**IoT 기반 스마트 엘리베이터 시스템 개발기능설계서**

**(요구기능명: 다수의 엘리베이터의 효율적 사용을 위한 알고리즘 개발)**

**(요구기능 ID: smart\_elevator\_001)**

**(참조파일명: [CS1415] 요구분석정의서 V1.0)**

문서번호 :

Version 1.0

**개정 이력**

**제.개정내역**

|  |  |  |  |
| --- | --- | --- | --- |
| **버전** | **승인일자** | **개요** | **작성자** |
| 0.1 | 2020.06.10 | 상세설계서 초안 | 김태균 |
| 0.2 | 2020.06.12 | UI, UX 수정 | 이도현 |
| 1.0 | 2020.06.14 | 요구 기능 수정 | 이윤호 |

**배포이력**

|  |  |  |
| --- | --- | --- |
| **버전** | **배포일자** | **배포처** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**검토이력**

|  |  |  |  |
| --- | --- | --- | --- |
| **버전** | **검토일자** | **검토방법** | **검토자** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |
| --- |
| **목 차** |

[**1.**](#_heading=h.1fob9te) **요구기능 설명 4**

[**1.1.**](#_heading=h.2et92p0) **주요 상세기능 설명 4**

[**1.2.**](#_heading=h.tyjcwt) **요구기능 동작 절차 4**

[**1.3.**](#_heading=h.3dy6vkm) **동작 규칙 5**

[**1.4.**](#_heading=h.1t3h5sf) **가정(Assumptions) 5**

[**2.**](#_heading=h.4d34og8) **주요 상세기능 #1 6**

[**2.1.**](#_heading=h.2s8eyo1) **기능 설계 6**

[**2.2.**](#_heading=h.17dp8vu) **요소 설계 6**

[2.2.1. Class 6](#_heading=h.3rdcrjn)

[2.2.2. 저장소 7](#_heading=h.26in1rg)

[**3.**](#_heading=h.lnxbz9) **주요 상세기능 #2 8**

[**3.1.**](#_heading=h.35nkun2) **UI 설계 8**

[3.1.1. 입력 UI 8](#_heading=h.1ksv4uv)

[3.1.2. 출력 결과 9](#_heading=h.44sinio)

1. **요구기능 설명**

* 다수의 엘리베이터 효율적 사용의 핵심 이슈는 “데이터의 수집, 추출 및 분석"이라고 할 수 있다. 엘리베이터의 사용은 기본적으로 다수의 층, 다수의 엘리베이터, 다수의 사용자로 인해 발생한다. 이 점을 착안하여 엘리베이터 사용을 이루는 3가지의 큰 요소들에 대한 데이터의 수집과 해당 데이터의 분석을 통해 엘리베이터의 효과적인 사용을 도모한다.
* 다수의 층에 대한 데이터는 각 층의 각 엘리베이터 전면부의 인원 계수용 카메라를 통해 수집한다. 다수의 엘리베이터에 대한 데이터는 엘리베이터의 제어반을 통해 현재 층 정보등을 추출하고 엘리베이터 내의 CCTV를 통해 인원 계수를 실시한다. 마지막으로 다수의 사용자에 대한 데이터는 각 사용자의 엘리베이터 사용 데이터(엘리베이터 호출, 출발 층, 목적층 등)을 수집함으로써 이루어진다.
* 이와 같이 크게 3가지의 데이터들에 대한 수집 및 종합적인 분석을 통해 새로운 엘리베이터 알고리즘을 개발한다면 ‘유의미한’ 엘리베이터 알고리즘이 도출될 것이라고 사료된다.
  1. **주요 상세기능 설명**

1. ID : smart\_elevator\_001\_001
2. 정의: 기본적인 엘리베이터의 알고리즘의 방식은 다음과 같다. 엘리베이터의 작동 방향(상, 하)과 동일한 각 층의 엘리베이터의 호출에 반응하여 같은 방향 내(지나온 층은 제외)의 각 층의 호출자(사용자)들을 승차시켜 목적층에 도착하는 것이다. 하지만 이와 같은 알고리즘은 한 건물 내에 다수의 엘리베이터가 존재한다면 어떤 엘리베이터가 특정 사용자에게 제일 빠르게 반응할 것인지 확인할 수 없다는 점에서 효율적이지 못하다. 따라서 본 정의서에서 제시하는 Smart Elevator Algorithm의 작동방식은 다음과 같다. 다수의 층, 다수의 엘리베이터, 다수의 사용자에 대한 실시간 데이터 수집 및 분석을 통해 실시간으로 특정 사용자에게 목적층에 가장 빠르게 도착할 수 있는 승강기를 할당하여 주는 것이다.

* **[주요상세기능] #1**
  + ID : smart\_eleveator\_001\_001
  + 정의 : 주요상세기능 #1은 다수의 엘리베이터 내/외부의 정보를 호출한다.
  + 상세 설명 : 승강기 제어반 데이터베이스로 부터 유저 호출이 일어난 순간의 엘리베이터들의 내부 정보/ 외부 정보를 호출해와야 한다.
* **[주요상세기능] #2**
  + ID : smart\_eleveator\_001\_002
  + 정의 : 주요상세기능 #2은 승강기 실시간 정보들을 정제한다.
  + 상세 설명: 주요상세기능 #1을 통해 모아진 내부정보로부터 각 승강기내 인원 계수 및 현재 승강기 위치, 이동방향 정보를 얻고 외부정보로부터 각 승강기 외부의 인원 계수를 실행한다.
* **[주요상세기능] #3**
  + ID : smart\_eleveator\_001\_003
  + 정의 : 주요상세기능 #3은 사용자에게 가장 효율적인 승강기를 할당한다.
  + 상세 설명 : 주요상세기능 1,2에서 언급한 데이터들의 종합적인 분석을 통해 실시간으로 특정 사용자에게 목적층에 가장 빠르게 도착할 수 있는 승강기를 할당하여 준다.
* **[주요상세기능] #4**
  + ID : smart\_elevator\_002\_001
  + 정의 : 주요상세기능 #4는 Smart Elevator Algorithm을 통해 사용자에게 할당된 엘리베이터에 대한 예상 대기 시간, 예상 이동 시간에 대한 비교 정보를 출력한다.
  + 상세 설명 : 실제 엘리베이터 사용에 있어서는 엘리베이터를 할당하고 호출하는 것이 결과이기 때문에 출력해줄 결과가 따로 존재하지 않는다. 따라서 개발 과정 내에서의 결과 출력을 위해 다음과 같은 시나리오를 통해 결과를 출력하여 주는 것으로 대체한다. 기존 알고리즘을 베이스로 작성된 가상의 엘리베이터 시뮬레이터와 본 정의서에서 제시하는 Smart Elevator Algorithm 베이스로 작성된 엘리베이터 시뮬레이터에 대해 같은 조건을 대입하고 실제 사용자의 대기시간, 실제 도착시간을 측정하여 그 차이를 보여준다.
* 요구기능 동작 절차

1. 사용자의 원격 호출(출발층, 목적층 입력)
2. 입력된 출발층, 목적층에 따라 Smart Elevator Algorithm 기반의 엘리베이터 할당

2.1. 데이터 호출

2.2. 데이터 정제

2.3. 알고리즘 적용 및 결과 반환(엘리베이터 할당)

1. 할당된 엘리베이터에 대한 시뮬레이터 작동 및 실제 효율성 출력

[단계별 상세 내용]단계의 상세내용은 다음과 같다.

단계 1: 임의의 출발층과 목적층을 설정하고 엘리베이터(Smart Elevator Algorithm)을 호출한다.

단계 2: 단계 1에서 호출될 때 설정하였던 출발층과 목적층을 Smart Elevator Algorithm에 입력으로 넘기고 Algorithm은 실시간 층 데이터, 엘리베이터 내부 데이터, 사용자 데이터들과의 종합적 분석을 실행한다. 실행에 따른 최적의 엘리베이터를 사용자에게 할당한다.

단계 2.1: 단계 1에서 수집한 사용자 입력 데이터와 같은 건물에 존재하는 엘리베이터 내부의 데이터, 그리고 엘리베이터 외부의 CCTV 데이터를 서버로 수집한다.

단계 2.2: 엘리베이터 내, 외부의 인원수,이동 방향, 현재 층수에 대한 데이터를 추출하고, 인원 계수가 필요한 CCTV 데이터들은 인원 계수를 실행한다.

단계 2.3: 수집, 정제된 데이터들을 기반으로 하여 Smart Elevator Algorithm을 적용하고 실행하여 최적의 엘리베이터를 사용자에게 할당한다.

단계 3: 단계 2에서 할당된 엘리베이터에 대한 시뮬레이팅 결과와 기존 알고리즘 사용을 통한 실제 엘리베이터 호출 시뮬레이팅 결과의 차이(실제 대기시간, 실제 이동시간)를 출력한다.

* 1. **동작 규칙**

1. 동작 규칙 1 : 사용자는 반드시 출발층과 목적층을 입력해야 한다. 두 입력 중 하나라도 공백이라면 호출이 불가능하다.

동작 규칙 1 설명 : 출발층과 목적층이 설정되지 않는다면 알고리즘의 작동은 이루어질 수 없다. 따라서 사용자는 두 가지 입력 모두 알고리즘의 입력으로 넘겨주어야 한다.

1. 동작 규칙 2 : 분석에 사용되는 각 데이터는 반드시 real time 데이터이어야 한다.

동작 규칙 2 설명 : 분석에 사용되는 각 데이터는 반드시 실시간 데이터이어야 한다. 실시간 데이터가 아닌 데이터 분석은 실시간 사용자에 대해 실질적으로 효율적인 엘리베이터 할당을 결정할 수 없다.

1. 동작 규칙 3: 인원계수는 사용자의 호출하는 순간의 프레임을 사용해야한다.

동작 규칙 3 설명 : 정확한 기준 없이 계속적인 비디오 데이터에 대한 인원계수는 많은 컴퓨팅 파워와 시간이 소모된다. 또한 인원 계수 종료 시점의 모호함을 없애기 위해 호출하는 순간 외부에 대기하는 인원의 계수만을 사용한다.

1. 동작 규칙 4 : 기존 알고리즘과 개선된 알고리즘의 실제 효율성 차이를 출력해야 한다.

동작 규칙 4 설명 : 제시한 알고리즘이 기존 알고리즘보다 효율성 측면에서 우수성을 보여야 한다. 실제 시뮬레이팅 결과(실제 대기시간, 실제 이동시간)에 대한 상세한 결과를 출력하여 효율성을 입증해야 한다.

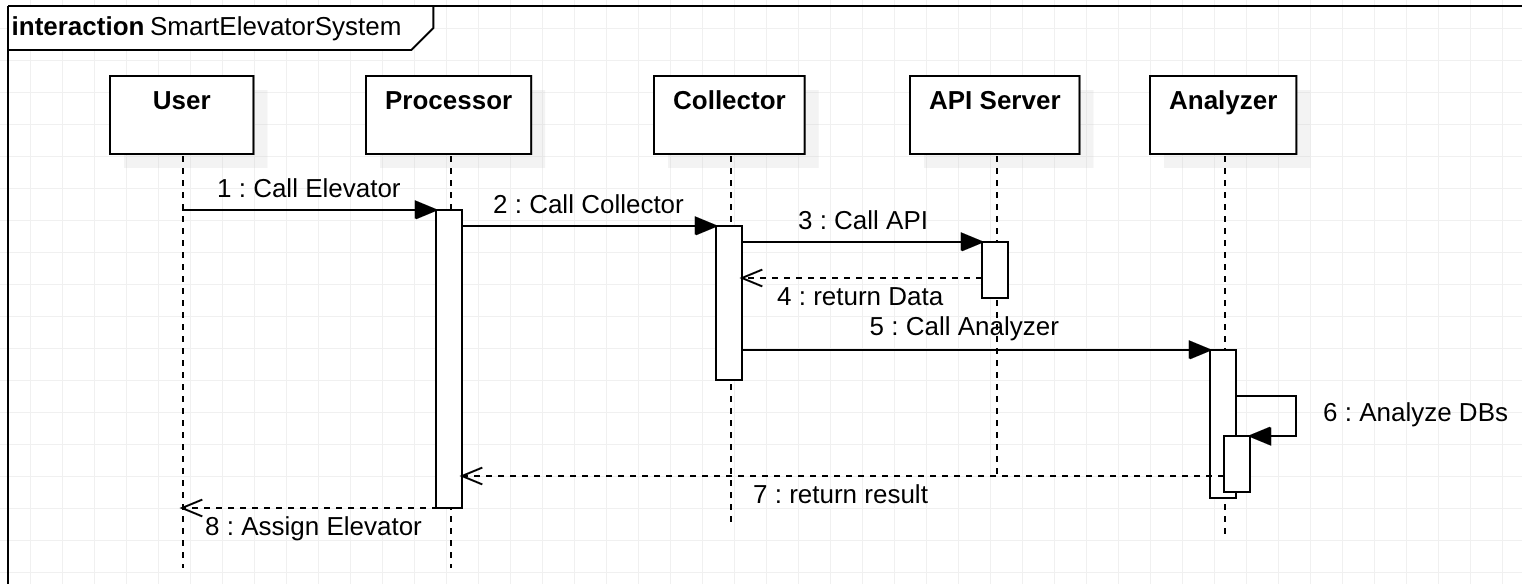
* 1. **가정(Assumptions)**

Smart Elevator System은 다음과 같은 내용의 가정하에서 실행된다.

* 사용자의 출발층,입력층 입력이 존재해야 한다.
* 사용자의 입력이 없다면 시스템은 진행되지 않는다
* 엘리베이터 외부에 CCTV가 존재해야 한다.
* 엘리베이터 내부에 CCTV 및 상태 정보가 존재해야 한다.
* 사용자의 입력이 없다면 시스템은 진행되지 않는다.
* 엘리베이터 외부 대기열에 대기하고 있는 인원이 아닌 사람이 없어야 한다.
* 각 데이터는 실시간 기반으로 수집된 데이터이어야 한다.
* 시뮬레이팅의 결과를 반드시 출력하여야 한다.

**주요 상세기능 #1**

* 1. **기능 설계**

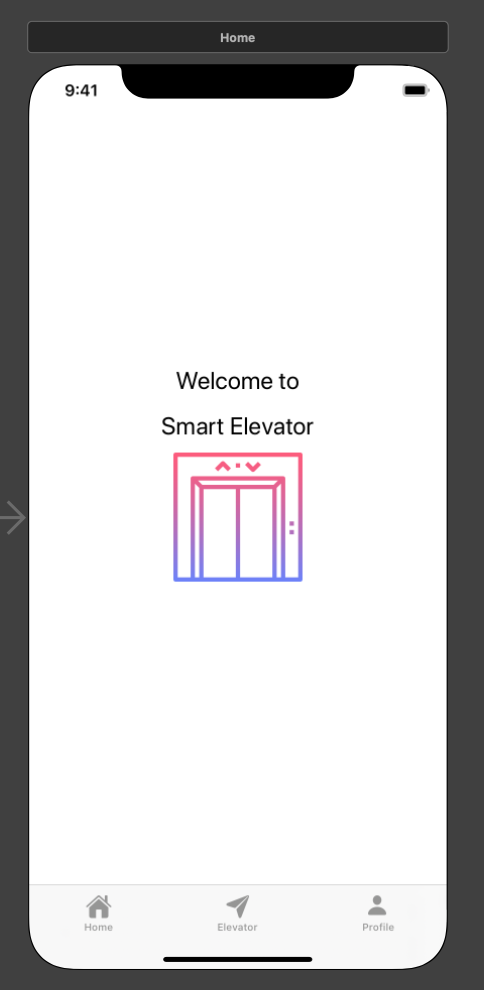


* Sequence diagram 설명

Smart Elevator System의 프로세스에 대한 과정을 다이어그램을 통해 가시적으로 작성한 것이다.

* 1. **요소 설계**
     1. Class
* Class User
  + 이름: User
  + 속성: client, 입력층과 목적층을 입력하여 엘리베이터 호출
  + 메소드: callElevator()
* class Processor
  + 이름: Processor
  + 속성: 실시간 데이터에 대한 수집을 위해 Collector를 호출, 최적의 엘리베이터 할당
  + 메소드: callCollector(), assignElevator()
* class Collector
  + 이름: Collector
  + 속성: 각 층, 각 엘리베이터, 각 사용자에 대한 데이터 베이스에 접근하여 실시간 데이터를 수집한다.
  + 메소드: getFloorData(), getElevatorData(), getUserData()
* class API Server
  + 이름: API Server
  + 속성: 필요한 실시간 데이터를 정제해 Collector에게 제공
  + 메소드: returnFloorData(), returnElevatorData(), returnUserData()
* class Analyzer
  + 이름 : Analyzer
  + 속성 : 실시간 데이터를 기반으로 최적의 엘리베이터 탐색
  + 메소드 : searchOpt(), returnOpt()
    1. 저장소
* 엘리베이터 내부 DB
  + 각 엘리베이터 내의 실시간 이용자에 대한 DB
* 엘리베이터 전면부 DB
  + 각 층의 엘리베이터 전면부의 실시간 대기열
* 사용자 DB
  + 다수의 실시간 사용자에 대한 DB

1. **주요 상세기능 #2** 
   1. **UI 설계**
      1. 스플래시 스크린 UI



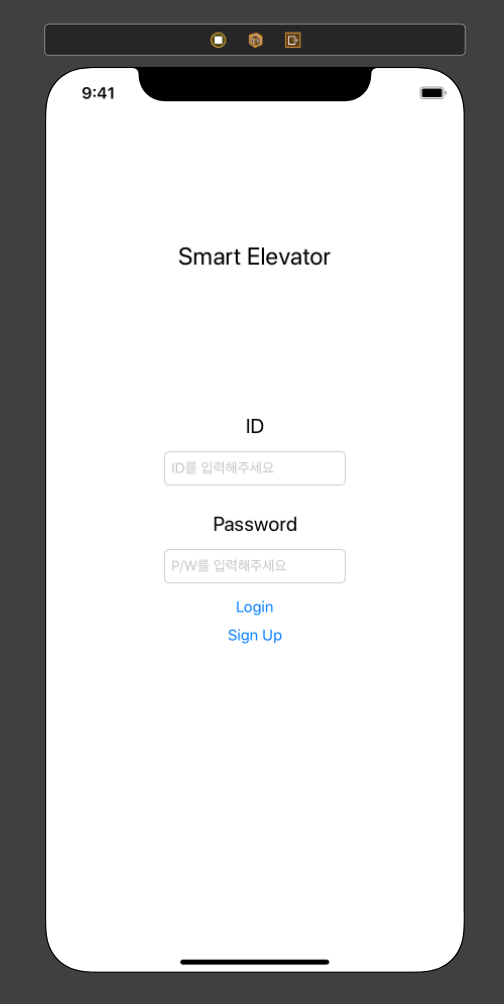
* 스플래시 UI 설명

- 어플리케이션 첫 실행 시 화면

* 스플래시 UI 제약사항

1. 정상 로딩이 완료되면 자동으로 메인 화면으로 이동해야한다.
2. 완료 후 로그인 화면으로 이동된다.
   * 1. 로그인 UI

* 입력 UI 구성

****

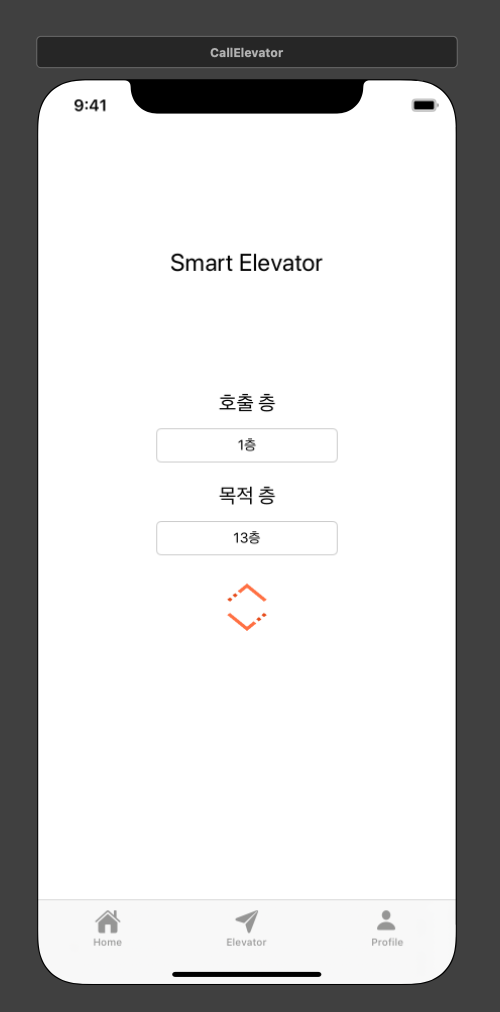
* 로그인 UI 설명

- 사용자가 자신의 아이디와 비밀번호를 입력하여 로그인 할 수 있는 페이지이다.

* 입력 값
  + ID: 유저 개인 ID
  + Password: 유저 개인 password
* 입력 UI 제약사항

1. ID는 대소문자를 구분한다
2. Password는 ‘\*’로 출력하여 내용을 볼 수 없어야한다.
3. ID와 Password가 모두 입력이 되어야 정상적인 Login이 되어야한다.
4. ID와 Password가 DB에 저장되어야 하고 서로 매칭이 되어 있어야 한다.
5. 올바른 ID, Password 쌍 에만 정상적 로그인이 이루어져야 한다.
6. 로그인이 안된 유저한하여 메인 화면을 대처하여 한다.
   * 1. 입력 UI

* 입력 UI 구성

****

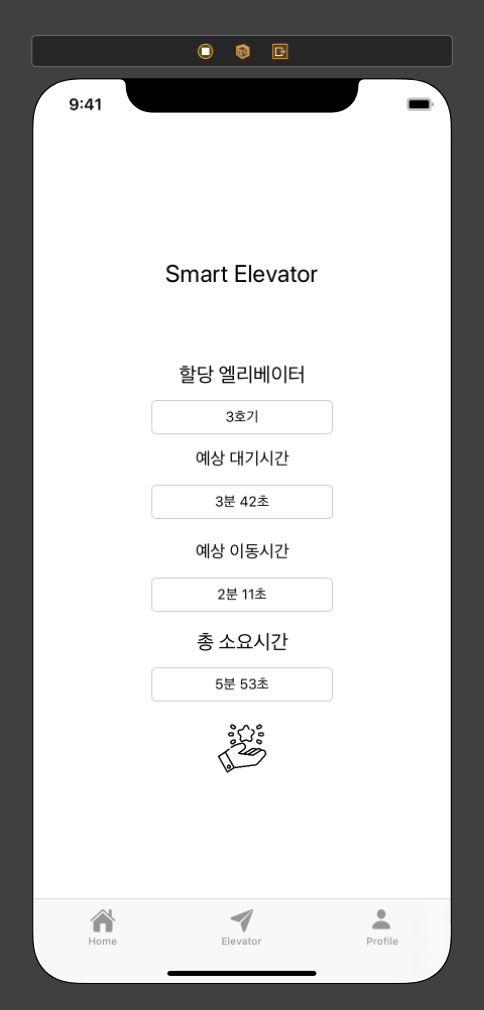
* 입력 UI 설명

- 사용자가 자신의 출발 층 및 목적 층을 입력하고 엘리베이터를 호출하는 화면이다.

* 입력 값
  + 출발 층 : 1
  + 목적 층 : 13
* 입력 UI 제약사항

1. 출발 층과 목적 층은 달라야 한다.
2. 엘리베이터의 구동 범위 이내의 층을 입력해야 한다.
3. 엘리베이터 호출 시에는 출발 층과 목적 층 모두 입력하여야 한다.
4. 실행 후 결과 UI로 이동되어야 한다.
   * 1. 결과 UI

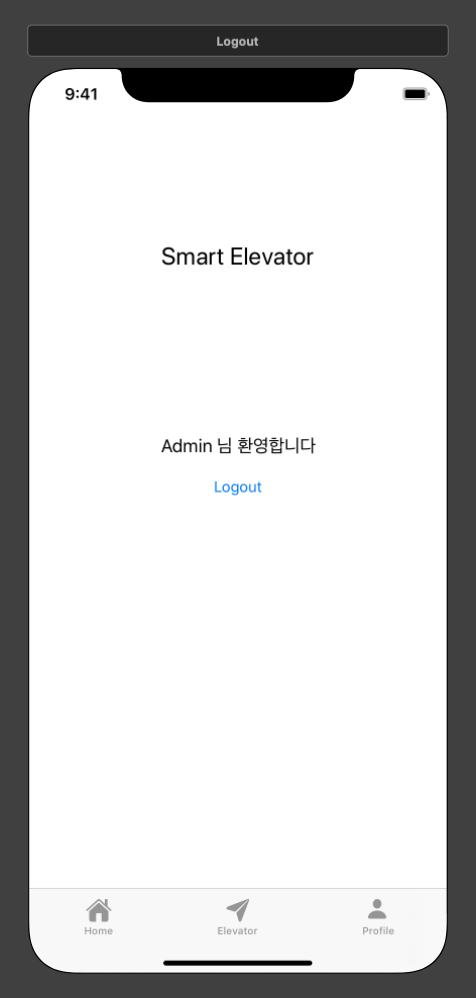
* 결과 UI 구성



* 결과 UI 설명
  + 할당된 엘리베이터의 호기 번호를 출력해주고, Smart Elevator System 기반으로 할당된 엘리베이터의 예상 대기 시간 및 예상 이동 시간을 출력해준다. 기존 알고리즘과의 비교를 위해 기존 알고리즘의 예상 대기 시간 및 예상 이동 시간 또한 출력하여 비교 출력한다.
* 결과 UI 제약사항

1. 총 소요시간은 예상 대기시간과 예상 이동시간의 합이어야 한다.
2. 다양한 엘리베이터 중 가장 적은 소요시간을 가지는 엘리베이터가 할당 되어야 한다.
3. 입력 UI로 다시 이동할 수 있어야한다.
4. 입력 UI를 거치지 않고 접근 할 수 없어야 한다.
   * 1. 메인 UI

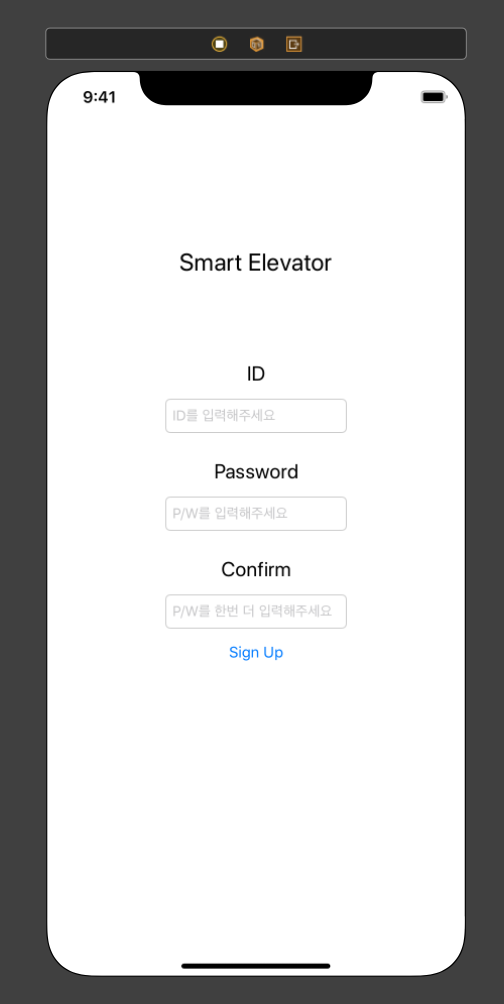
* 메인 UI 구성



* 메인 UI 설명
  + 로그인 된 회원의 이름을 보여주고 어플리케이션의 기본이 되는 화면이다.
* 메인 UI 제약사항

1. 로그인이 된 사용자에 한하여 보여줘야 한다.
   1. 로그인이 안된 유저는 로그인 화면을 보여줘야 한다.
2. 로그인 된 회원의 이름이 출력되어야 한다.
3. 입력 UI로 이동이 가능하도록 Tab 바가 필요하다.
   * 1. 회원가입 UI

* 회원가입 UI 구성



* 회원가입 UI 설명
  + 회원가입을 위한 화면이다.
* 입력 값
* ID : 새로 가입하는 아이디
* Password: 새로 가입하는 아이디에 대한 Password
* Confirm: password가 의도된 대로 확인하기 위해 재입력
* 회원가입 UI 제약사항

1. 중복된 ID로 회원가입은 불가해야 한다.
2. Password는 최소 12자 이상, 알파벳, 숫자, 특수문자의 조합이어야 한다.
3. Password와 Confirm은 ‘\*’로 출력 되어야 한다.
4. Password와 Confirm은 같은 값을 가져야 한다.
5. Sign up 후 가입한 계정으로 자동 로그인되어 메인 화면으로 이동해야 한다.