG-04 |
| SF-03 | 효율적이고 간편한 아파트 관리기능 | 거주자 관리, 단말기 이상여부확인, 비정상적인 출입 시도 발생시 알림 등 편리하고 효율적으로 아파트를 관리할 수 있는 기능을 제공하고자 한다. | 중 | BG-02 |
| SF-04 | 미리 등록되지 않은 출입에 대한 보안장치 제공 | 미리 등록되지 않은 출입을 판단하는 기능을 제공하며, 등록되지 않은 출입 발생시 시스템 관리자 및 아파트 관리자에게 빠르게 통보하여 조치를 취할 수 있는 기능을 제공하고자 한다. | 상 | BG-02  BG-03 |
| SF-05 | 기존 사용중이던 출입단말기의 안정성 향상 | 기존 사용중이던 출입단말기를 모니터링 하여 이상여부를 체크하는 기능을 제공하며, 장애발생시 아파트 관리자에게 빠르게 통보하여 조치가 가능하도록 기능을 제공하고자 한다. | 상 | BG-02 |
| SF-06 | 안정적인 시스템 환경 제공 | 시스템은 장애발생시 사용불가한 경우가 없도록 즉각 대체할 수 있는 방안을 마련한다. 또한 발생한 장애에 대해서는 자동으로 복구하는 기능을 제공하고자 한다. | 상 | BG-02  BG-03 |
| SF-07 | 간편한 신규 단말기 추가 | 시스템 운영자는 Web Application환경을 통해 간편하게 단말기를 신규 등록할 수 있으며, 새로운 종류의 단말기가 출시되더라도 빠르게 지원이 가능하도록 한다. | 하 | BG-03  BG-05 |

\* 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 등록권한 * 거주자에 대한 정보 : 동/호수, 사진 |
| 시스템 운영자 | 출입관리시스템을 운영하는 운영자. 아파트 단위로 단말기 정보를 등록할 수 있다.  시스템 관점에서의 Actor 역할:   * 단말기의 등록권한 * 단말기 정보 : 아파트, 단말기 모델명, Serial 정보 |
| 푸시알림시스템 |  |
| 문자전송시스템 | 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 푸시알림을 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, 푸시알림을 통해 방문자에게 QR코드를 전송한다. | 상 | 상 | SF-02 |
| 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-03 |
| UC-04 | 단말기 모니터링 | 출입문 관리 시스템은 모니터링 타이머를 통해 단말기의 상태를 모니터링 한다. 단말기 상태를 주기적으로 모니터링 한 뒤, 문제가 발생되는 경우 푸시알림을 통해 해당 내용을 아파트 관리자에게 통보한다. | 중 | 하 | SF-05 |
| 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-06 |

\* I: Importance, D: Difficulty

### UC-01 출입

#### Scenario List

| Scenario Title | Kind | Description |
| --- | --- | --- |
| 거주자 출입 | 기본 | 등록된 거주자가 공동현관 단말기에 얼굴을 인식한다. |
| 방문자 출입 | 기본 | QR코드를 발급받은 방문자가 공동현관 단말기에 QR코드를 인식한다. |
| 미등록자 출입 | 기본 | 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 | 출입 단말기는 카메라의 영상정보를 출입관리시스템으로 전송한다. |
| 2 | 출입관리시스템은 전달받은 영상정보에서 얼굴정보를 확인한다. |
| 3 | 출입관리시스템은 얼굴정보의 등록여부를 판단한다. |
| 4 | 출입관리시스템은 등록된 사용자임을 확인하고 출입단말기로 출입허가 명령을 보낸다. |

#### 방문자 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 | 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(기본)

#### 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 | 얼굴 인식 정확도 향상 | Accuracy | 상 | 상 | UC-01  UC-02  UC-03 | SF-01  SF-04 |
| QA-02 | 침입 시도 통보 시간 | Performance | 중 | 중 | UC-01  UC-02  UC-03 | SF-01  SF-03  SF-04 |
| QA-03 | 시스템 모니터링 및 복구 | Availability | 상 | 상 | UC-05 | SF-06 |
| QA-04 | 단말기 이상 모니터링 | Availability | 상 | 상 | UC-04 | SF-05 |
| QA-05 | 새로운 출입 단말기 지원 | Maintainability | 중 | 중 | UC-01  UC-04 | SF-05  SF-07 |

### QA-01 출입 인가 처리 정확성

| QA Type | Accuracy |
| --- | --- |
| Description | 시스템은 정상동작 상태에서 인식된 데이터와 등록된 데이터를 비교하여 동일한 얼굴인 경우 99%로 인증 허가를 발급한다. |
| Source of Stimulus | 출입단말기   * 출입단말기 최대 2만대 |
| Stimulus | 출입단말기로부터 시스템으로 얼굴데이터를 수신   * 등록가능한 얼굴정보 최대 100만개 * 영상유형 : 사진 * 영상 해상도 : 출입단말기 모델에 따라 SD, HD, FHD 등의 다양한해상도로 제공될 수 있음 * 영상품질 : 다양한 촬영환경에 따라 품질이 달라질 수 있음 |
| Artifact | 시스템 |
| Environment | 시스템에 등록된 거주자의 얼굴인식   * 얼굴인식이 원활하게 가능한 카메라 상태 * 출입문이 닫혀있는 상태 * 카메라 시야에 얼굴이 잘 들어오는 상태   시스템 정상동작 상태   * CPU 사용량 70%, Memory 사용량 70% 이하 * 최대 동시 인증요청 1000개 * 네트워크 속도 : 약 10Gbps |
| Response | 전달받은 얼굴 데이터의 출입가능 여부 |
| Response Measure | 등록된 얼굴에 대한 출입허가 정확도 99% 이상 |

### QA-02 침입 시도 통보 시간

| QA Type | Performance |
| --- | --- |
| Description | 시스템은 정상동작 상태에서 인식된 얼굴데이터가 미등록으로 판단되는 경우 침입으로 판단하여 아파트 관리자에게 통보한다. |
| Source of Stimulus | 출입단말기   * 출입단말기 최대 2만대 |
| Stimulus | 출입단말기로부터 시스템으로 얼굴데이터를 수신   * 등록가능한 얼굴정보 최대 100만개 * 영상유형 : 사진 * 영상 해상도 : 출입단말기 모델에 따라 SD, HD, FHD 등의 다양한해상도로 제공될 수 있음 * 영상품질 : 다양한 촬영환경에 따라 품질이 달라질 수 있음 |
| Artifact | 시스템 |
| Environment | 시스템에 등록된 거주자의 얼굴인식   * 얼굴인식이 원활하게 가능한 카메라 상태 * 출입문이 닫혀있는 상태 * 카메라 시야에 얼굴이 잘 들어오는 상태   시스템 정상동작 상태   * CPU 사용량 70%, Memory 사용량 70% 이하 * 최대 동시 인증요청 1000개 * 네트워크 속도 : 약 10Gbps   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 모니터링 및 복구

| 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/echo방식의 단말기는 [DD-04 단말기 장애 진단]에 따라 주기가 결정됨 * heartbeat방식의 단말기는 단말기 설정에 맞게 주기를 전달받음   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 | 장애발생시 30초내에 장애상황을 판단하여야 하며, 발생시점으로부터 1분내에 아파트 운영자에게 Push Message가 전달되어야 한다. |

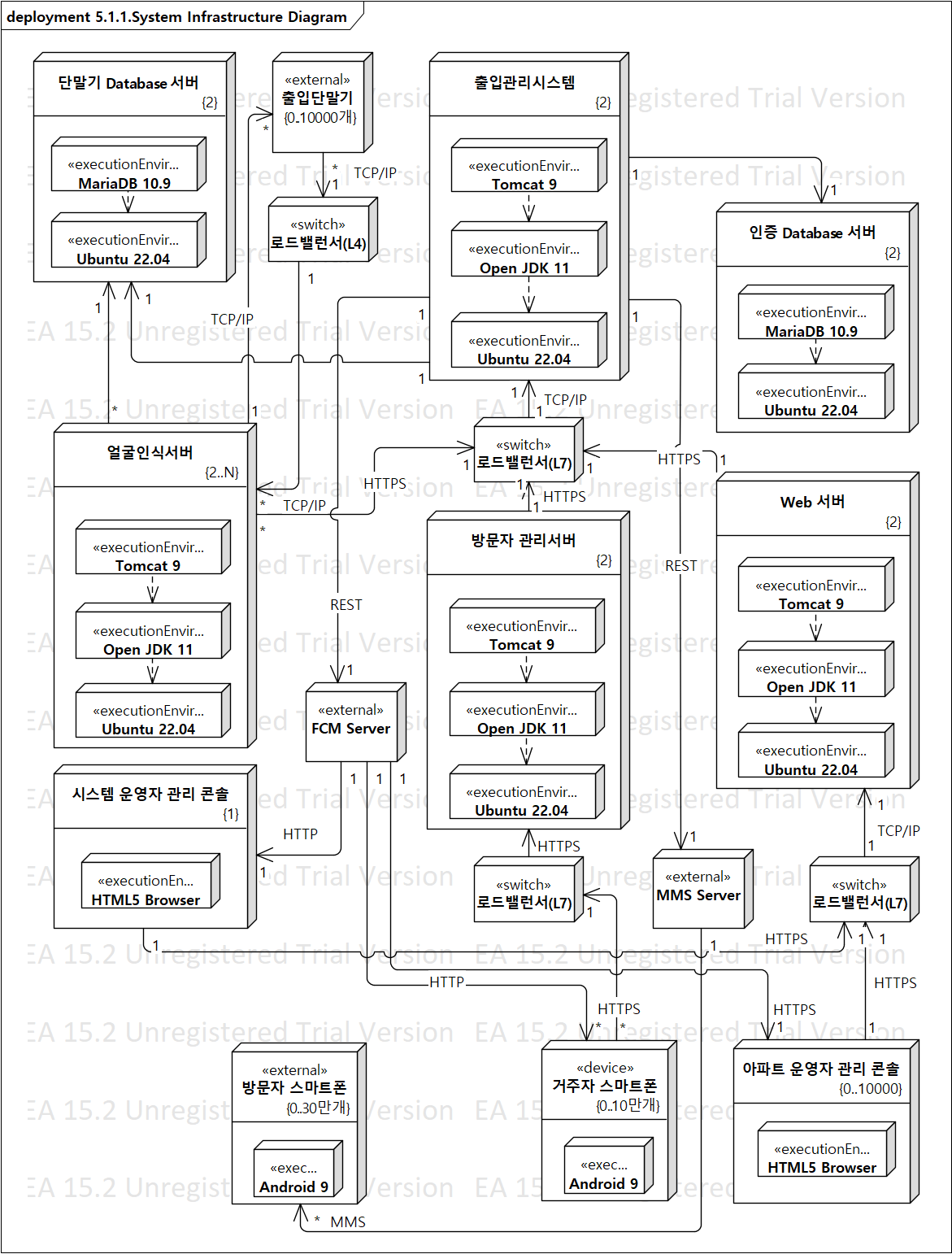
### QA-05 새로운 출입 단말기 지원

| 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 : 얼굴인식서버, 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) |
| 얼굴인식서버 | 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 얼굴인식서버, 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  얼굴인식서버  방문자관리서버  인증 Database 서버  단말 Database 서버  Web 서버 | Ubuntu 22.04 | 배포 후 장기간 지원을 위하여 현재 최신버전 사용 |
| 출입관리시스템  얼굴인식서버  방문자관리서버  Web 서버 | Open JDK 11 | Tomcat 9을 안정적으로 지원하기 위하여 사용 |
| 출입관리시스템  얼굴인식서버  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 |

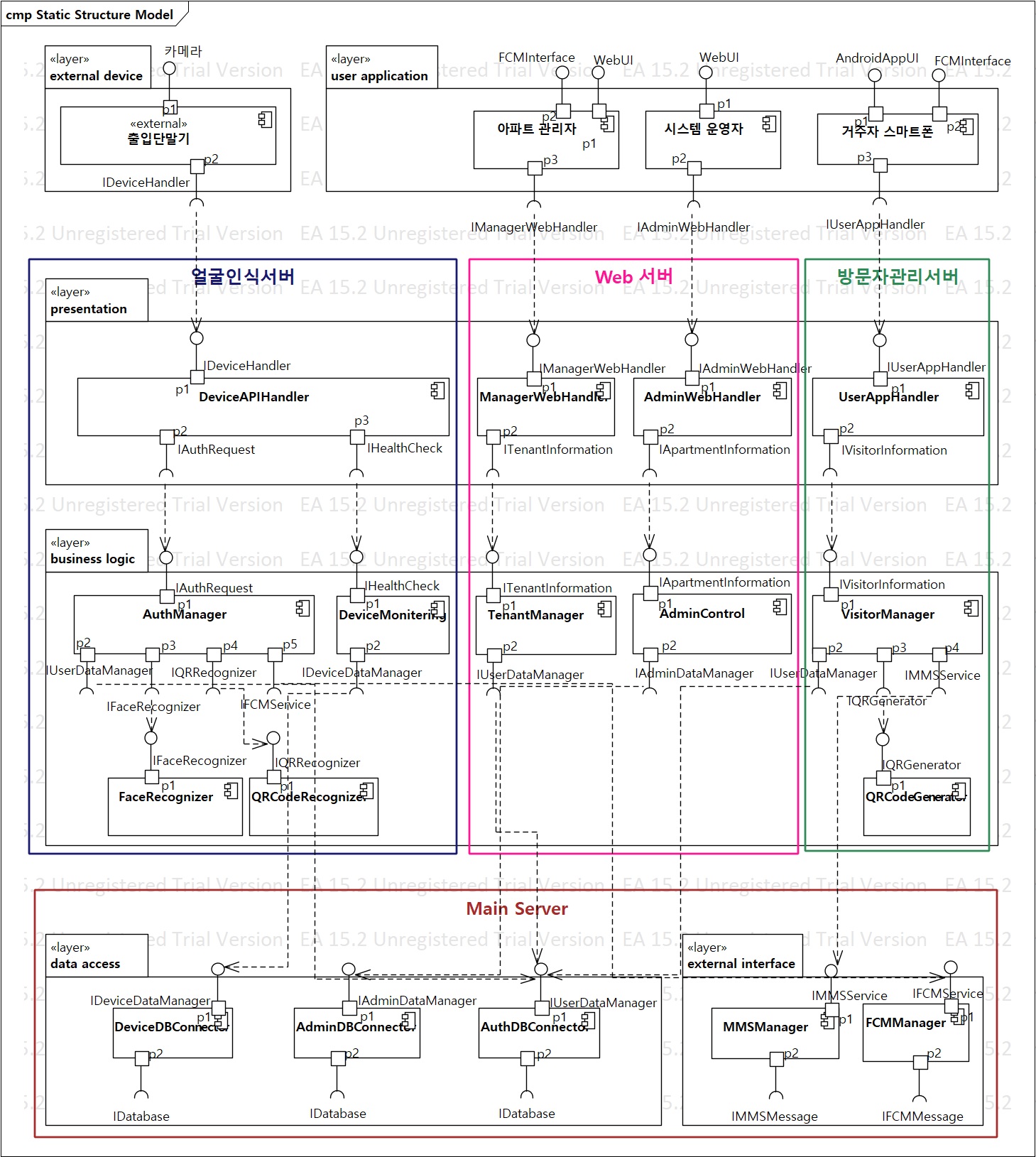
### Communication Path Specification

| Path | Description |
| --- | --- |
| 출입단말기 -> 얼굴인식서버 | TCP/IP   * 출입단말기가 지원하는 카메라 영상 전송 프로토콜 * 출입단말기에서 획득한 카메라 영상을 얼굴인식서버로 전송하기 위해 사용   SFTP   * 출입단말기가 지원이 가능한 경우 보안을 위해 SFTP 사용 |
| 각종서버 -> 로드밸런서  (얼굴인식서버, 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

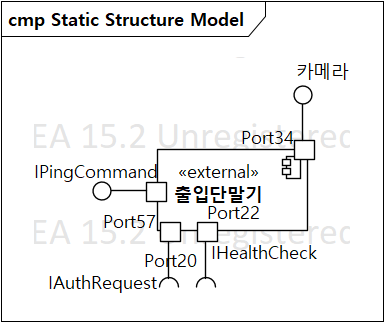


#### Element List

| Name | Responsibility | Relevant ADs |
| --- | --- | --- |
| *external device* | *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 |
| *user application* | *user의 입력단 application layer* |  |
| *아파트 관리자* | 거주자 등록을 위한 Web Page | QA-03 |
| *시스템 운영자* | 아파트 등록을 위한 Web Page | UC-01 |
| *거주자 스마트폰* | Android OS사용 스마트폰 Application | UC-02 |
| *presentation* | *출입단말기, 아파트관리자, 시스템운영자, 거주자 스마트폰으로 부터 들어오는 요청을 받는 역할* | *-* |
| DeviceAPIHandler | 출입단말기로부터 데이터를 전송받아 인증을 처리하거나 가용성체크를 담당하는 컴포넌트 |  |
| ManagerWebHandler | Web 서버로부터 아파트 거주자 등록정보를 입력받아 등록을 요청하는 컴포넌트 |  |
| AdminWebHandler | Web 서버로부터 아파트 등록정보를 입력받아 등록을 요청하는 컴포넌트 | UC-01 |
| UserAppHandler | 거주자 스마트폰으로부터 방문자 등록정보를 입력받아 등록을 요청하는 컴포넌트 | UC-02  QA-01 |
| *business logic* |  |  |
| AuthManager | 단말기로부터 영상데이터를 받아오고 얼굴 및 QR여부를 확인한다. 얼굴인식 및 QR Decoding을 완료한 후 Database에 등록여부를 확인하고, 등록된 사용자의 경우 단말기로 출입문 개방 메시지를 전달한다. 미등록 사용자의 경우 푸시메시지를 전송한다. | UC-01  QA-01 |
| FaceRecognizer | 얼굴인식 알고리즘을 담당하는 컴포넌트 |  |
| QRCodeRecognizer | QR코드 인식을 담당하는 컴포넌트 |  |
| DeviceMonitering | 단말기의 가용성을 체크하기 위한 컴포넌트  ping/echo 방식과 heartbeat 방식을 모두 지원한다. |  |
| TenantManager | 거주자 관리를 위한 컴포넌트  웹서버 내부에서 거주자 등록을 요청받아 Database에 등록한다. |  |
| AdminControl | 아파트 관리를 위한 컴포넌트  웹서버 내부에서 아파트 정보 등록을 요청받아 Database에 등록한다. |  |
| VisitorManager | 방문자 관리를 위한 컴포넌트  방문자관리서버 내부에서 방문자 등록을 요청받아 Database에 등록한다. |  |
| QRCodeGenerator | QR코드 생성을 담당하는 컴포넌트 |  |
| *data access* |  | *-* |
| DeviceDBConnect | 단말기 Database에 등록, 조회를 요청하는 컴포넌트 | UC-04  QA-04 |
| AdminDBConnect | 아파트 정보 Database에 등록을 요청하는 컴포넌트 | UC-01 |
| AuthDBConnect | 인증 Database에 등록, 조회를 요청하는 컴포넌트 | UC-01  UC-02  UC-03  QA-01  QA-02 |
| *external interface* | *시스템 외부로 interaction* | *-* |
| MMSManager | 문자메시지를 전송하기 위한 서버 인터페이스 | UC-02 |
| FCMManager | Push Message를 전송하기 위한 서버 인터페이스 | UC-01  UC-04  QA-02  QA-04 |

### 출입단말기 - User interface

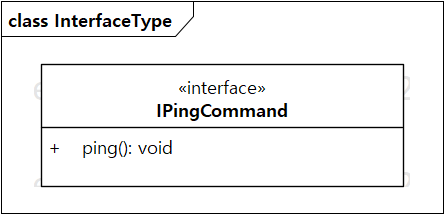
#### 출입단말기 Specification



#### Interface List

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

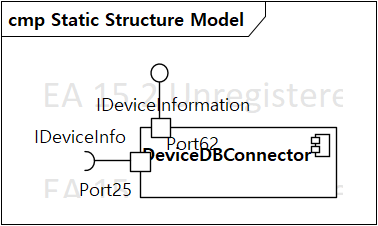
#### IPingCommand Interface Specification



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

### 얼굴인식서버 – Business Layer

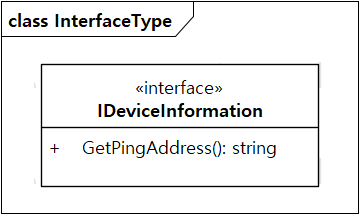
#### DeviceDBConnector Specification



#### Interface List

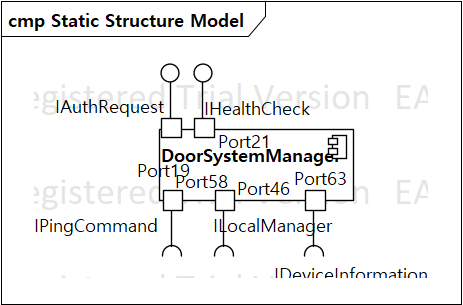
| Name | Kind | Description |
| --- | --- | --- |
| IDeviceInfomation | Provided | 출입단말기의 가용여부 판단을 위해 ping message를 전달할 주소를 제공하는 컴포넌트 |

#### IDeviceInfomation Interface Specification



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

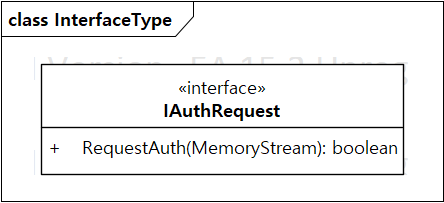
#### DoorSystemManager Specification



#### Interface List

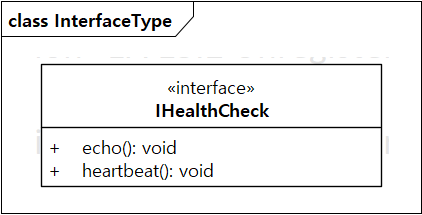
| Name | Kind | Description |
| --- | --- | --- |
| IAuthRequest | Provided | 출입단말기가 인증을 요청하기 위한 인터페이스 |
| IHealthCheck | Provided | 출입단말기로부터 heartbeat신호를 전달받기 위한 인터페이스 |

#### IAuthRequest Interface Specification



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

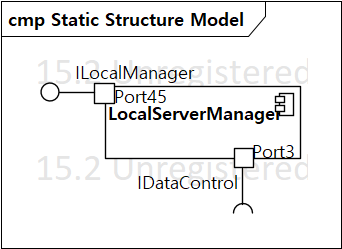
#### IHealthCheck Interface Specification



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

### 출입관리시스템 – internal interface(FrontServerController)

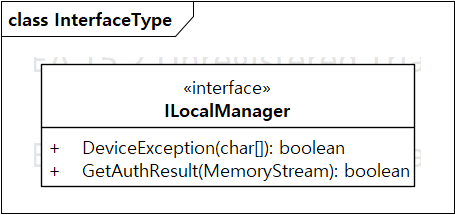
#### LocalServerManager Specification



#### Interface List

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

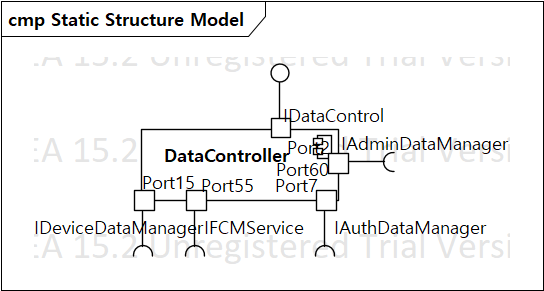
#### ILocalManager Interface Specification



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

### 출입관리시스템 – business logic

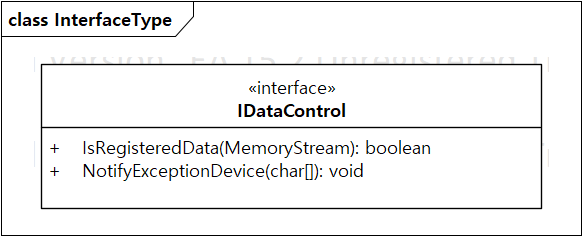
#### DataController Specification



#### Interface List

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

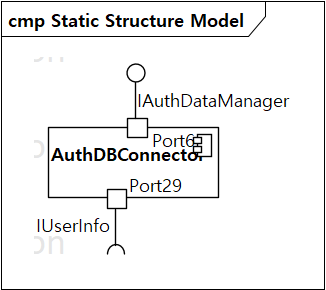
#### IDataControl Interface Specification



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

### 출입관리시스템 – data

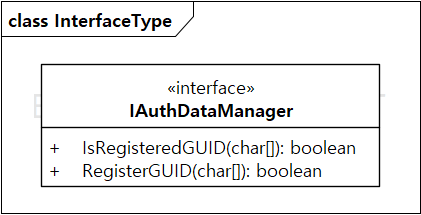
#### AuthDBConnector Specification



#### Interface List

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

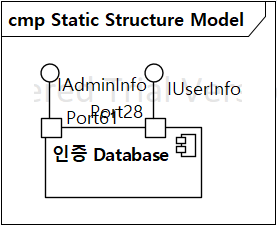
#### IAuthDataManager Interface Specification



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

### Database – 단말기Database

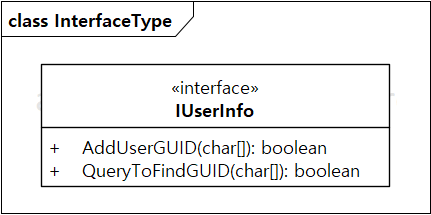
#### 인증 Database Specification



#### Interface List

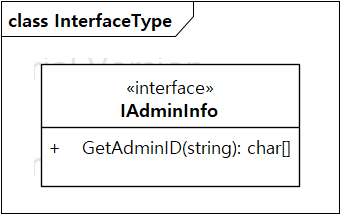
| Name | Kind | Description |
| --- | --- | --- |
| IUserInfo | Provided | 사용자 등록, 조회를 위한 인터페이스 |
| IAdminInfo | Provided | 아파트 운영자 정보를 등록, 조회하는 인터페이스 |

#### IUserInfo Interface Specification



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

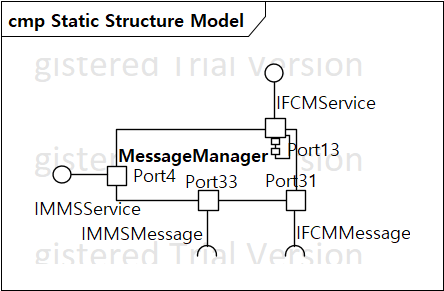
#### IAdminInfo Interface Specification



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

### MessageManager – external interface

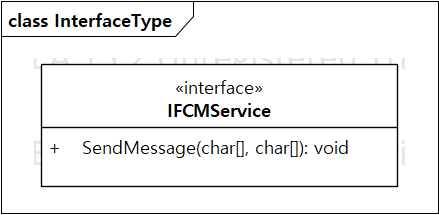
#### MessageManager Specification



#### Interface List

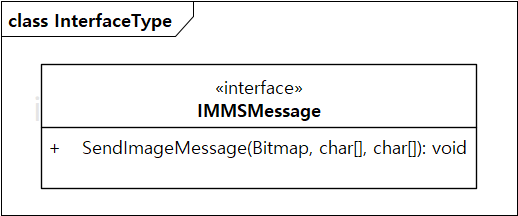
| Name | Kind | Description |
| --- | --- | --- |
| IFCMService | Provided | Push Message를 전송하기 위한 인터페이스 |
| IMMSService | Provided | MMS를 전송하기 위한 인터페이스 |

#### IFCMService Interface Specification



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

#### IMMSService Interface Specification

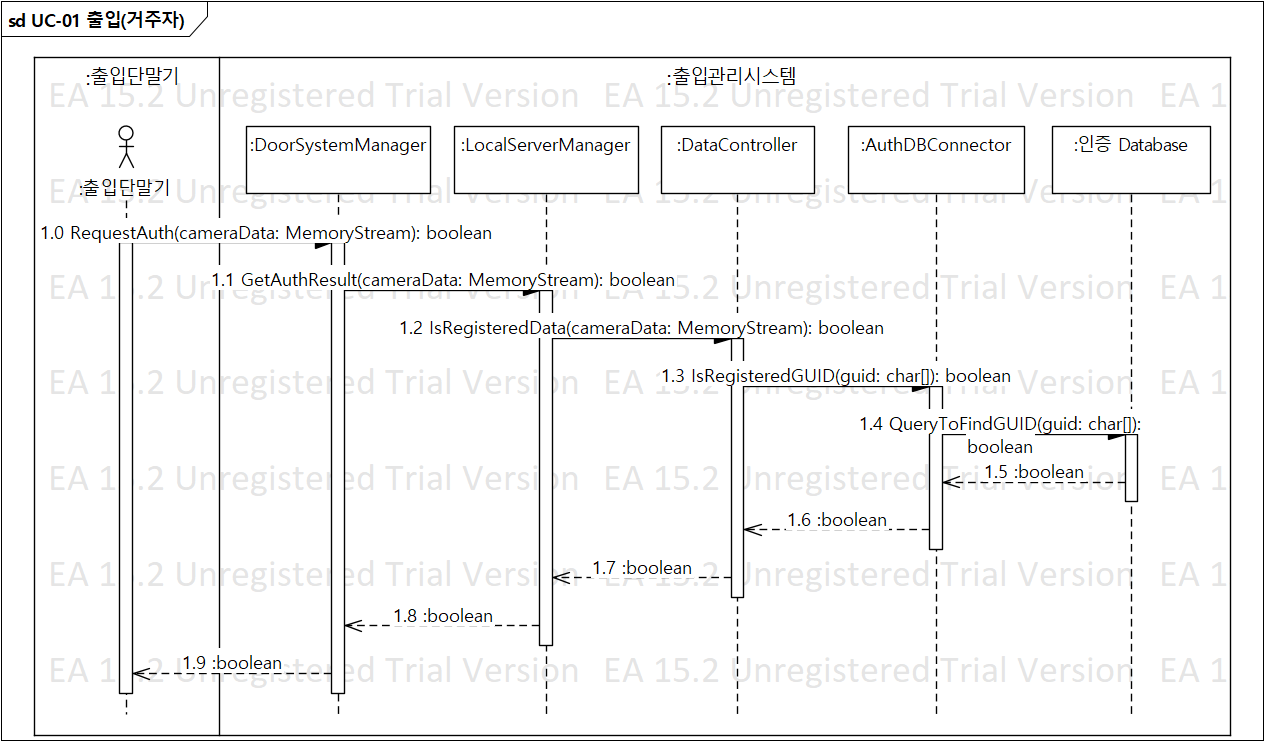


| Operation | Responsibility |
| --- | --- |
| SendMessage | MMS를 전송하는 오퍼레이션 |

## Behavior View

### UC-01 출입 Use Case Behavior Model

#### Behavior Diagram



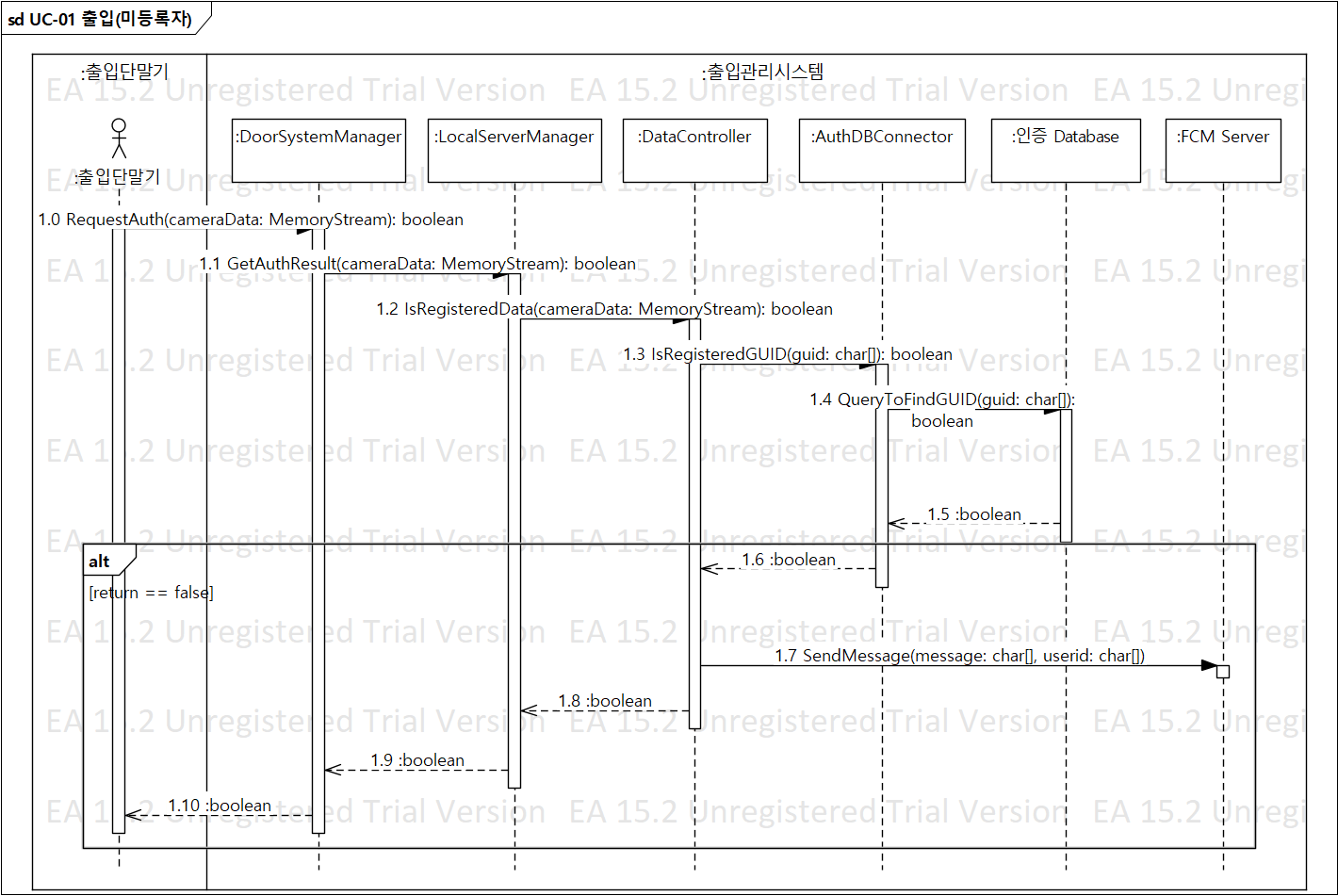
#### Behavior Description

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 얼굴인식서버로 전달되며 얼굴인식서버는 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

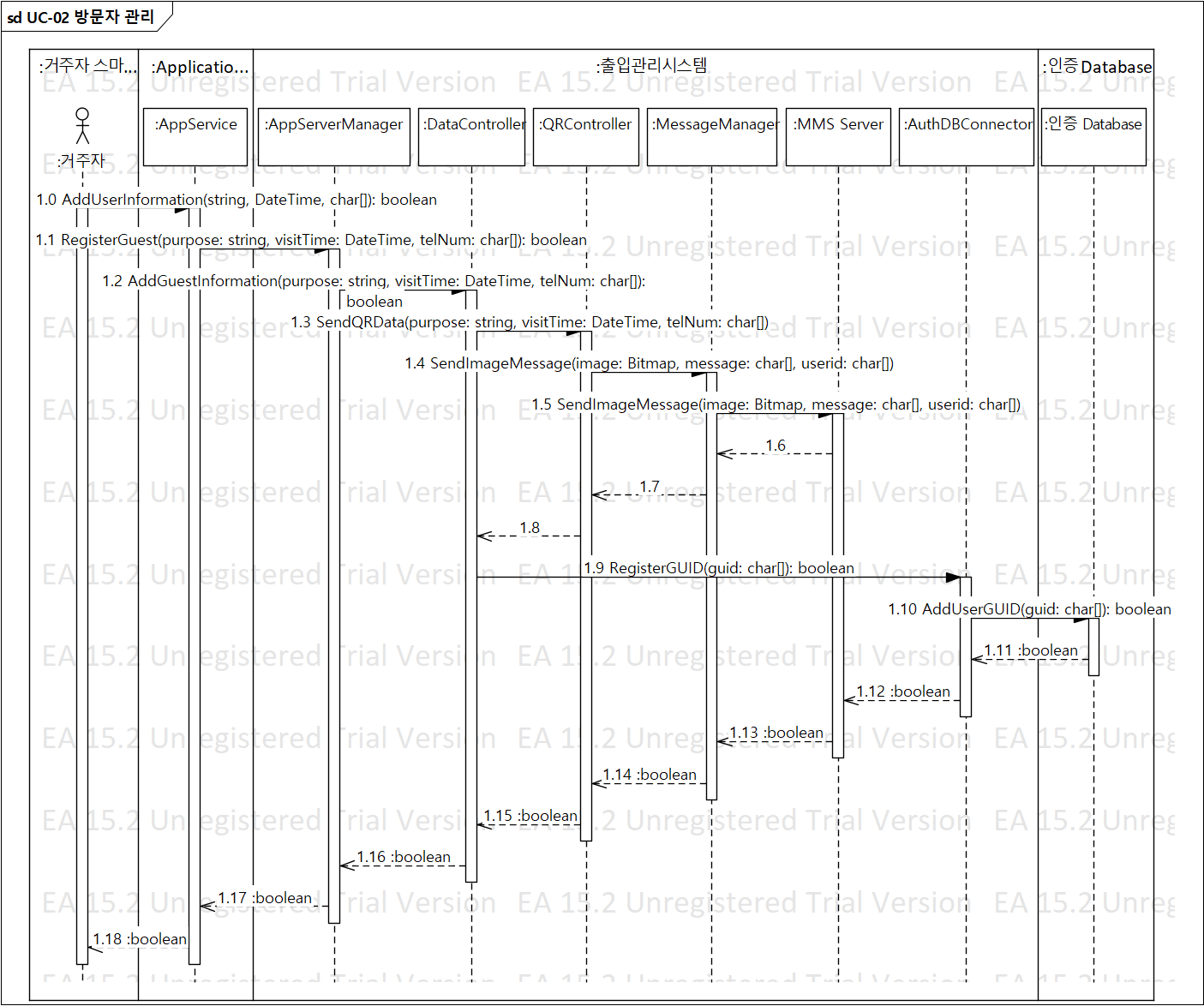


#### Behavior Description

| 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.

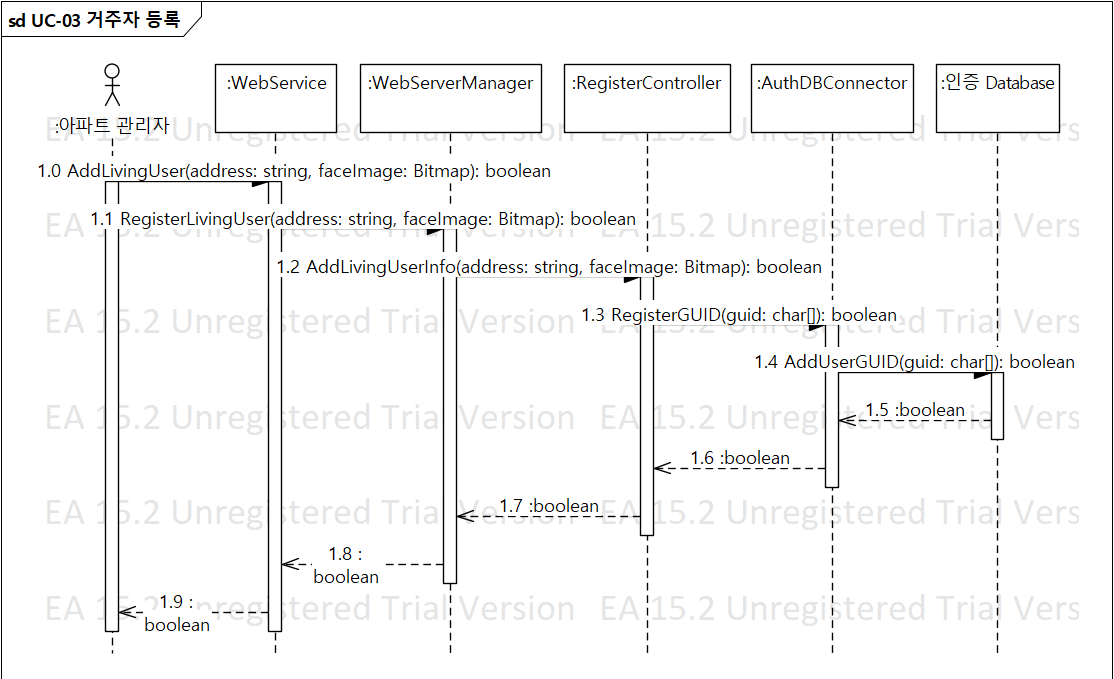
거주자가 스마트폰 어플리케이션을 통해 방문자를 등록하면, 해당 정보를 통해 QR코드를 생성하고 생성된 QR코드 이미지를 이미지 포멧으로 MMS를 전송한다.

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.

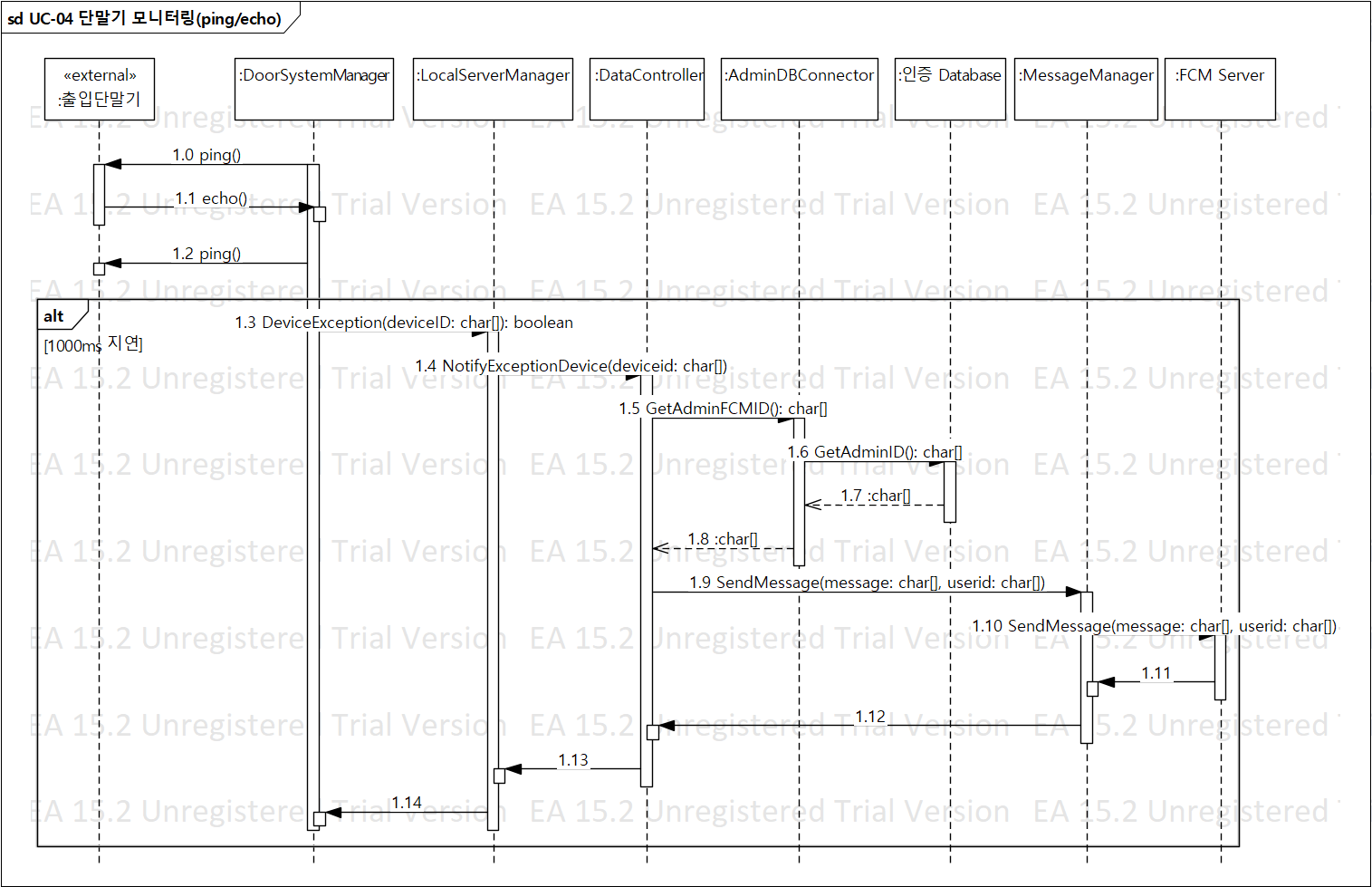
거주자로부터 동/호수, 얼굴사진 데이터를 받아 웹사이트를 통해 입력이 가능하다.

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

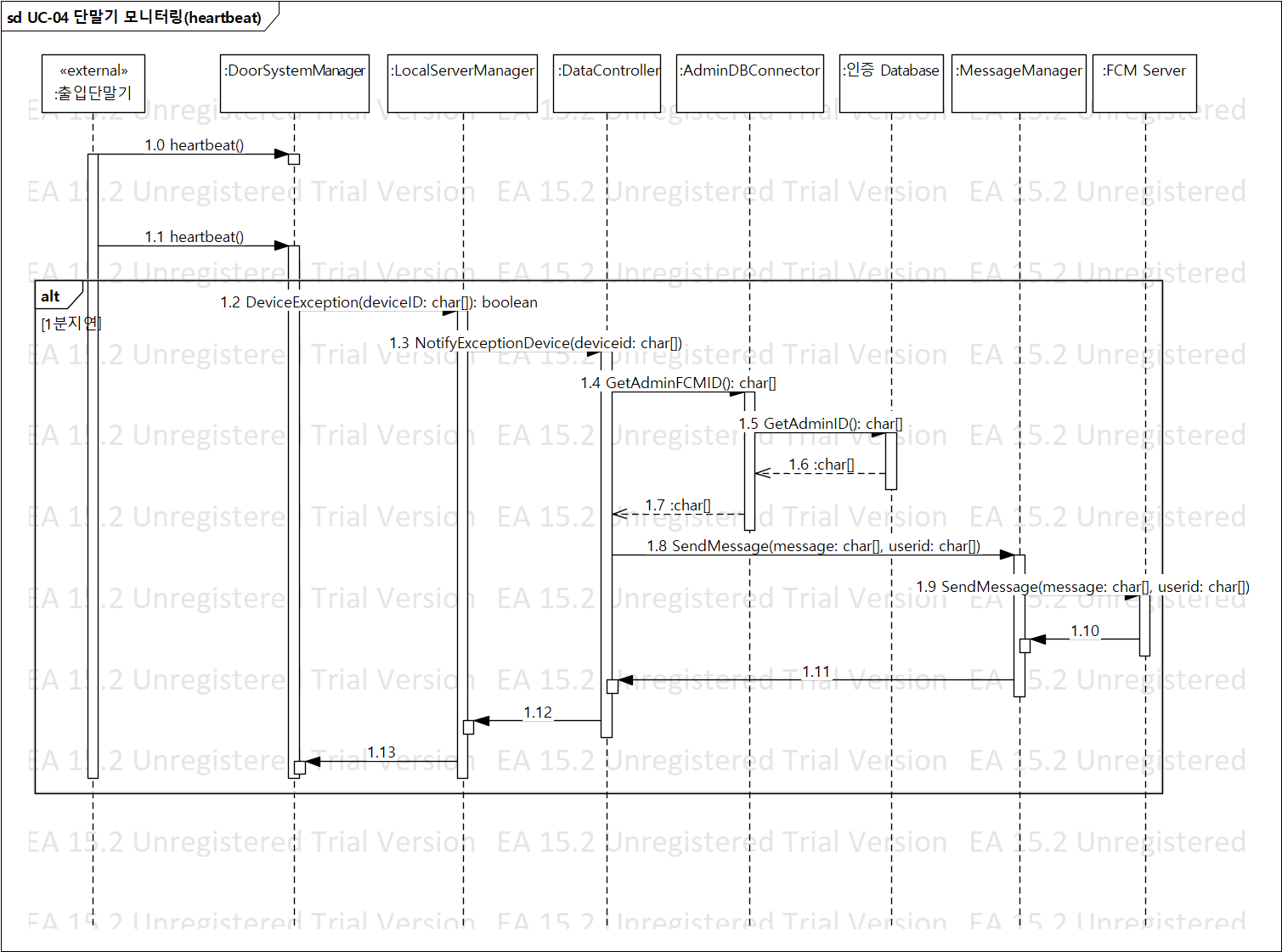
출입단말기는 제조사에 따라 지원하는 가용성 판단 인터페이스가 달라진다.

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.

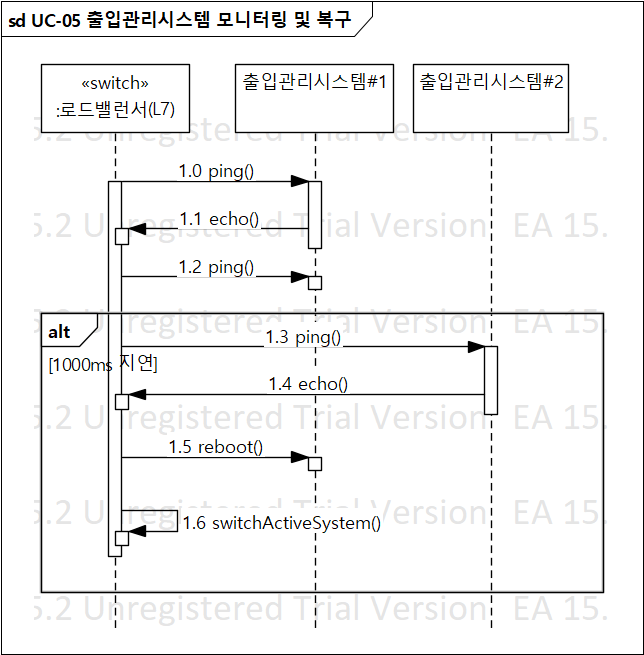
위 Sequence Diagram은 heartbeat방식의 출입단말기에 대한 가용성 판단 Sequence이다.

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.

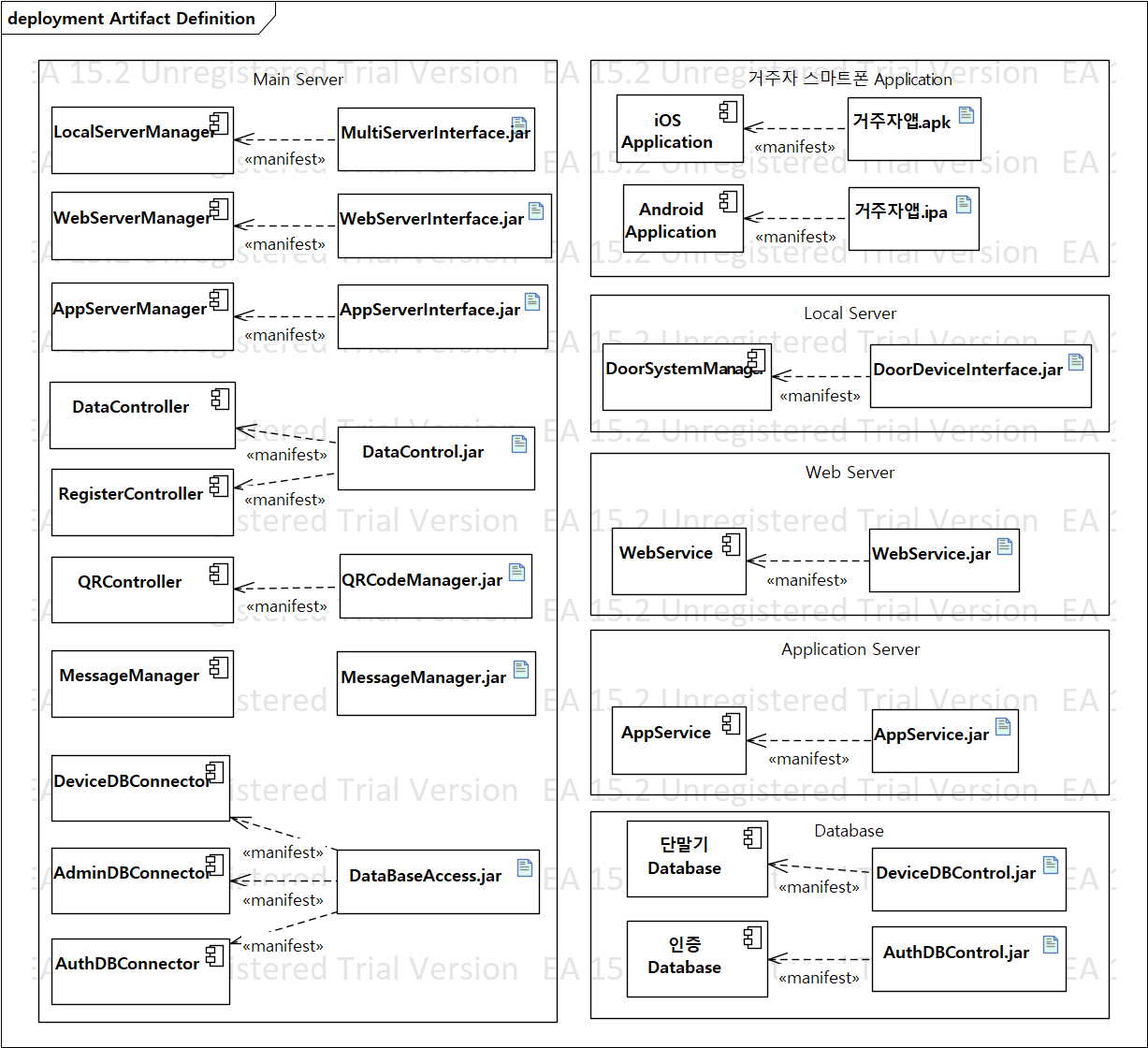
서버 장애가 탐지되면 대체서버로 역할을 이동함과 동시에 장애발생 서버는 재부팅을 진행한다.

| 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

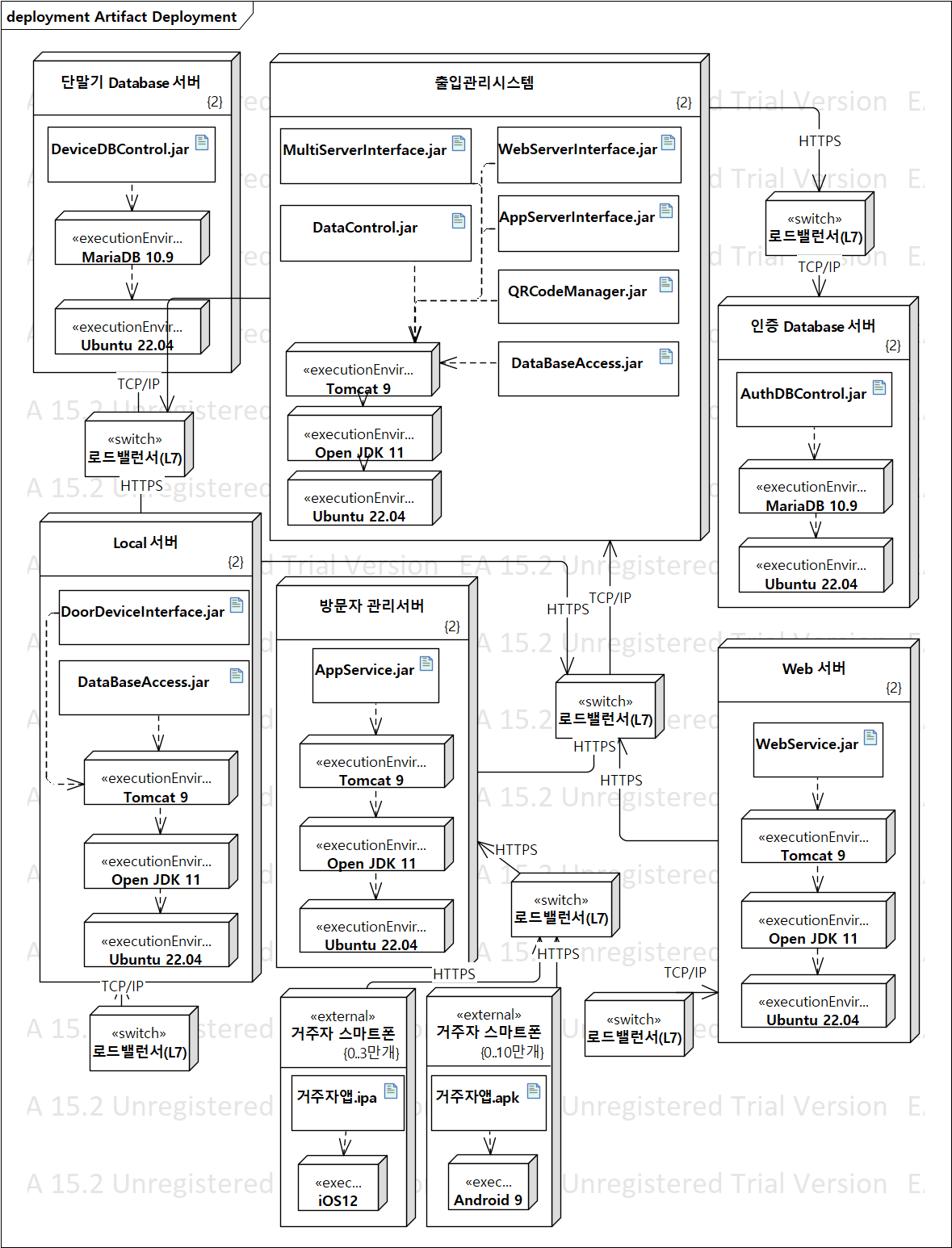
### Artifact Definition Model

#### Artifact Definition Diagram



### Artifact Deployment Model

#### Artifact Deployment Diagram



## Documenting Design Decisions

### Design Decision List

| ID | Title | 목표 QAs |
| --- | --- | --- |
| DD-01 | 얼굴인식 정확도 향상을 알고리즘 적용방식 | QA-01(얼굴 인식 정확도 향상) |
| DD-02 | 성능 향상을 위한 서버 부하 분산 | QA-02(침입 시도 통보 시간) |
| DD-03 | 시스템 모니터링 및 복구 | QA-03(시스템 모니터링 및 복구) |
| DD-04 | 단말기 장애 진단 | QA-04(단말기 이상 모니터링) |
| DD-05 | 출입단말기 연동방식 | QA-05(새로운 출입 단말기 지원) |

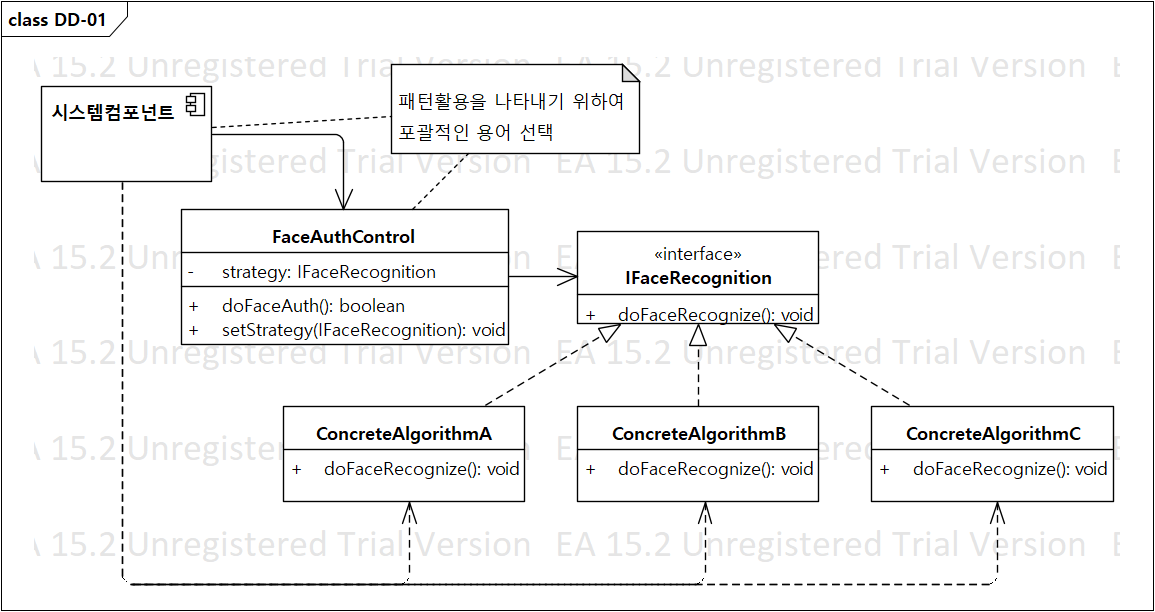
### DD-01 얼굴인식 정확도 향상을 위한 알고리즘 적용방식

#### Design Goal

본 시스템의 핵심기능인 [QA-01 얼굴 인식 정확도 향상]을 달성하기 위해서는 정확도 99%에 도달해야 한다. 이를 위하여 여러가지 알고리즘을 시스템에 적용한 뒤, 정확도를 높일 수 있는 방안을 모색하였다.

여러가지 알고리즘을 시스템에 적용하기 위한 전략으로는 Strategy Pattern을 사용한다.

<**Strategy Pattern을 통해 여러 알고리즘 탑재>**



Strategy Pattern은 여러 알고리즘을 별도의 클래스에 넣고, 해당 객체를 교체할 수 있도록 한다.

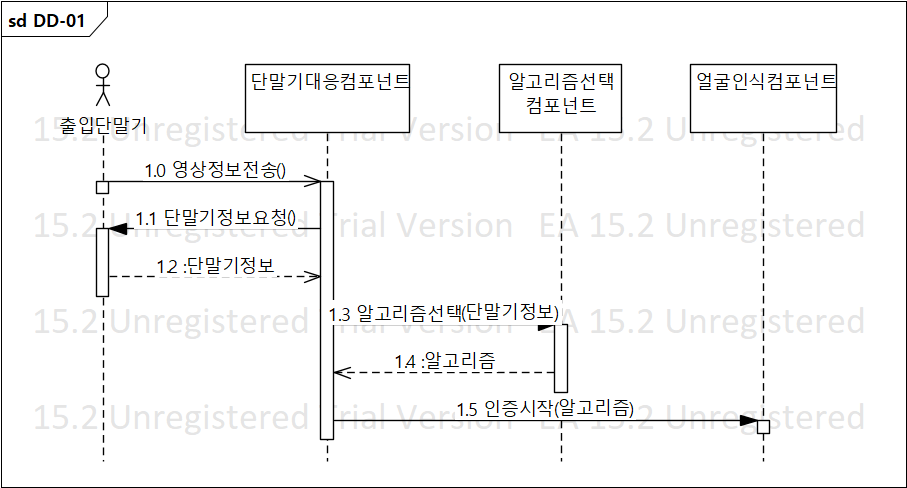
정확도가 높은 알고리즘 세가지를 선정하여, 시스템에 탑재하는 방식이다.

Algorithm A, B, C를 concrete하게 구현된 클래스 내에서 사용하도록 하였으며, 이를 interface로 추상화하였다.

여러 알고리즘을 같이 탑재한 후 아래 Design Approach에서 어떤식으로 활용할지를 선택할 수 있다.

#### Design Approach List

#### Design Approach #2 Description: 단말기별 최적 알고리즘 적용방식



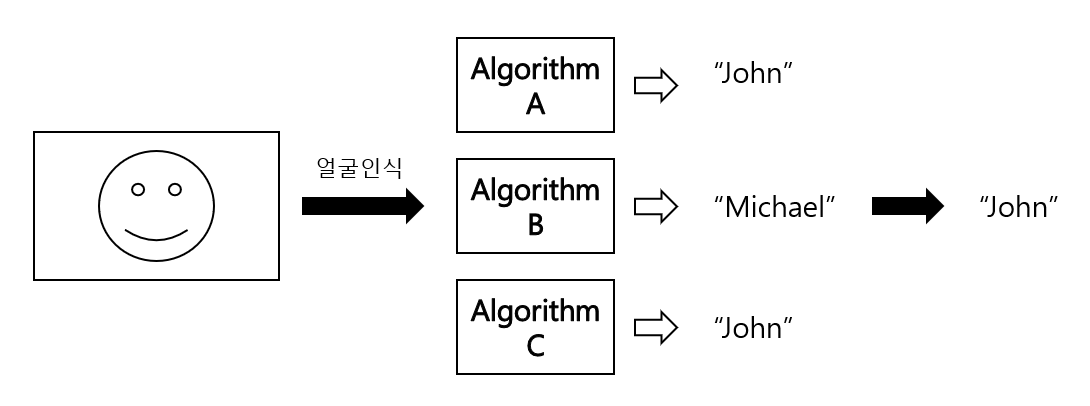
<아래 정보를 시스템에 입력한다>

|  | 알고리즘 A | 알고리즘 B | 알고리즘 C |
| --- | --- | --- | --- |
| 출입단말기 A | v |  |  |
| 출입단말기 B |  |  | v |
| 출입단말기 C |  | v |  |
| 출입단말기 D |  | v |  |
| … | … | … | … |

단말기별 특성을 고려하여, 단말기마다 적용알고리즘을 설정한 뒤, 해당 알고리즘으로 얼굴인식을 진행한다. 하나의 알고리즘만을 수행하므로 사용하는 리소스가 적고 시간도 적게 들어 성능 측면에서는 가장 좋다.

그러나, 단말기가 추가될 경우 단말기 특성에 따라 어떤 알고리즘이 적합한지 여부를 입력해야 하는 번거로움이 존재한다. 정확도 측면에서는 해당 알고리즘의 정확도에 의존하게 되므로 얼굴을 잘못 인식하는 경우 출입이 불가한 상황이 벌어질 수 있다.

#### Design Approach #3 Description: 여러 알고리즘 사용 (voting)



여러 알고리즘을 사용하여 얼굴을 인식하고, 그 결과들을 비교하여 가장 많이 나온 결과를 최종 인식된 얼굴정보로 채택하는 방식이다. 단일 알고리즘을 사용할 때보다 얼굴 인식의 정확도가 좋으며, 다양한 알고리즘을 많이 사용할수록 정확도가 점점 더 높아진다. 그러나 사용할 수 있는 리소스에 한계가 있고, 연산량이 증가하여 성능측면에서 손실이 많은 방식이다.

**\*. 통계적으로 알고리즘의 정확도가 높은 순서로 A, B, C를 책정하였으며, 모두 다른 결과값이 도출되는 경우 A의 결과를 사용한다**.

#### Decision and Rationale

| Quality Attribute | | Analysis | 단말기별 최적 알고리즘 적용방식 | 여러 알고리즘 사용(voting)  (Selected) |
| --- | --- | --- | --- | --- |
| ID | Title |
| QA-01 | 출입 인가 처리 정확성 | Pros/Cons | (+) 단말기에 따라 적절한 알고리즘을 통해 얼굴을 인식함으로써 정확도 향상 | (++) 여러 알고리즘의 결과를 통해 얼굴인식을 진행하여 인식 정확도 크게 향상 |
| QA-02 | 침입 시도 통보시간 | Pros/Cons | (+) 단일 알고리즘 사용으로 인하여 소요되는 처리시간에 손해가 발생하지 않음 | (--) 알고리즘의 개수만큼 성능에 미치는 영향이 커지게 됨 |
| QA-05 | 새로운 출입 단말기 지원 | Pros/Cons | (-) 단말기 추가에 따라 최적알고리즘을 등록해야 하는 번거로움이 존재함 | (+) 단말기 추가에 의한 정확도 저하 요인이 존재하지 않음 |

### DD-02 성능 향상을 위한 서버 부하 분산

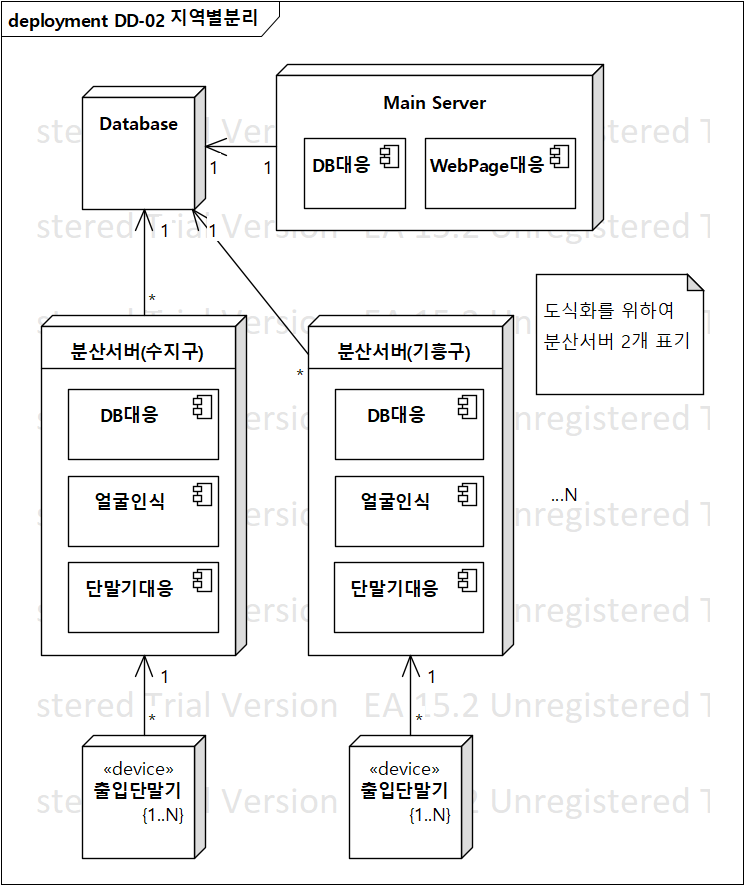
#### Design Goal

[QA-02 침입 시도 통보시간]을 달성하기 위하여, 내부 프로세스에 걸리는 부하를 최대한 줄이고자 한다. 부하를 줄이기 위해서는 resource의 부하 등으로 인 한 blocked time과, HW/SW 자원의 한계로 인한 processing time을 줄여야 한다. 적당한 비용의 HW/SW 자원을 사용하면서 많은 수의 출입단말기 연결로 인한 부하까지 고려해야 하므로, 서버에 직접 연결되는 기기의 수가 중요하다는 사실을 알 수 있다.

출입단말기로부터 영상을 전달받은 뒤, 가장 부하가 많이 걸리는 부분은 얼굴인식 알고리즘 관련 컴포넌트이다. 해당 컴포넌트를 중앙서버가 관리하지 않고, 여러개의 분산서버에 탑재 후 얼굴인식 관련사항을 처리하도록 한다.

#### Design Approach List

#### Design Approach #1 Description: 지역별 분리



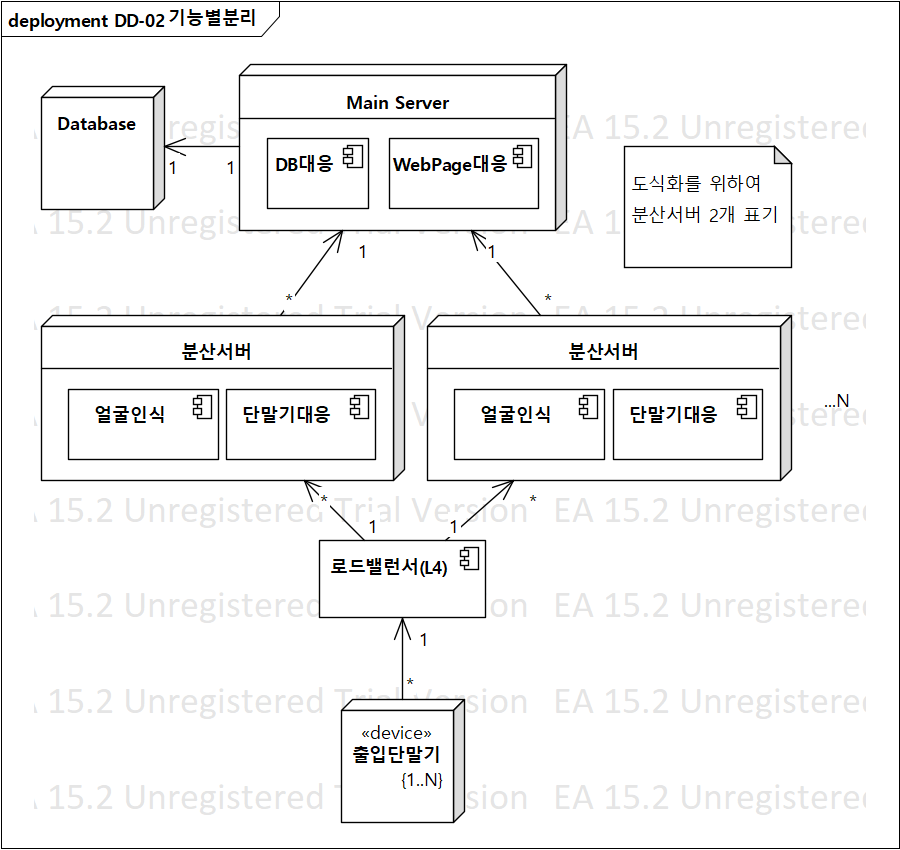
지역별로 서버를 설치하여 Main Server에 부하를 분할하였다.

용인시 전체 아파트에 시스템이 설치될 예정이므로, 각 구별로 서버를 분할한다. 이러한 설계로 인해 얻을 수 있는 가장 큰 장점은 분산서버와 출입단말기간 물리적 거리가 가까워 짐으로 인해 latency가 감소함을 알 수 있다.

각각의 분산서버는 단말기로부터 들어오는 이미지에서 얼굴인식 알고리즘, 인식된 데이터를 통하여 데이터베이스와 직접 통신하게 된다.

위 설계의 단점은 지역별로 담당하는 단말기의 수가 고르지 못한 경우 부하 분산의 loss가 발생한다는 점이고, 새로 추가되는 지역에 대하여 어려움이 있다는 것이다.

#### Design Approach #2 Description: 기능별 분리



출입단말기로부터 영상을 수신하고, 얼굴인식을 담당하는 역할을 분산서버가 수행하도록 한다.

상대적으로 부하가 큰 얼굴인식을 로드밸런서를 통해 진행함으로써, 부하분산에 대한 효율이 극대화 되는 설계이다.

분산되어 있는 서버에서 얼굴인식을 진행하고 나면 Main Server에 인증데이터가 등록되어 있는지 여부를 확인하는 과정을 거치고, 출입문의 개방 여부를 출입단말기에 알린다.

#### Decision and Rationale

| Quality Attribute | | Analysis | 지역별 분리 | 기능별 분리  (Selected) |
| --- | --- | --- | --- | --- |
| ID | Title |
| QA-02 | 침입시도 통보시간 | Pros/Cons | (+) 중앙서버에서 모든걸 처리하는 과정보다 부하분산을 통해 성능향상 | (++) 부하 분산을 통해 최적의 효율로 성능 향상 가능 |
| QA-03 | 시스템 모니터링 및 복구 | Pros/Cons | (-) 지역서버 단독으로만 존재하므로, 대체서버 없음 | (+) 로드 밸런싱 과정에서 부하분산시 가용성까지 확인이 가능하므로, 대체가능 서버가 항상 존재 |
| QA-04 | 단말기 이상 모니터링 | Pros/Cons | (-) 모니터링 기능도 부하의 일종으로 해당지역의 출입단말기 수에 따라 부하 정도가 다를 수 있음 | (+) 모니터링 기능의 부하도 분산되는 효과로 이점이 존재 |
| QA-05 | 새로운 출입 단말기 지원 | Pros/Cons | (-) 새로운 지역에 시스템이 도입될 때, 출입단말기 수와 상관없이 서버가 설치되어야 함 | (+) 출입단말기가 추가되면 해당 부하에따라 로드밸런서는 scale out을 지원할 수 있으므로, 출입단말기 추가 지원이 용이함 |

### DD-03 시스템 모니터링 및 복구

#### Design Goal

[DD-02 성능 향상을 위한 서버 부하 분산]을 통해 Main Server와 기능적 분산서버가 도입이 결정되었다.

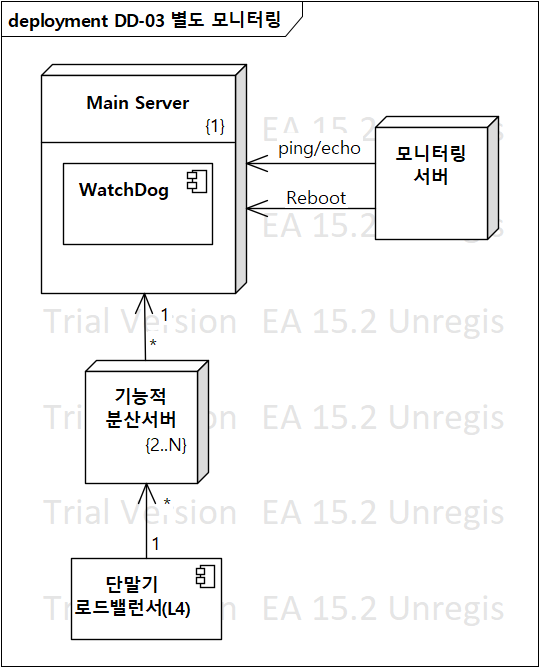
전체 시스템 중 기능적으로 분산된 서버는 로드밸런서를 통해 부하분산은 물론 가용성까지 확보할 수 있게 되었다.

Main Server는 기능적 분산서버에서 수행한 부하높은 얼굴인식 알고리즘의 결과를 받아 데이터베이스로 부터 인증 절차를 담당하게 된다.

이 때 Main Server의 가용성 확보를 위한 전략을 결정하고자 한다.

#### Design Approach List

#### Design Approach #1 Description: 별도의 모니터링 서버를 활용하여 시스템 모니터링 및 복구



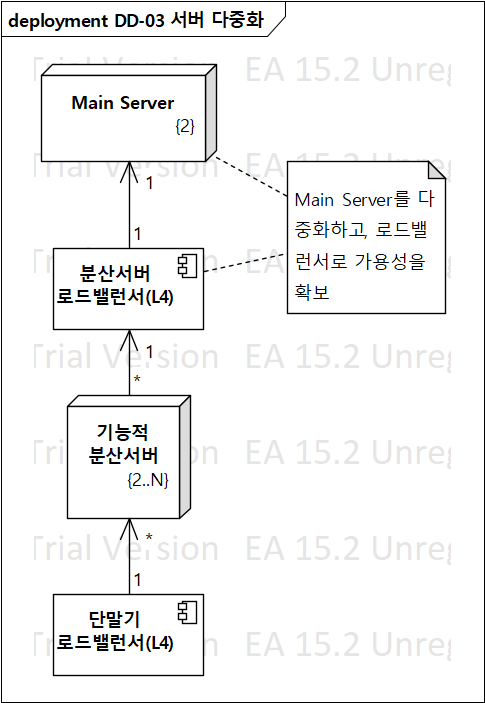
먼저 별도의 모니터링 서버를 통하여, ping/echo방식을 통해 가용여부를 확인한 후 Reboot 명령으로 재시작하는 방법이다.

모니터링 서버의 구현방식이 간단하고, Main Server에 존재하는 Watch Dog 컴포넌트에만 interaction하기 때문에 구조적으로 단순하다.

하지만, Main Server의 Reboot 시간은 평균적으로 1분내외로 예상되며, 장애 발생시 해당 시간동안 Main Server는 block 상태가 된다.

이 경우 시스템이 도입된 전체 아파트의 출입이 제한될 수 있는 단점이 있다.

#### Design Approach #2 Description: 서버 다중화 및 로드밸런서 활용으로 가용성 확보



Main Server를 다중화하고, 로드밸런서를 통해 부하를 분산한다. 이미 큰 부하는 분산서버에서 동작하고 있기 때문에, 2개의 Main Server로 구성하였다.

로드밸런서를 통해 장애가 발생한 Main Server는 Reboot을 수행하게 되며, 나머지 서버로 역할을 수행하기 때문에 시스템의 block상태는 없어지게 된다.

#### Decision and Rationale

| Quality Attribute | | Analysis | 별도의 모니터링 서버를 활용하여 시스템 모니터링 및 복구 | 서버 다중화 및 로드밸런서 활용으로 가용성 확보 |
| --- | --- | --- | --- | --- |
| ID | Title |
| QA-03 | 시스템 모니터링 및 복구 | Pros/Cons | (+) 모니터링 서버가 시스템의 가용성을 확인하고 장애 발생시 reboot 동작 | (++) 다중화된 서버에 로드밸런서가 지속적으로 가용한 서버를 선택하고 있으며 장애발생시 복구가능 |
| QA-02 | 침입 시도 통보 시간 | Pros/Cons | (-) 장애발생시 reboot되는 시간동안 시스템 block됨 | (+) 침입 시도에 대한 검출기능이 상시 동작가능 |
| QA-04 | 단말기 이상 모니터링 | Pros/Cons | (-) 장애발생시 reboot되는 시간동안 시스템 block으로 인하여 모니터링 기능이 정상동작되지 않음 | (+) 단말기 모니터링 기능이 상시 동작 가능 |

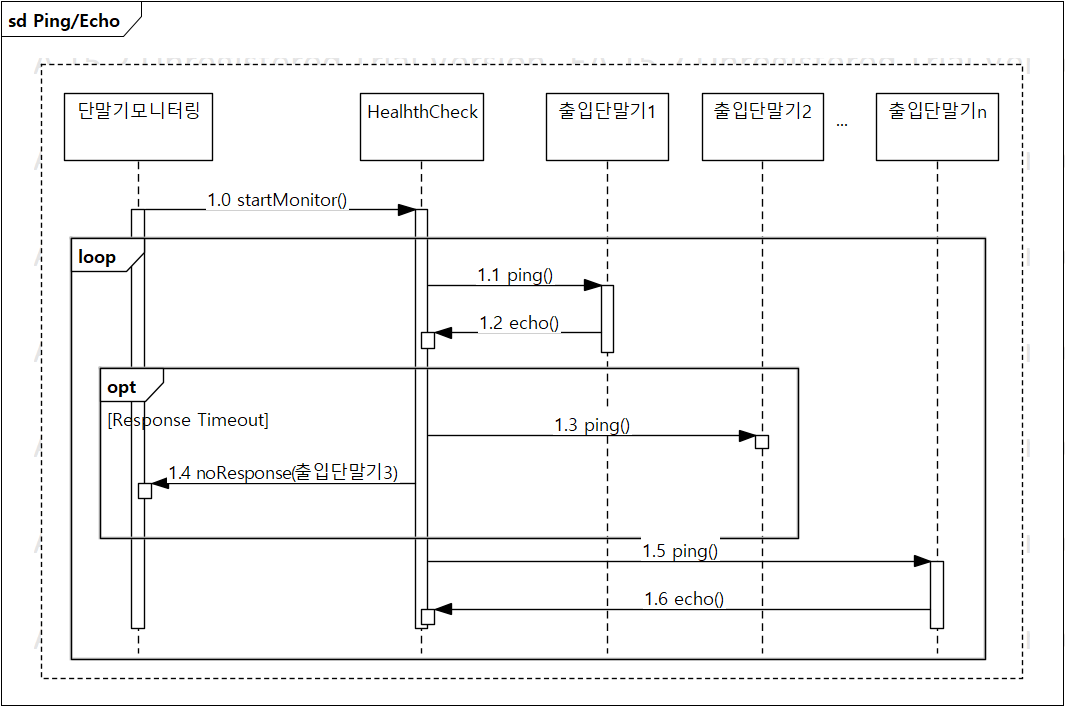
### DD-04 단말기 장애 진단

#### Design Goal

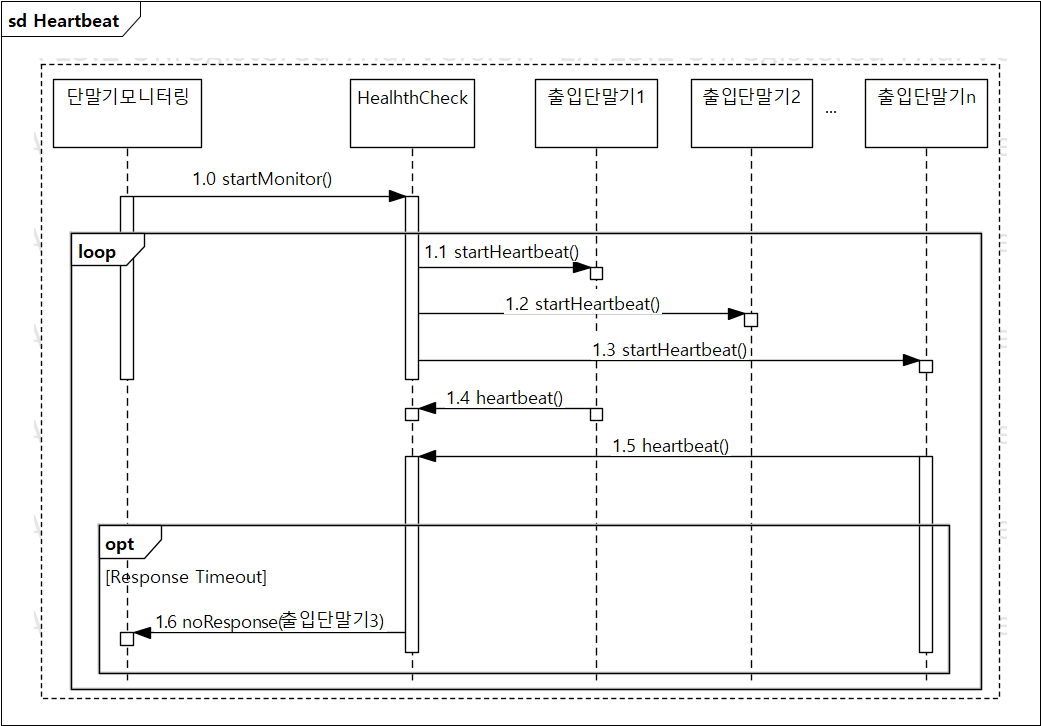
[QA-04 단말기 이상 모니터링]을 지원하기 위하여, 출입단말기에 가용성을 확인하고, 아파트 관리자에게 통보하는 기능을 제공한다.

출입단말기는 모델에 따라 ping/echo방식과 heartbeat방식으로 가용성 인터페이스를 제공하고 있다. 따라서 시스템은 ping/echo방식, heartbeat방식을 모두 지원해야 하며, 두가지 방식의 동작흐름은 다음과 같다.

<ping/echo>



<heartbeat>



시스템은 기본적으로 두 가지 방식을 모두 지원하도록 설계한다.

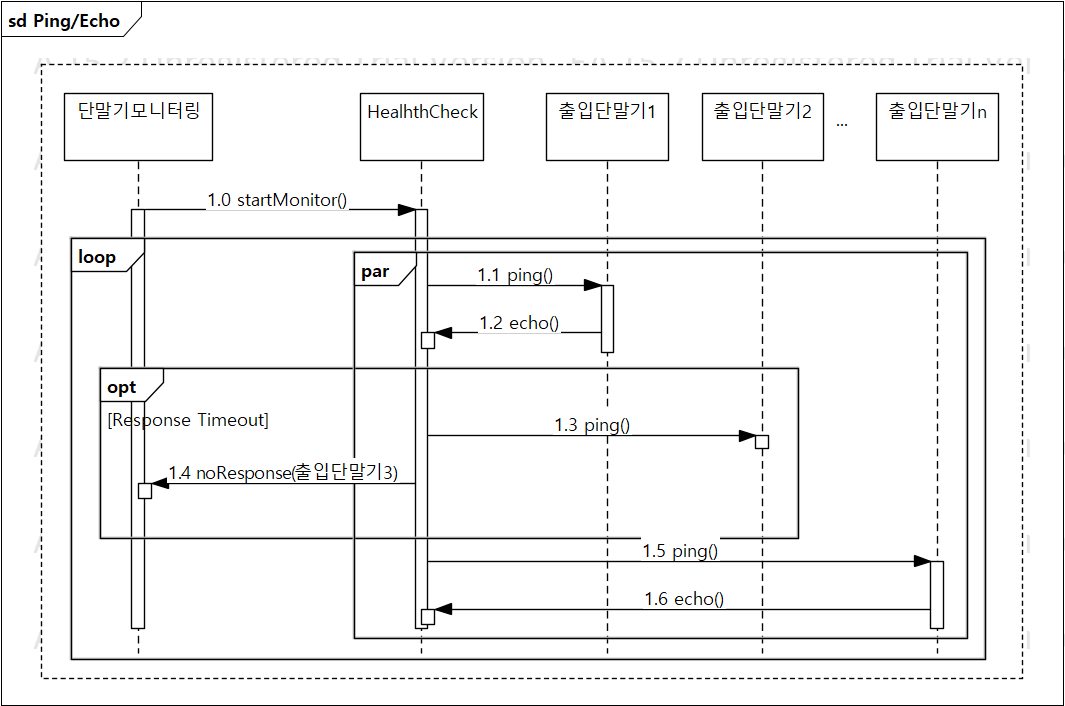
heartbeat방식은 단말기에서 시스템으로 보내는 주기적 신호이므로, 단말기에 설정된 내용대로 동작한다.

ping/echo방식은 시스템에서 부하를 관리할 수 있는 방식이다. ping의 송신주기를 설정할 수 있으며, Multi Threading방식을 통해 여러 단말에 한꺼번에 가용성확인이 가능하다.

본 시스템에서는 Multi Threading방식을 사용하여 ping을 송신하기로 하였으며, 아래 두가지 Design Approach를 통해 설계하였다.

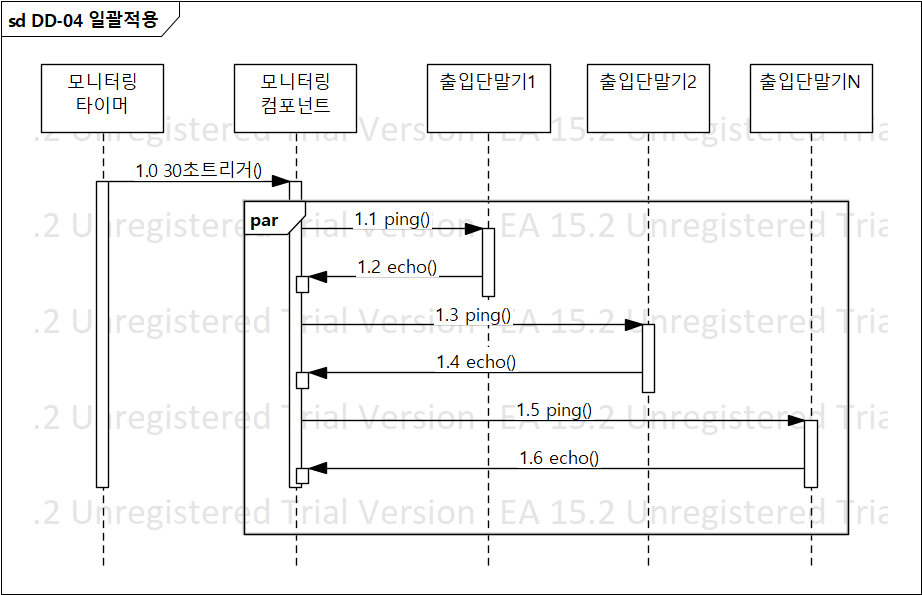
\*. heartbeat방식의 출입단말기는 단말기 설정을 따라야 하며, 여기서는 ping/echo방식에 대한 설계방법을 고민하였다.

<ping/echo Multi Threading>



#### Design Approach List

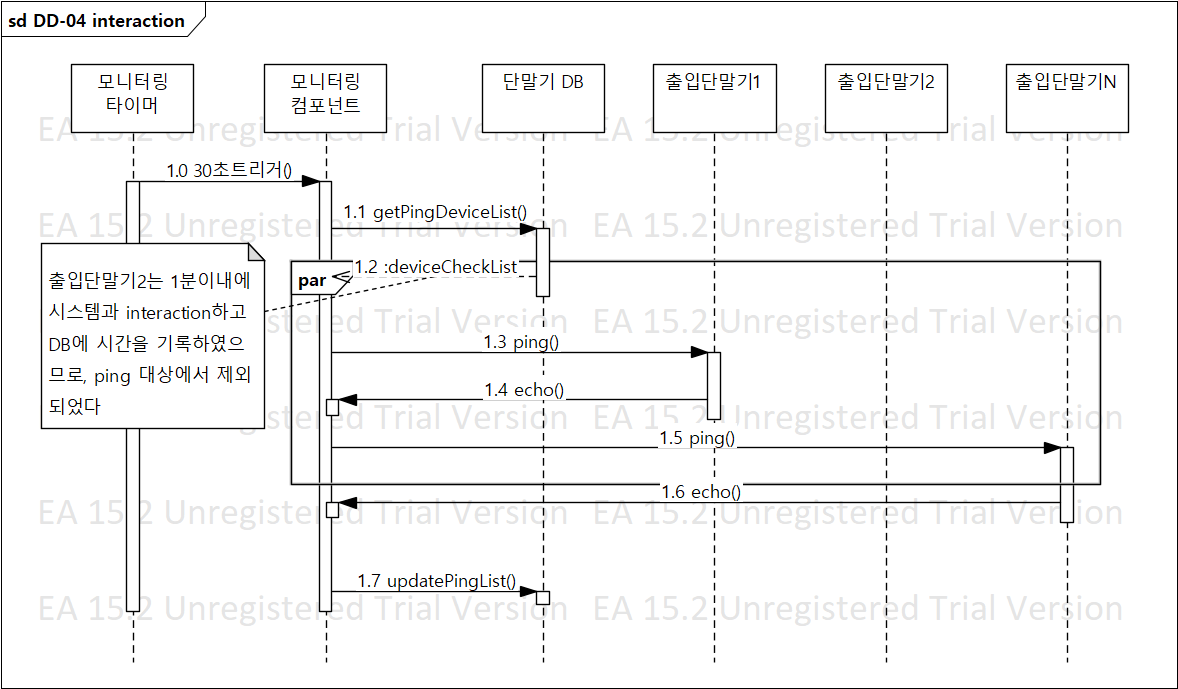
#### Design Approach #1 Description: 모든 단말기에 30초마다 ping/echo 적용



출입단말기의 가용성을 확인하기 위하여, 일정시간마다 각 분산서버에서는 ping/echo를 지원하는 모든 단말기에 ping신호를 보낸다.

가장 일반적이고 가장 정확한 방법으로, 모든 단말기의 echo신호를 통해 정상동작 여부를 판단할 수 있으며, 장애 발생시 검출이 용이하고 구현이 간편하다.

#### Design Approach #2 Description: 단말기와의 마지막 interaction을 기준으로 선택적 ping/echo 적용



시스템은 모든 출입단말기와의 마지막 interaction 시간을 단말기 DB에 기록한다.

따라서 모니터링 타이머의 트리거 시점에 단말기 DB로 부터 1분이내에 interaction이 없었던 단말기 목록을 받아오고, 해당 목록에 존재하는 출입단말기에만 ping신호를 보낸다.

어떠한 interaction이라도 단말기의 가용성을 판단하는 기준이 될 수 있으며, 실제 ping을 보내는 시점에는 그만큼 부하를 줄일 수 있는 방법이다.

#### Decision and Rationale

| Quality Attribute | | Analysis | 모든 단말기에 1분마다 ping/echo 적용 | 단말기와의 마지막 interaction을 기준으로 선택적 ping/echo 적용  (Selected) |
| --- | --- | --- | --- | --- |
| ID | Title |
| QA-04 | 단말기 이상 모니터링 | Pros/Cons | (+) 모든 단말기의 가용성을 판단할 수 있다. | (+) 모든 단말기의 가용성을 판단할 수 있다. |
| QA-02 | 침입 시도 통보 시간 | Pros/Cons | (--) ping/echo신호로 인한 부하가 걸리는 시점에 약간의 성능 저하가 발생할 수 있다.. | (-) ping을 보내는 단말기 수가 줄어들어 ping/echo신호로 인한 부하를 줄일 수 있다. |
| QA-03 | 시스템 모니터링 및 복구 | Pros/Cons |  |  |
| QA-05 | 새로운 출입 단말기 지원 | Pros/Cons |  |  |

### DD-05 출입단말기 연동방식

#### Design Goal

새로운 아파트에도 지속적으로 시스템을 도입할 예정이다.

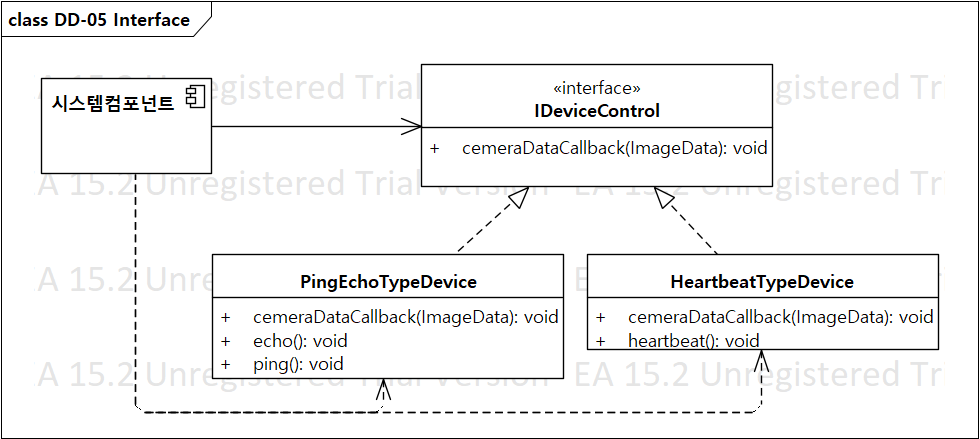
따라서 지속적으로 단말기 인터페이스를 맞춰줘야하는 [QA-5 새로운 출입단말기 지원]을 만족해야 한다.

단말기의 인터페이스가 변경되는 경우 시스템에서는 최소한의 비용과 시간으로 해당 단말기를 지원해야 하는 상황이다.

단말기와 직접적으로 interaction하는 부분에 아래와 같은 디자인 패턴을 적용하여 가장 효율적인 방법으로 단말기를 추가할 수 있는 방법을 찾아보았다.

#### Design Approach List

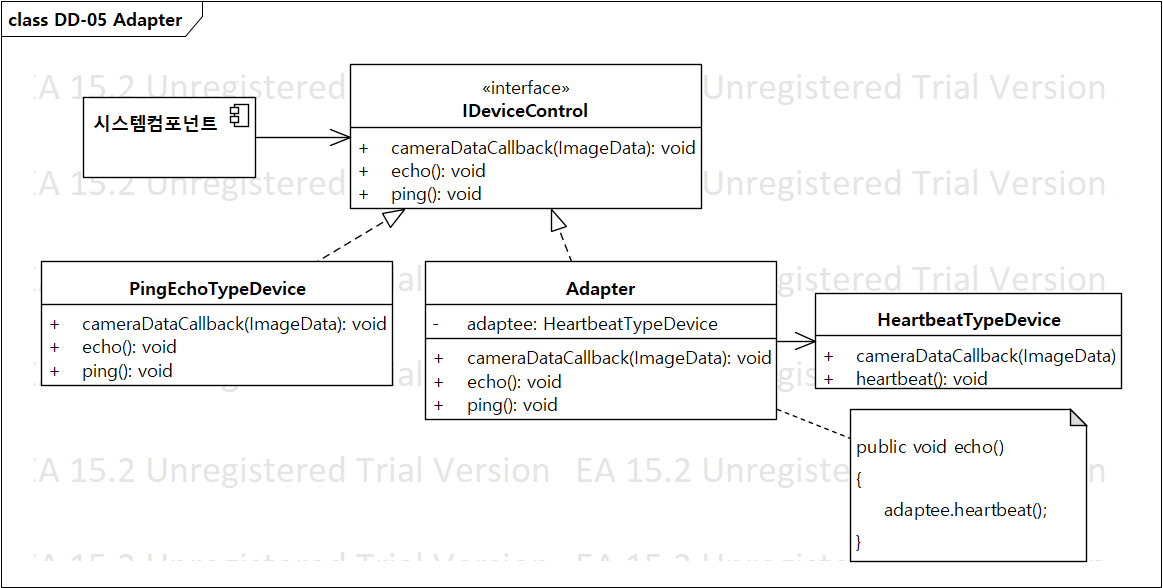
#### Design Approach #1 Description: 공통부 Interface



다형성을 이용하여 시스템은 인터페이스 객체만을 가지고 각 단말기 마다 의존성을 주입하며 사용한다. 즉, 이 방법으로 구현하는 경우 추가되는 단말기 인터페이스에 대하여 수정사항이 생기는 부분을 Interface, 단말기 대응 구체클래스로 제한할 수 있다.

하지만 공통부가 비교적 적고, 단말기 고유의 함수가 여전히 존재함으로 인하여 시스템은 구체클래스의 형태를 알아야만 하고, 추가되는 자료형에 시스템의 수정도 발생할 여지가 있다.

#### Design Approach #2 Description: Adapter Pattern 적용



[Design Approach #1]과 같이 다형성을 사용하되, Adapter Pattern을 적용하여 가장 범용적으로 사용할 수 있는 함수를 묶어 인터페이스화 한다.

새로운 형태의 단말기가 추가되면 동일한 Adapter 내에서 처리 가능한지를 확인하고, 큰 변경점이 발생하는 경우에는 Additional Adapter 형태가 생겨나게 된다.

위 그림에서는 heartbeat 신호를 echo함수에서 핸들링하게 되면서, [DD-04 단말기 장애진단]에서 선택된 최종 interaction시간이 반영된다. 따라서 Adapter의 ping신호는 자동으로 보내지 않게되는 이점이 있다.

#### Decision and Rationale

| Quality Attribute | | Analysis | 공통부 Interface | Adapter Pattern적용  (Selected) |
| --- | --- | --- | --- | --- |
| ID | Title |
| QA-05 | 새로운 출입 단말기 지원 | Pros/Cons | (+) 새로운 출입 단말기 지원시 수정의 폭이 작다. | (++) 새로운 출입 단말기 지원시 수정의 폭이 매우 작다. |
| QA-04 | 단말기 이상 모니터링 | Pros/Cons |  | (+) heartbeat신호를 echo에서 처리하여, 모니터링 시스템 구현이 단순화 된다. |
| QA-02 | 침입 시도 통보 시간 | Pros/Cons |  | (-) Adapter도입으로 인하여 성능저하의 가능성이 있다. |

# Component Level Design Description

## *Component 1* Design Description

### Overview

### Static Structure Diagram

### Element List

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

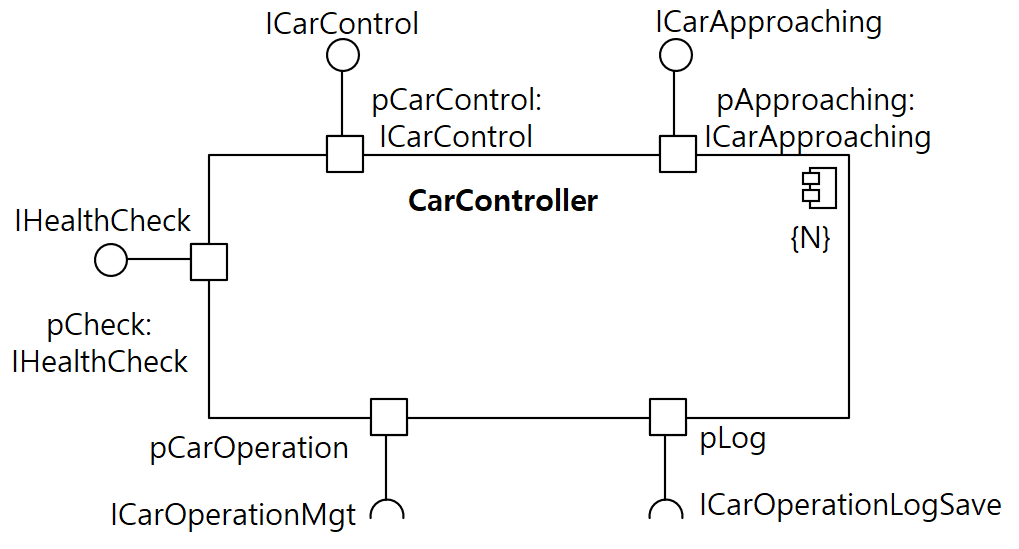
### Design Rationale

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

## CarController Design Description

### Overview

<작성 방법>



**각 인터페이스를 Class diagram으로 표시**

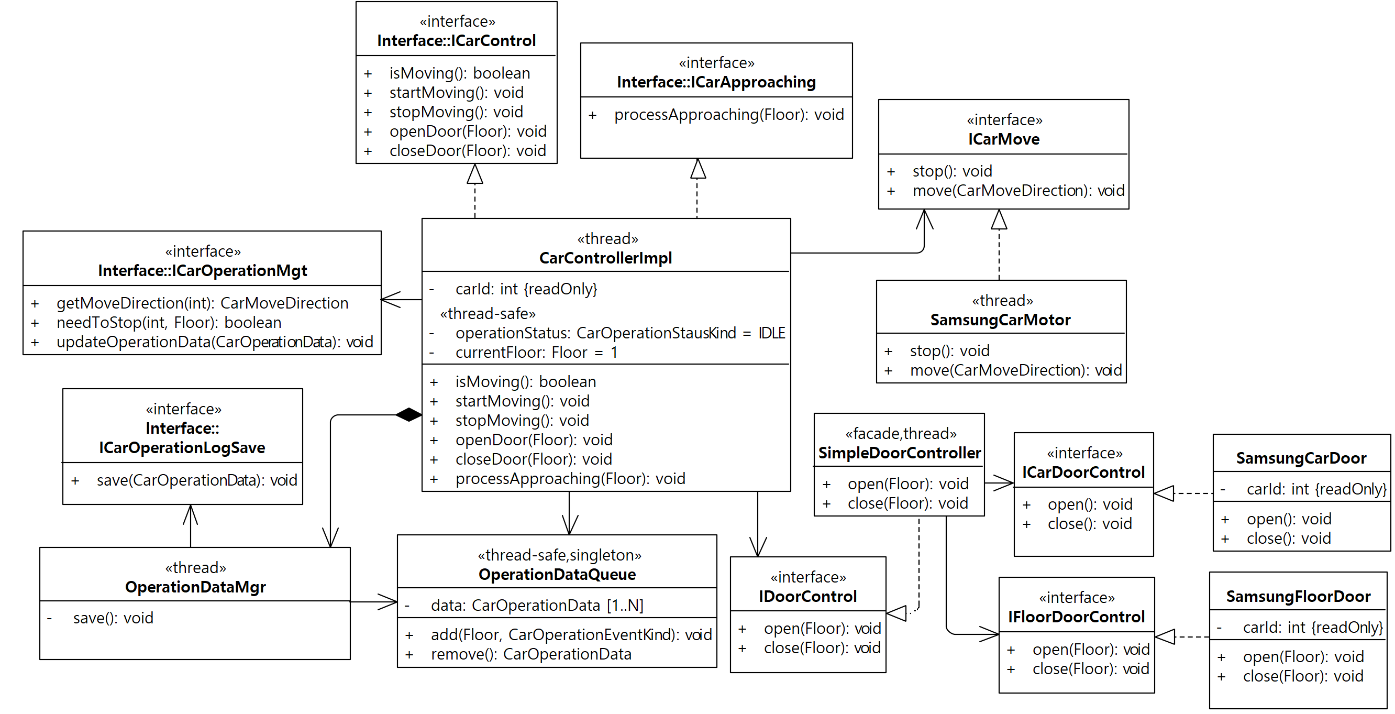
| 항목 | 설명 |
| --- | --- |
| 개요 | 이 컴포넌트의 역할을 한/두 줄로 간략하면서 명확하게 서술한다. |
| 컴포넌트 기능 요구사항 | 이 컴포넌트에 부여된 기능 요구사항을 서술한다. Provided Interface를 중심으로 구체적으로 설명한다. |
| 컴포넌트 품질 요구사항 | 이 컴포넌트가 만족해야 할 품질 요구사항을 서술한다. 시스템의 QA를 충족시키기 위하여 이 컴포넌트가 가져야할 QA(성능, 유지보수성)을 구체적으로 서술한다. |

### Static Structure Diagram

<작성 방법>

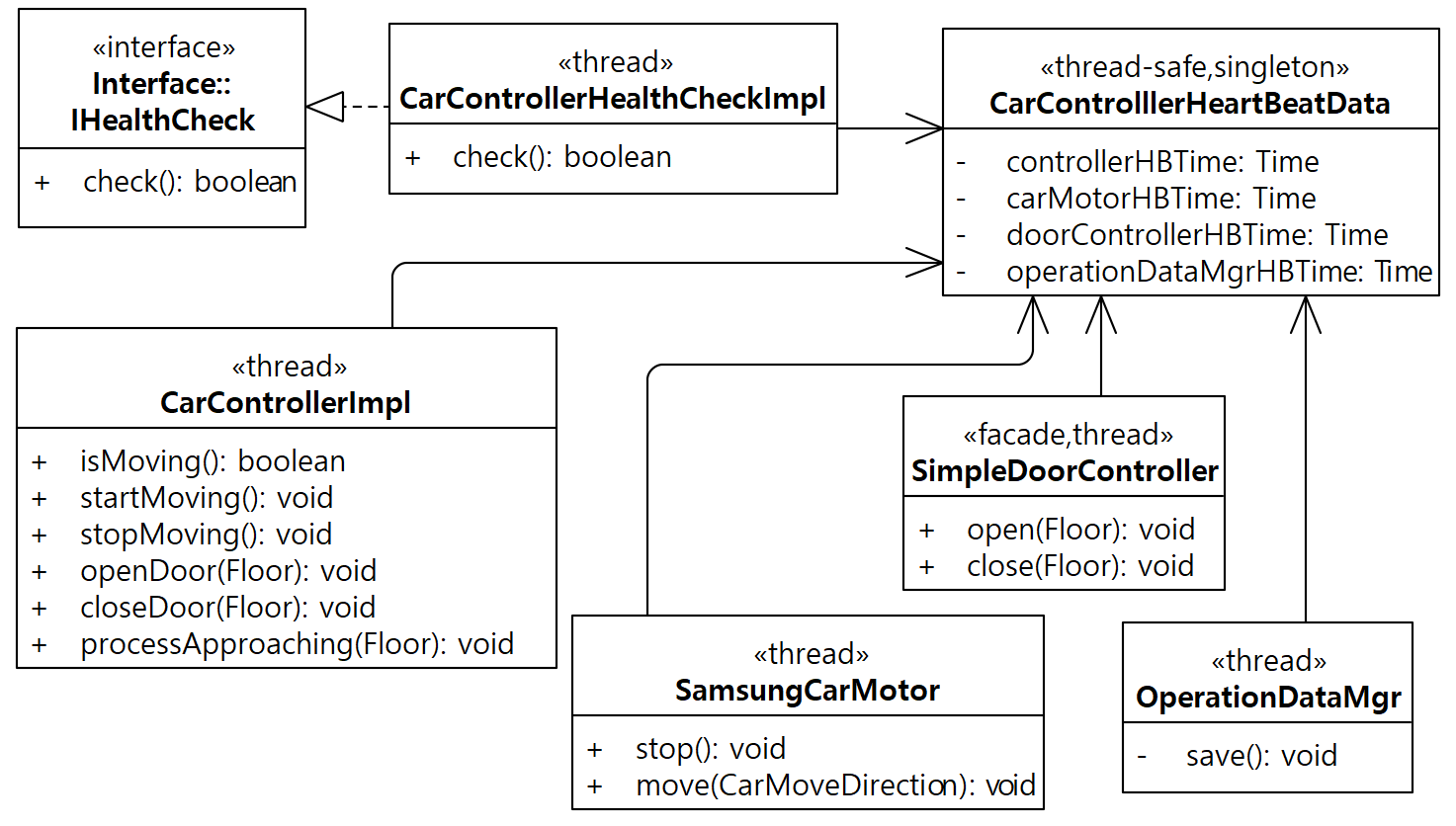
* 앞에서 명세한 기능 및 품질이 Static Structure Diagram을 통해서 구체화되었는가?
* Component를 구현하기 위하여 필요한 Class 및 Interface들이 모두 식별되었는가? 즉 Component에 주어진 기능 및 품질 요구사항을 충족시키기 위하여 필요한 설계가 수행되었는가?
  + 성능을 고려하면 multi-threading, thread-safe queue 등이 필요함
  + 유지보수성을 고려하면 응집도, SOLID 등을 적용해서 세분화 필요함
* Class 간의 관계(generalization, association, 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