**Capstone Design Project Report**

**Project Title : RPA를 활용한 IoT 디바이스 관리 및 제어 서비스**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Lecture** **Class** | **컴퓨터공학종합설계01\_03** | | | | |
| **Term** | **2020.03.16. ~ 2020.11.27** | | | | |
| **Supervisor** | **Class Advisor** | | | **손윤식** | |
| **Industry Mentor** | **Industry** | | **주식회사 조인트리** | |
| **Mentor** | | **권창원** | |
| **Team Member Students** | **Student ID No.** | | **Name** | | **E-mail** |
| **2015112155** | | **김우주** | | woojoo1602@naver.com |
| **2015112154** | | **김석윤** | | blacswansong@naver.com |
| **2015112135** | | **윤종연** | | 0417yjy@naver.com |
| **2016113517** | | **조주아** | | my\_sg1@naver.com |

<Summary in English>

|  |  |  |  |
| --- | --- | --- | --- |
| Project Goal and Objectives | This research plans to develop the IoT(Inter of Things) devices control and management service using the RPA(Robotic Process Automation). RPA is a form of metamorphic software robot or business process automation. RPA is an automation technology by software doing human’s reiterative works based on rules, so It could reduce work time and personnel expenses and raise the manager’s job satisfaction by letting him/her concentrate on high-valued works. Therefore, this research’s objective is improve existing IoT management system to be more efficient and easy by merging RPA and IoT device control and management service and developing algorithms and service system that automates simple and regular jobs. | | |
| Project Contents | This research consists of three sub-tasks as below.  【1st task】The development of algorithms to understand devices’ status by accumulated log data using RNN algorithm  (1) To develop an algorithm that classifies the device’s status into Fine, Bad, Cautious, Serious using device’s information such as multiple device’s values, arrival log time, etc.  【2nd task】The development of an RPA robot that manages IoT devices automatically  (1) To develop an RPA script that watches devices’ status by keep reconnecting monitoring page  (2) To develop an algorithm recognizing the upcoming status value and doing proper operations  【3rd task】The development of an RPA robot that generates report files by given report templates  (1) To develop an RPA script that loads saved template and inserts each value into the desired location  (2) To develop an algorithm verifying specified values are properly saved  (3) To develop a module that could display generated reports to the manager and save them into any location. | | |
| Anticipated Results and Impacts on Industry | This research develops a system that could manage the facility using an RPA solution to manage IoT equipment that is the basis of the 4th industrial revolution efficiently, and algorithms that distinguish devices’ status using devices’ log data. The development of this kind of system achieves facility manager’s effective IoT equipment management, and decreases the manager’s amount of simple works like reiterative report making and required manpower. Also, it watches IoT equipment automatically so the responsibility and transparency of the system are enhanced. It will do an important role in being the base of the expansion of the management technology of IoT equipment and devices. | | |
| Keywords | RPA | IoT | Management System |
| Document Automation |  |  |
|  |  |  |

<국문 요약>

|  |  |  |  |
| --- | --- | --- | --- |
| 과제의 목적과 목표 | 이 연구에서는 RPA(Robotic Process Automation)를 활용한 IoT 디바이스 제어 및 관리 서비스를 개발한다. RPA는 은유적인 소프트웨어 로봇 혹은 비즈니스 프로세스 자동화의 한 형태이다. RPA는 사람이 하는 규칙 기반의 반복적인 업무를 소프트웨어를 통해 자동화하는 기술로써 단순 작업에 대한 시간 및 인건비를 줄이고 담당자는 보다 부가가치가 높은 업무에 집중하여 업무 만족도를 높일 수 있다. 따라서 이 연구에서는 RPA를 IoT 디바이스 제어 및 관리 서비스에 접목시켜 단순하고 규칙적인 업무를 자동화하는 알고리즘과 서비스 시스템을 개발하여, 기존의 IoT 디바이스 관리 시스템을 보다 효율적이고 용이하게 개선하는 것을 목표로 한다. | | |
| 과제의  내용 | 이 프로젝트는 다음과 같이 n개의 세부과제로 구성된다.  【1세부과제】축적되는 로그 데이터에서 3-시그마 규칙을 통하여 디바이스의 상태를 파악하기 위한 알고리즘 개발  (1) 다수의 로그 데이터의 값, 로그가 쌓인 시각 등의 디바이스 정보를 받아 디바이스의 상태를 작동, 고장, 일시적 오류, 경고 네 가지 값으로의 분류 알고리즘 개발  【2세부과제】 IoT 디바이스 자동 관리 RPA 로봇 개발  (1) 수시로 관리 페이지를 조회하여 디바이스의 상태를 감시하는 RPA 스크립트 개발  (2) 받은 데이터로부터 상태 값을 인식하고 불량 디바이스에 대한 적절한 동작을 취하는 알고리즘 개발  【3세부과제】주어진 양식에 맞추어 보고서 파일을 생성하는 RPA 로봇 개발  (1) 미리 저장된 양식을 불러와 원하는 위치에 원하는 값을 입력하는 RPA 스크립트 개발  (2) 지정된 값들이 전부 저장되고 이를 검증하는 알고리즘 개발  (3) 작성한 문서를 사용자에게 보여주고 임의의 공간에 저장할 수 있는 모듈 개발 | | |
| 예상되는 결과와 파급효과 | 이 연구에서는 제4차 산업 혁명의 기본이라고 할 수 있는 IoT 장비를 보다 효율적으로 관리하기 위하여 RPA 솔루션을 활용하여 시설을 관리할 수 있는 시스템과, 디바이스의 로그 데이터를 보고 디바이스의 상태를 판별할 수 있는 알고리즘을 개발한다. 이러한 시스템의 개발은 시설 관리 담당자의 효과적인 IoT 장비 관리를 달성하고, 주기적으로 문서를 작성해야 하는 관리 담당자의 단순 작업량을 감소시키며, 요구 인력 감소를 달성할 수 있다. 또한 자동으로 IoT 장비를 감시하므로 시스템 설비의 신뢰성과 투명성이 향상된다. 추후의 IoT 장비 및 디바이스의 관리 기술을 확장하는 데 기초가 되는 중요한 역할을 할 것이다. | | |
| 중요단어 | RPA | IoT | 관리 시스템 |
| 문서 자동화 |  |  |
|  |  |  |

**1. Project Goals and Objectives (과제의 목표와 목적)**

**1.1 Project Goal (과제의 목적)**

오늘날 4차 산업혁명을 통한 IT계의 혁신이 확산되면서, 기존에 정보시스템을 이용하지 않았던 영역에서도 다양한 단계에서 정보시스템을 활용하고 있다. 기술이 사람을 대체할 수 있도록 변화되고 있는 환경 속에서, AI를 기반으로 하는 RPA(Robotic Process Automation)가 활성화되고 있다. RPA는 업무상 흔히 나타나는 데이터의 반복적인 입력, 복사와 붙여넣기와 같은 규칙적인 업무를 자동화할 수 있게 한다.

교내 수많은 IoT 디바이스들은 현재 사용자가 직접 데이터를 열람하거나 인터페이스를 통해 접근해 긴급 상황을 포함한 여러 가지 문제들을 인식하고 제어하였다. 이러한 업무환경에 RPA를 도입하여 긴급 상황 알림 및 문서작성 업무 등을 자동화한다면, 사용자는 상황 별 문제를 더 유연하고 폭 넓게 인식할 수 있고 보다 안정적으로 디바이스들을 관리할 수 있으며, 소중한 인력과 시간을 절약할 수 있을 것이다.

**1.2 Project Objectives (과제의 목표)**

교내 IoT 디바이스들을 관리하는 환경에 RPA를 도입하여 사용자가 디바이스들의 상태를 보다 편리하게 관찰하고 관리할 수 있도록 다음과 같은 목표를 세울 수 있다.

첫째, IoT 디바이스의 로그데이터를 딥러닝 기술을 통해 분석한다.

둘째, 분석한 결과를 토대로 IoT디바이스의 상태가 긴급하다고 판단될 경우 SMS를 통해 사용자에게 이를 알린다.

셋째, 웹 어플리케이션 환경에서 사용자가 로그 및 상태데이터를 확인할 수 있도록 한다.

넷째, RPA를 통해 IoT디바이스의 상태에 따라 자동적으로 문서를 작성할 수 있도록 한다.

**2. Background of the Project (과제의 필요성)**

**2.1 Motivations (필요성)**

**Technical Motivations**

기존의 IoT 디바이스 제어 및 관리 서비스는, 데이터를 축적하기만 할 뿐, 실시간 모니터링, 알림 및 보고에 대한 기능이 존재하지 않았다. 엄밀히 말해, 사람이 조치를 취하는 등의 행위를 배제한다면, 제어 및 관리의 기능은 없고 단순 데이터 축적 서비스라고 할 수 있다.

그러나 RPA를 활용한 IoT디바이스 제어 및 관리 서비스는 기술적 의미에 있어서 다음과 같은 필요성을 갖는다. 사용자가 실시간으로 모니터링 할 필요가 없어지고, 모니터링 한 결과에 대해 알림을 보내거나 보고를 하는 등의 조치를 취하기 때문에, ‘자동화된 기술’ 이라는 점이 본 서비스의 기술적 동기이다.

**Economical Motivations**

경제적 관점에서 RPA를 활용한 IoT디바이스 제어 및 관리 서비스는 다음과 같은 동기를 갖는다.

1. 비용의 감소

기존의 IoT 디바이스 제어 및 관리를 위해선, 사용자가 계속 모니터링 하거나, 문제 발생 시, 직접 디바이스를 제어하는 과정이 필요했다. 그러나 실시간 모니터링을 프로그램으로 자동화하고, 디바이스 제어 또한 자동화 함으로서, 제어 및 관리에 대한 인력을 줄일 수 있다.

2. 효율적 IoT 디바이스 운용 및 생산성

모든 장비가 그러하 듯이, IoT 디바이스 또한 시간이 지나면서 노화되고 처음의 제 기능을 못하기 마련이다. RPA를 활용해 IoT디바이스 제어 및 관리하면, 문제가 생긴 디바이스에 대한 알림을 즉각적으로 받을 수 있기 때문에, 후에 더 커질 수 있는 문제를 더 일찍 해결할 수 있다. 이로 인해, 효율적인 IoT 디바이스 운용이 가능하다.

**Social Motivations**

RPA를 활용한 IoT 디바이스 제어 및 관리 서비스는 다음과 같은 사회적 동기를 갖는다.

1. 사용자의 작업 환경 개선

RPA를 활용한 IoT 디바이스 제어 및 관리 서비스는 사용자가 IoT 디바이스를 모니터링을 하는 것이 아니라 프로그램이 실시간으로 모니터링 한다. 따라서 실시간으로 모니터링 해야 하는 디바이스의 경우, 자동화된 실시간 모니터링을 통한 알림 서비스를 통해, 필요한 경우에만 사용자가 조치를 취할 수 있게 함으로써 작업 환경이 개선된다.

2. 문제 조기 조치

가령 화재 사고가 날 수 있는 수치가 나타났다고 하자. 사람이 디바이스를 관리하는 경우, 해당 수치를 놓치거나, 인지했다 하더라도 대응하는 데 시간이 걸릴 수 있다. 그러나 디바이스 제어 및 관리를 프로그램을 통해 자동화하면, 실수로 인해 해당 수치를 놓칠 일이 없어지고, 대응 또한 더 빠르게 할 수 있다.

**2.2 Significances (중요성)**

**Market**

4차 산업혁명의 핵심이 되는 기술은 다양하다. 빅데이터, 인공지능 등이 그러하다. 그중 하나에 사물인터넷(IoT)이 있다. IoT는 다음과 같이 정의된다.

“실생활에 해당하는 오프라인의 모든 정보를 온라인으로 넘기는 O2O를 통해, 인터넷을 이용한 최적의 해법을 제시하고, 시행하게 하여 생산성을 최대한으로 올리는 기술이다.”

우리는 생산성을 올린다는 점에 중점을 둘 필요가 있다. 생산성을 올리기 위해, 자동화를 도입하기 전 기존의 방식은, 사용자가 직접 온라인으로 넘겨진 정보를 보고 조치를 위하여 생산성을 올리는 방식이다. 그러나 ‘자동화’ 기술을 융합한다면, 이는 곧 생산성을 자동화하여 올리는 것으로 해석할 수 있다. 이와 같은 RPA를 활용한 IoT 디바이스 제어 및 관리 서비스의 중요성이 market에 영향을 미친다.

**Technique**

기존의 IoT 디바이스 관리 및 제어를 자동화하는 데 있어, RPA 솔루션으로 구현한다는 점이 기술적인 면의 중요성으로 영향을 미친다. RPA는 단순 반복업무에 적용하기 좋은 기술이므로, IoT 디바이스 모니터링과 제어에 적합한 특징을 갖고 있다.

**Society**

본 서비스는 IoT 센서 디바이스를 제어 및 관리한다. 센서는 온도, 습도 등 다양한 정보를 수집한다. 가령 화재가 발생하거나, 발생할 수 있는 문제가 생긴 경우, IoT 디바이스를 관리 및 제어를 자동화하여 문제를 초기에 해결하거나 문제가 더 심각해지기 전에 조치를 취할 수 있다. 이와 같이 발생한 또는 발생 가능한 문제/재해에 대해 사전에 빠르게 조치를 취할 수 있다는 점이 사회적 중요성으로 작용한다.

**3. Related Works (관련연구)**

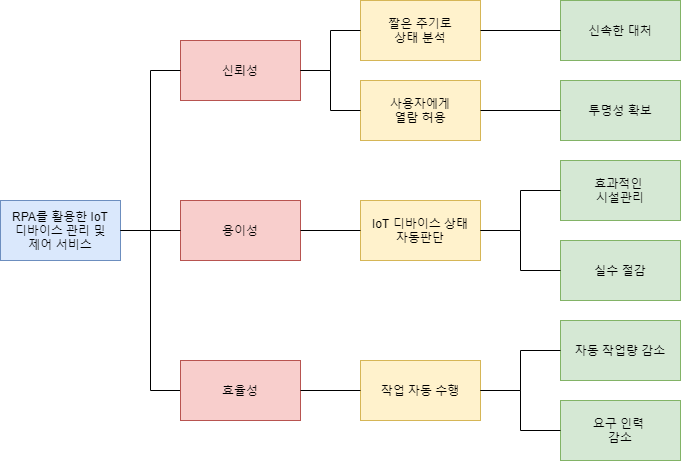
**3.1 Related Techniques**

관련기술 명 : Robotic process automation for supply chain management operation

1) Function : 관련 기술은 RPA와 IoT 기술을 사용하고, 특히나 RPA 기술을 이용해 시스템에 대한 전반적인 운영을 한다는 점이 유사하고, IoT 기술은 IoT 디바이스와의 통신을 연결하여 사용한다는 점이 유사하다. 그러나 관련기술은 IoT 기술을 서버와 통신하는 데에 사용하지만, RPA를 활용한 IoT 디바이스 제어 및 관리 서비스는 IoT 기기와 운영 시스템의 통신이 주가 된다는 점이 차이가 있고, 사용자 인터페이스와 문서 자동화 , 빅 데이터 분석기술은 포함되지 않는다는 점이 있다.

**4. Requirement Analysis (요구 분석)**

그림 1. Objective Tree



**4.2 Functional Requirement Analysis (요구 분석)**

**4.2.1 Function-Related Requirement (기능 관련 요구사항)**

본 프로젝트에서 요구 기능 중 첫 번째는 IoT 디바이스로부터 로그 데이터를 수신하는 기능이다. 이 기능은 건물 내에 존재하는 IoT 디바이스가 전송하는 로그 데이터를 수신하여 데이터베이스에 저장하는 기능이다. 위 기능은 네트워크 연결이 필요하며, IoT 디바이스가 데이터를 전송하는 서버와의 연결이 필요하다. 또한 로그 데이터를 저장할 데이터베이스를 구성할 DBMS가 필요하다. 두 번째 요구 기능은 IoT 상태를 분석하고 시각화 하는 기능이다. 이 기능은 수신한 IoT 디바이스 로그 데이터의 정보를 3-시그마 규칙을 통해 분석하여 IoT 디바이스의 상태정보를 예측하여 생성한다. 위 기능은 데이터베이스에 연결하여 로그데이터를 가져오고, 딥러닝 알고리즘을 통해 결과를 분석하는 프로그램과 분석 결과와 로그 데이터를 가지고 정보를 표시해주는 웹 어플리케이션이 필요하며, 상시 네트워크 연결이 필요하다. 세 번째 기능은 문서 작성 자동화 기능이다. 이 기능은 두 번째 기능에서 보고서 작성 버튼을 누르거나, 분석된 IoT 디바이스의 상태가 비정상일 경우에 RPA 솔루션을 통하여 현재시각을 기준으로 마지막으로 수신된 로그 데이터와 상태를 엑셀파일 형태의 보고서 양식에 자동으로 입력한다. 그 후에 정기 보고서는 자동으로 관리자 PC에 저장하고, 긴급한 상황의 경우에는 해당 내용을 관리자에게 SMS 전송 기능을 통해 전송하고, 긴급 보고서를 작성하여 저장한다. 위 기능은 RPA 솔루션과 RPA 솔루션이 상시 가동되는 서버 PC가 필요하며, Excel 프로그램과 SMS 자동 전송 프로그램이 필요하고, 상시 네트워크에 연결되어 있어야 한다. 또한 첫 번째 기능과 두 번째 기능은 웹 서버 환경에서 동작하며, 상시 가동되는 서버 PC가 필요하다.

**4.2.2 Data-Related Requirement (자료 관련 요구사항)**

본 프로젝트에서 데이터의 입력은 IoT 디바이스에서 전송되는 로그 데이터이고, 반정형 데이터이다. 이 데이터를 각 건물의 모든 IoT 디바이스에서 짧은 주기로 수집을 하므로 매우 많은 양의 데이터가 수집된다. 이 데이터는 10일동안 보관되며 10일 후에는 폐기된다. 데이터의 출력은 IoT 디바이스의 상태 데이터이다. 실수 형태인 이 데이터는 약 10분 주기로 생성되며, IoT 디바이스의 양이 많으므로 마찬가지로 매우 많은 양의 데이터가 생성된다. 이 데이터는 로그 데이터와 마찬가지로 10일 동안 보관되며 10일 후에는 폐기된다.

**4.2.3 Interface-Related Requirement (인터페이스 관련 요구사항)**

다른 시스템에서 공급되는 데이터는 IoT 디바이스에서 전송되는 로그데이터로, 반정형 데이터를 공급한다. 이 외에는 특별한 인터페이스는 존재하지 않는다.

**4.2.4 User-Related Requirement (사용자 관련 요구사항)**

본 프로젝트의 잠재적 사용자는 IoT 디바이스 관리 시스템 관리자이다. 이 사용자는 IoT 디바이스 모니터링관련 경험이 있으며, 각 IoT 디바이스별로 어떻게 관리를 해야 하는지 숙지하고 있는 상태여야 한다.

**4.3 Realistic Constrains (현실적 제한조건)**

**4.3.1 Constraints for Available Resources (가용 자원에 대한 제한조건)**

본 프로젝트에서 가용 자원에 대한 제한조건은 다음과 같다.

1. RPA 솔루션에 대한 숙련도

RPA 솔루션에서 필요한 요구 기능을 구현하기 위해서는 RPA 솔루션이 제공하는 기능들 중 데이터베이스 연동, 스크립트 작성을 통한 필요기능 구현과 기본 활용 기능 등을 능숙하게 활용할 줄 알아야한다.

1. 상시 가동 가능한 PC

RPA 솔루션은 Windows 운영체제에서 동작하고, 프로젝트의 핵심 기능들은 WAS 환경으로 제공된다. IoT 디바이스 관리 및 제어는 상시 운영되어야 하기 때문에 위 기능들을 상시 가동이 가능한 환경을 구성하여야 하는데, 현실적으로 서버용 PC를 제작하기엔 개발비용이 많이 들기 때문에 Amazon Web Service같은 클라우드 서비스를 이용하여 환경을 구성하여야 한다.

1. IoT 디바이스 모니터링

교내의 모든 IoT 디바이스를 모니터링하기에는 현실적으로 데이터의 양이 매우 크기 때문에 IoT 디바이스 테스트베드를 사용하여 프로젝트에 필요한 데이터를 획득하여야 한다.

**4.3.2 Constraints for Minimum Performance (최소 성능에 대한 제한조건)**

본 프로젝트에서 최소 성능에 대한 제한조건은 다음과 같다.

1. 데이터 분석 기능은 각 IoT 디바이스별로 전송되는 데이터를 최소 로그 전송 주기 내에 분석하여야 한다.
2. 데이터 분석 기능은 90%의 계산 정확도를 가져야한다.

3. 데이터 시각화 기능은 최소 로그 전송 주기 내로 시각화 해야 한다.

4. 관리자에게 긴급 SMS 전송하는 기능은 약 2초 내로 전송할 수 있어야 한다.

**4.3.3 Constraints for Security (보안에 대한 제한조건)**

본 프로젝트에서 보안에 대한 제한조건은 다음과 같다.

1. IoT 디바이스 로그 데이터는 10일 동안 보관 후 삭제한다.
2. 데이터 시각화 기능에서 층별 약도에는 건물에 존재하는 보안상 중요한 위치는 표시하지 않는다.
3. 문서 작성 자동화기능에서 생성한 보고서는 암호화된 디렉터리에 저장하여 관리자 외에는 열람하지 못하게 한다.

**4.3.4 Constraints for Compatibility (호환성에 대한 제한조건)**

1. 웹 애플리케이션은 파이썬 기반 Django 프레임워크를 사용하여 제작하였으며, Django 프레임워크는 데이터베이스가 변경되더라도 같은 인터페이스로 신속하게 백업, 복원 및 동작이 가능하다.

2. 모든 디바이스 값 데이터 테이블은 로그 데이터 테이블의 id를 프라이머리 키로 가지므로, 디바이스의 종류가 추가되더라도 기존 테이블의 수정 없이 새 디바이스 테이블을 추가하여 동작 시키도록 설계하였다.

3. 건물과 층수, 그리고 디바이스의 추가 및 수정을 용이하게 하기 위하여 이들의 자료구조는 트리 구조로 이루어지도록 설계하였다.

**4.3.5 Constraints for Ethics (윤리성에 대한 제한조건)**

1. 사용하는 RPA 솔루션은 직접 SYMATION 사에서 연구용으로 라이선스를 받아 개발자용 라이선스 4개, 그리고 서버용 라이선스 1개를 취득하여 사용하였다.

2. 본 연구에서 개발하는 시스템으로 인하여 IoT 시설 관리자의 과중한 업무를 덜어내도록 목표를 잡아 개발하였다.

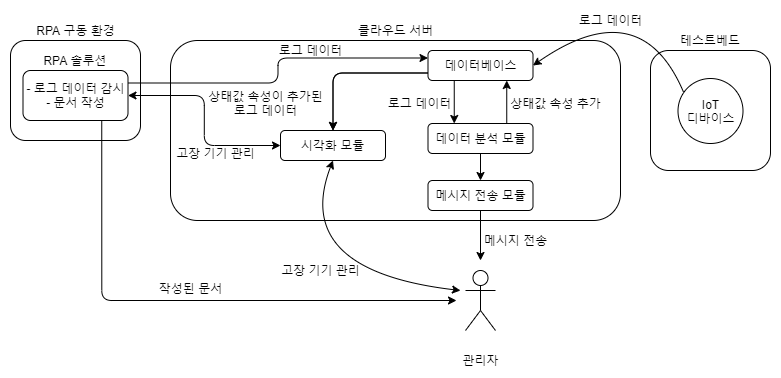
**5. Project Contents (과제 내용)**

**5.1 Design in Detail (상세 설계)**

**5.1.1 System Structure (시스템 구조)**

전체 시스템의 구조는 다음 그림과 같다.

그림 2 . 전체 시스템 구조도



시스템은 크게 3가지 컴포넌트로 나뉘며, 각각 RPA 구동 환경, 클라우드 서버, 그리고 테스트 베드이다. RPA 구동 환경 컴포넌트는, RPA 솔루션이 실행되는 환경을 뜻하며, 이 컴포넌트에서 RPA 봇이 운용되어 관리자가 해야 할 단순 반복 작업 분담한다. 예를 들면, 작성 명령이 떨어지거나, 혹은 주기적으로 자동으로 문서를 작성하여 완성된 파일을 관리자에게 넘겨줄 수 있고, 관리자와 같이 클라우드 서버에 접속하여 로그 데이터를 감시하다가 고장 현상이 발생했을 때 보고서 작성을 한다. 클라우드 서버 컴포넌트는 시스템 운용에 필요한 데이터베이스, 알고리즘, 웹 서버가 포함된 환경이며, 테스트베드의 IoT 디바이스의 로그 데이터가 클라우드 서버의 데이터베이스에 쌓이고, 쌓인 데이터를 관리자에게 보여주는 웹 페이지의 서버 역할과 쌓인 데이터를 분석하는 역할을 한다. 마지막으로 테스트베드는 실제 시설에서 운용 중인 IoT 디바이스를 논리적으로 표현한 것이며, 디바이스의 로그 데이터를 클라우드 서버의 데이터베이스로 넘겨준다.

**5.1.2 Module Specification (모듈 설계)**

RPA 솔루션 중 문서 작성 봇은, 문서 작성을 해야 할 명령이 들어왔을 때 작성하는 문서의 종류, 작성할 디바이스의 정보, 작성 시각, 사용할 양식 등을 입력으로 받아 입력한 양식에 맞추어 문서를 작성한다. 작성한 문서는 지정된 위치에 파일로 저장한다.

RPA 솔루션 중 로그 데이터 감시 봇은, 짧은 주기에 따라 클라우드 서버의 데이터베이스를 조회하고, 디바이스의 상태 불량이 나타나면 문서 작성 봇을 실행하고 이를 시스템에 표시한다.

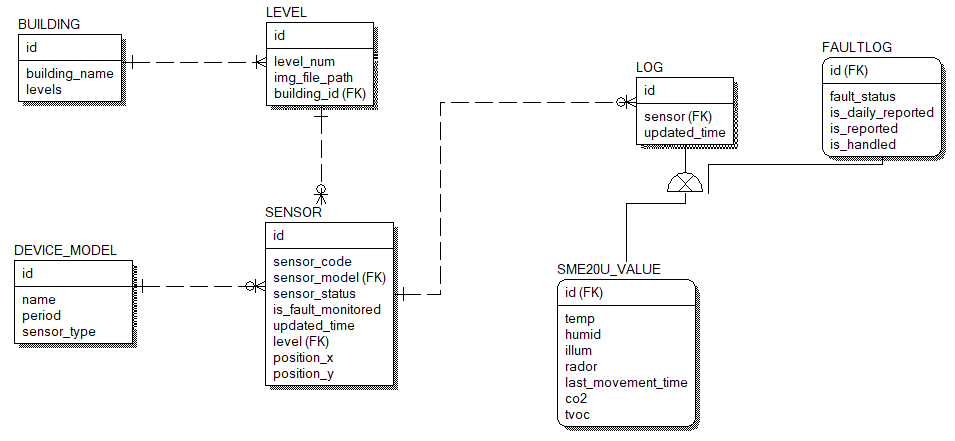
클라우드 서버 내의 데이터 분석 모듈은, 새로 들어온 데이터베이스 로그 값을 입력으로 받아, 3-시그마 통계 분석 알고리즘을 사용하여 현재 로그를 넘겨준 디바이스가 정상적으로 작동하는지, 비정상적으로 작동하는 지의 상태 값을 업데이트한다.

클라우드 서버 내의 시각화 모듈은, 데이터베이스 로그 값들과 각 디바이스의 상태들을 입력으로 받아 관리자가 원하는 위치의 로그와 디바이스들의 일람을 웹 브라우저를 통해 볼 수 있도록 한다. 또한 시설 내의 건물들이 여러 가지가 있고 건물마다 다른 층들이 존재하므로, 이를 트리 구조로 가져 디바이스를 볼 수 있도록 한다.

클라우드 서버 내의 SMS 전송 모듈은, 데이터베이스 분석 모듈이 디바이스의 상태 값을 업데이트 할 때 그 정도의 심각성에 따라 관리자에게 직접 SMS로 전송해야 할 때로 판단이 되면, 해당 디바이스의 정보와 상태 등을 입력으로 받아 관리자에게 SMS로 직접 전송한다.

클라우드 서버 내의 데이터베이스 스키마는 다음과 같다.

그림 3. 클라우드 서버 데이터베이스 스키마



**5.2 Implementation (구현)**

**5.2.1 Implementation Result (구현 결과)**

관리자가 접속하는 페이지의 모습은 그림 4와 같다.

그림 4. 관리자용 페이지 메인 화면

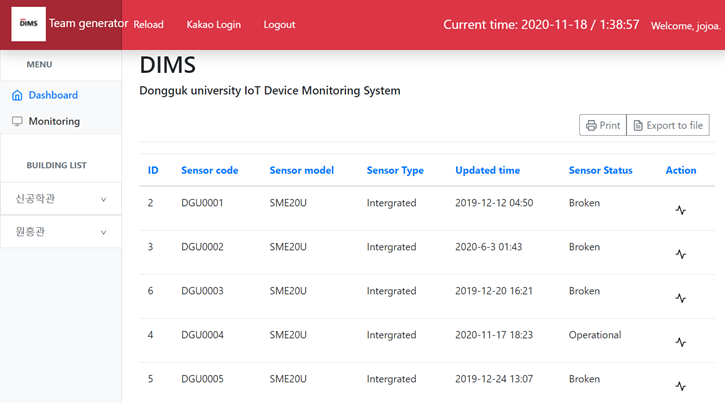
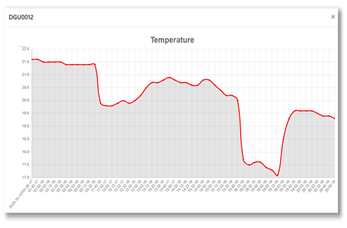


그림 5. 센서 그래프



페이지에 로그인 후 나오는 메인 화면이다. Monitoring Tab의 화면을 보여주고 있으며 모든 디바이스의 정보를 나타낸다. 각각의 column은 device code, device model, device Type, Updated Time, device Status, Action이다. device Status는 디바이스의 현재 상태를 나타낸다. Action column을 클릭하면 그림5와 같이 해당 디바이스의 기록을 그래프로 나타낸다.

또한 Export to file 버튼을 통해 IoT 디바이스의 현황을 csv 파일 형태로 다운로드 할 수 있다. 그림 6은 csv 파일 예시이다.

그림 6. device\_code.csv

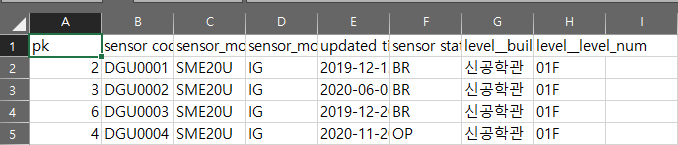
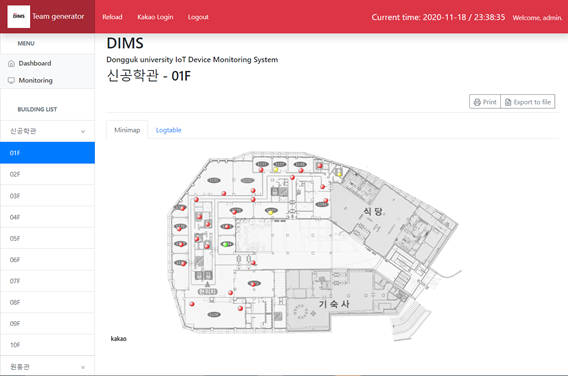


그림 7. 건물 목록 중 특정 건물 클릭 시 화면



확인하고자 하는 건물을 클릭하고 층을 클릭하면 Minimap 탭이 나타나고 이는 그림7과 같다. 해당 층의 도면과 존재하는 디바이스의 위치가 나타난다. 또한 디바이스는 초록색, 노란색, 빨간색 세가지 색으로 구분되고, 초록색은 Active(작동 중), 노란색은 Warning(경고), 빨간색은 Down(고장)을 의미한다.

그림 8. Logtable

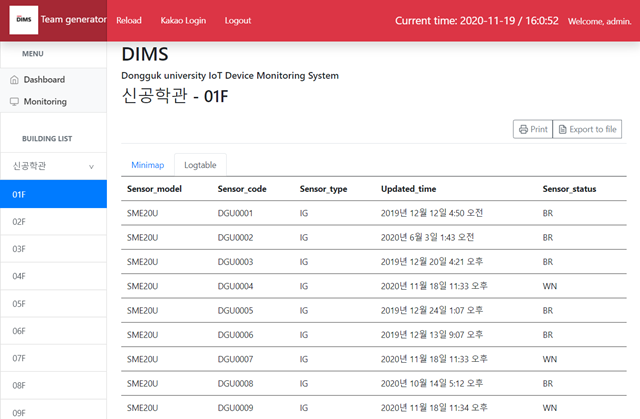


그림 8과 같이 Logtable tab을 클릭하면 해당 건물의 층에 위치하는 디바이스들의 상태를 표 형태로 열람할 수 있다.

그림 9. Monitoring Tab

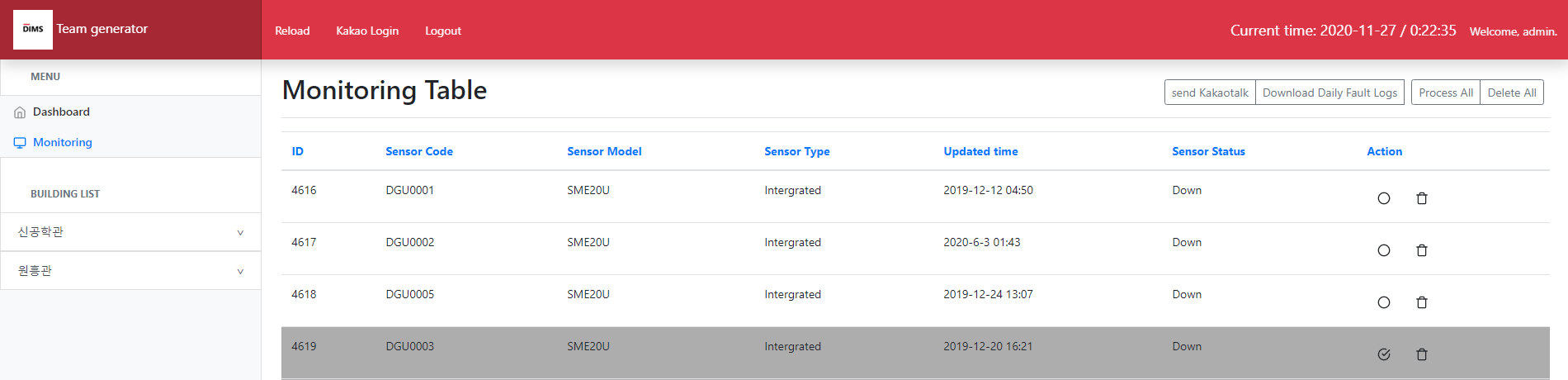


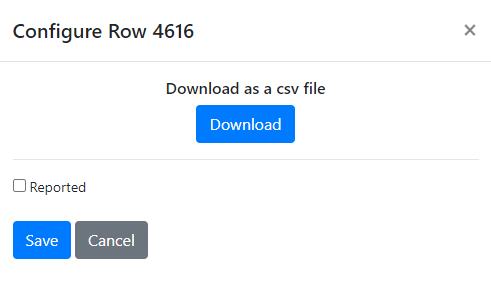
그림 9는 Monitoring Tab 화면이다. 해당 화면을 통해 문제가 생긴 디바이스의 기록을 확인할 수 있다. Updated time 과 Fault status를 통해 문제가 발생한 시점과 당시 status를 확인할 수 있다.

그림 10. Action Column



Action column에는 그림 10과 같이 두개의 버튼이 존재한다. 좌측의 동그란 버튼을 클릭하면 그림 11과 같은 창이 나타난다.

그림 11. Row Configure



만약 해당 문제 상황에 대한 긴급 보고서가 작성된 경우, Reported 체크박스를 체크하고 Save 버튼을 클릭한다. 이는 긴급 보고서가 작성되었음을 저장하는 행동이고 그 결과 그림 9 의 음영 표시된 row 처럼 보고서 작성이 완료되었음을 보여준다.

그림 10의 오른쪽에 위치한 Action column의 휴지통 모양 버튼은 해당 기록을 삭제하는 버튼이다.

그림 9의 Send Kakaotalk 버튼을 클릭하면 카카오톡 계정으로 로그인 할 수 있는 창이 나타난다. 로그인 후 긴급 알림 메시지를 보내고자 하는 계정을 선택하여 메시지를 보내고 나면 그림 12 와 같은 창이 나타난다.

그림 12. Send Kakaotalk Message

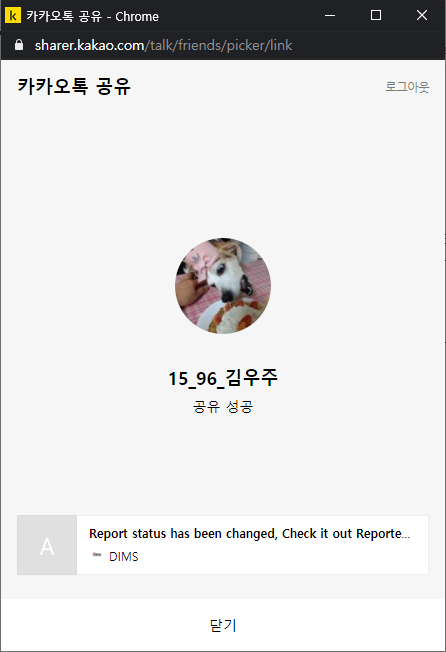
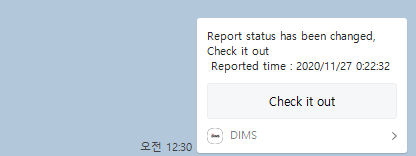


그림 13. Kakaotalk Message



해당 메시지를 받은 관리자는 그림 13 과 같이 메시지를 받게 되고 메시지에 연결된 링크를 클릭하면 웹 페이지로 이동하게 된다.

그림 9의 Download Daily Fault Logs 버튼을 클릭하면, 하루에 있었던 모든 문제 상황을 csv 파일 형식으로 다운로드한다. 다운로드 과정에서 그림 14와 같은 창이 나타나는데 이는 handled 된, 즉, 문제 상황이 해결된 log들은 삭제하는지 여부를 묻는 창이다. Confirm 버튼을 클릭하면 handled 된 log list는 삭제된다. 그림 15는 Download Daily Fault Logs 버튼을 통해 다운받은 monitoring.csv의 예시이다.

그림 14. Daily Fault Logs

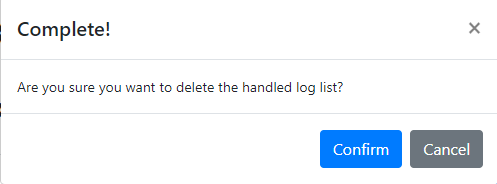


그림 15. monitoring.csv

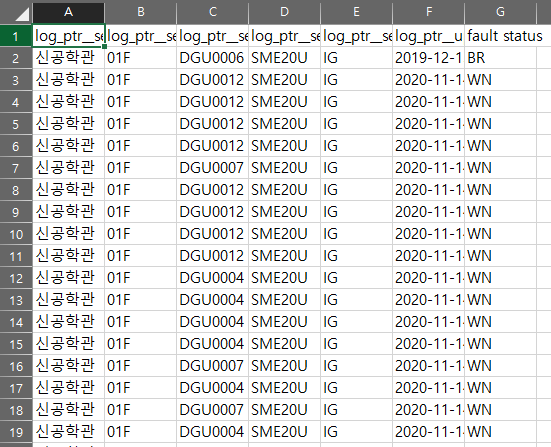


그림 9의 Process All 버튼을 클릭하면 Monitoring Table에 나타나는 모든 log list를 csv 형식의 파일로 다운로드 한다. 다운로드 과정에서 그림 16과 같은 창이 나타나는데 이는 모든 log들을 Reported 상태로 체크하는지 묻는 창이다. 만약 Reported 체크박스를 클릭하고 save 하면 모든 log 들은 보고서가 작성된 Reported 상태가 되고, Reported 된 row들은 음영 처리된다.

그림 16. Configure All Rows

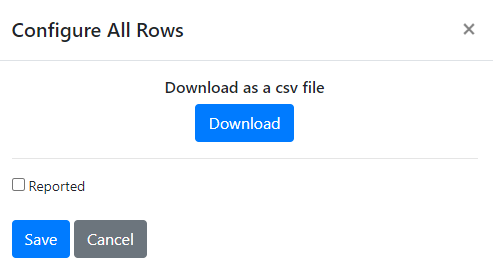


그림 17 는 Process All 버튼을 통해 다운받은 faultlogs\_unreported.csv 의 일부이다.

그림 17. faultlogs\_unreported.csv

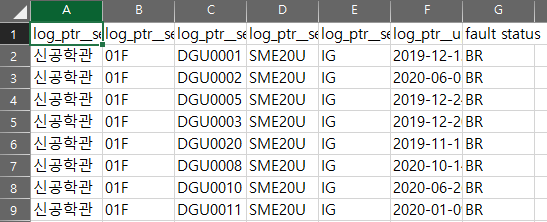


그림 9 의 Delete 버튼을 클릭하면 모든 log 들은 삭제된다.

CheckMate RPA를 통해 구현된 실행파일인 Generator\_RPA.exe 를 실행하면 가장 먼저 Monitoring Table 에서 문제가 발생한 log 존재여부를 확인한다. 존재하면 먼저 Send Kakaotalk 버튼을 통해 관리자에게 메시지를 발송한다. 이후 각 디바이스에 대해 긴급 보고서를 작성한다.

그림 18. Sensor Code Directory

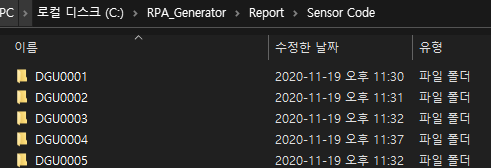


그림 19. Sensor Code Report Directory

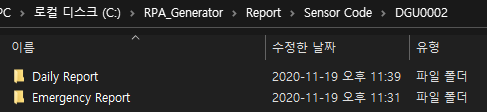


그림 20. Daily Report Directory

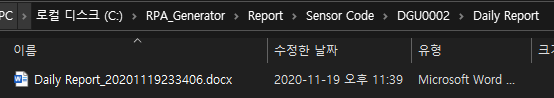
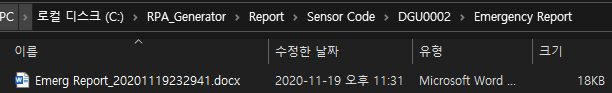


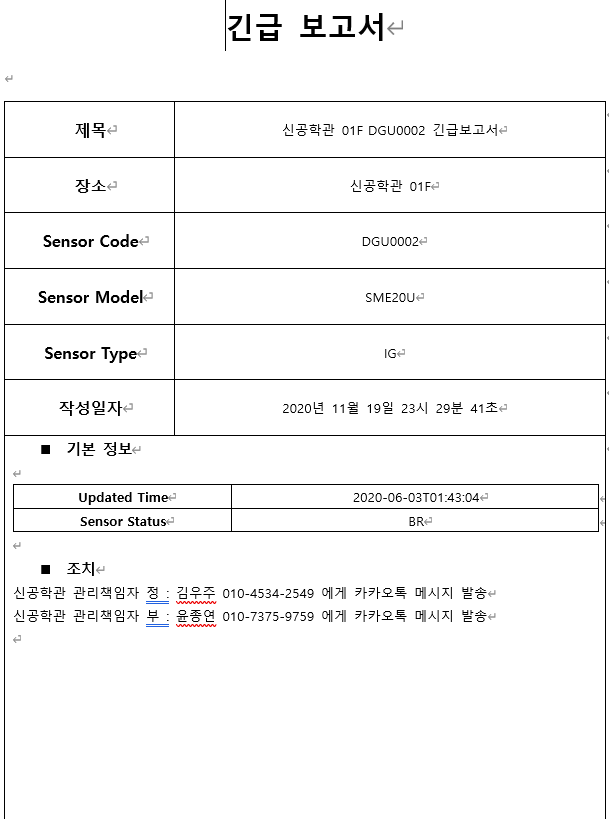
그림 21. Emergency Report Directory



긴급보고서와 일일보고서는 그림 18, 그림 19, 그림 20, 그림 21과 같이 “센서 코드명 폴더 / 긴급보고서 폴더 / 긴급보고서”, “센서 코드명 폴더/ 일일보고서 폴더 / 일일보고서” 와 같은 디렉토리 구조로 생성된다.

RPA를 통한 긴급보고서는 그림 22 와 같이 작성된다.

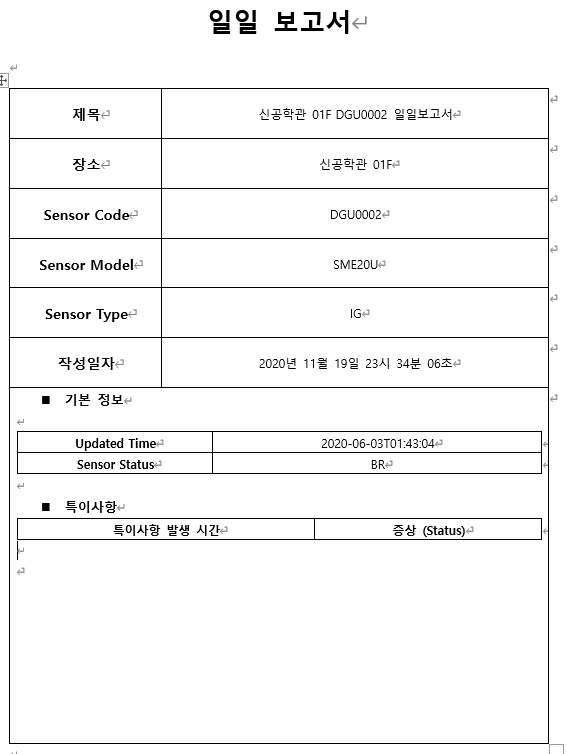
그림 22. Emergency Report



긴급 보고서 작성이 끝난 후 RPA를 통해 계속해서 문제 여부를 모니터링 하고 문제가 발생하면 다시 메시지를 전송하고 보고서를 작성한다.

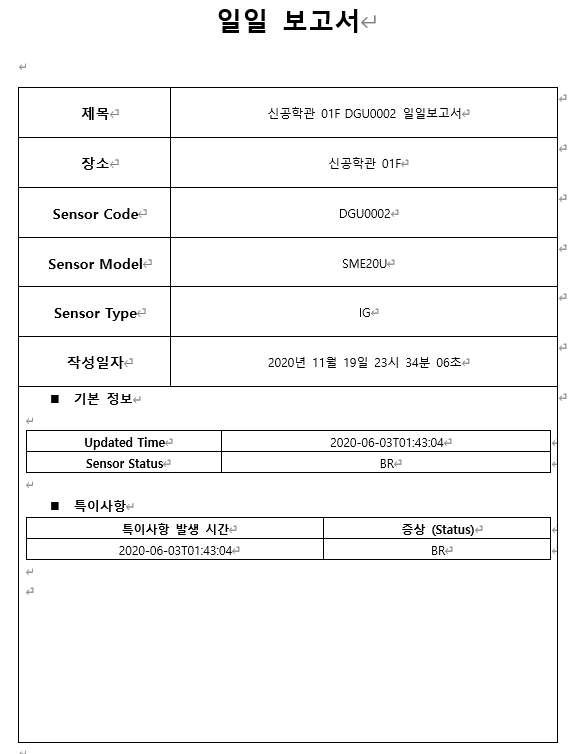
일일보고서를 작성하는 시간인 23:30~23:59이 되면 RPA는 일일 보고서를 작성한다. 먼저 그림 4 화면인 Dashboard Tab으로 이동하여 Export to file 버튼을 클릭하고 모든 디바이스의 현 상태를 csv 파일로 다운로드 한다. 이후 일일보고서를 작성한다. 일일보고서는 그림 23 과 같이 작성된다.

그림 23. Daily Report



모든 디바이스에 대한 일일보고서 작성이 끝나면 Monitoring Tab으로 이동한다. Download Daily Fault Logs 버튼을 클릭해서 하루에 있었던 모든 문제 log 를 다운로드 하고, 해당 디바이스의 일일보고서에 추가한다. 추가된 보고서는 그림 24 와 같다. 그림 23 과 비교하면 특이사항 란에 문제 발생 시간과 그 당시 status 가 추가되었음을 확인할 수 있다.

그림 24. Daily Report (Daily Fault Logs added)



모든 일일보고서 작성이 완료된 후 RPA는 다시 웹 페이지를 모니터링하면서 실시간으로 오류 항목이 발생했는지 확인한다.

**5.2.2 Implementation Tools (구현 도구)**

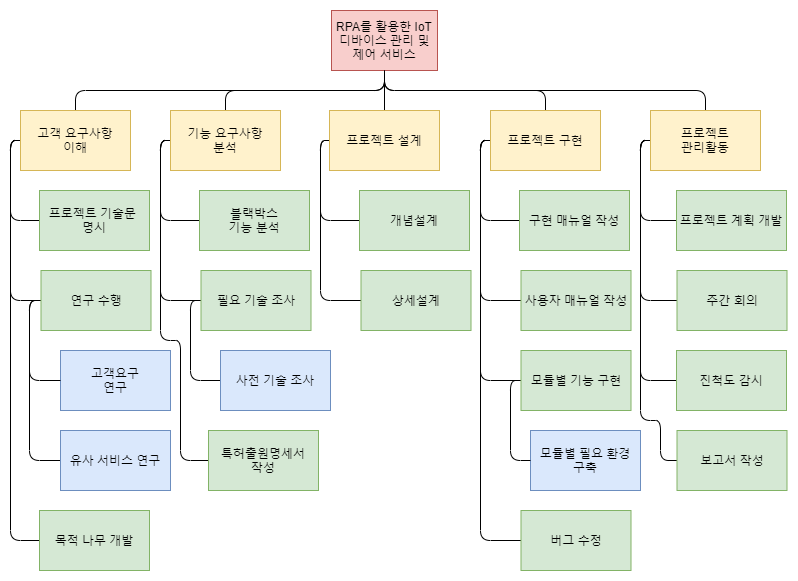
클라우드 서버로, 아마존 사의 클라우드 서버 플랫폼인 AWS를 사용한다. AWS는 리눅스 쉘을 사용자에게 제공하므로 서버 설정 및 조작이 용이하기 때문이다.

RPA 솔루션은 SYMATION 사의 CheckMATE RPA를 사용한다. CheckMATE RPA는 이미지 데이터 인식기능을 가져 AutoScript 기능을 통해 손쉽게 행동기반의 자동화 스크립트를 작성이 가능하므로 개발 언어를 모르는 사람도 손쉽게 스크립트를 작성할 수 있다. 또한 UI 및 Object 인식기능을 가져 Win32 프로그램, 각종 인터넷 브라우저 프로그램의 오브젝트를 인식하여 스크립트 작성이 가능하므로 Windows 내에서 동작하는 모든 Application의 자동화가 가능하다.

**5.3 Tasks and Task Assignments (과업 및 과업 분담)**

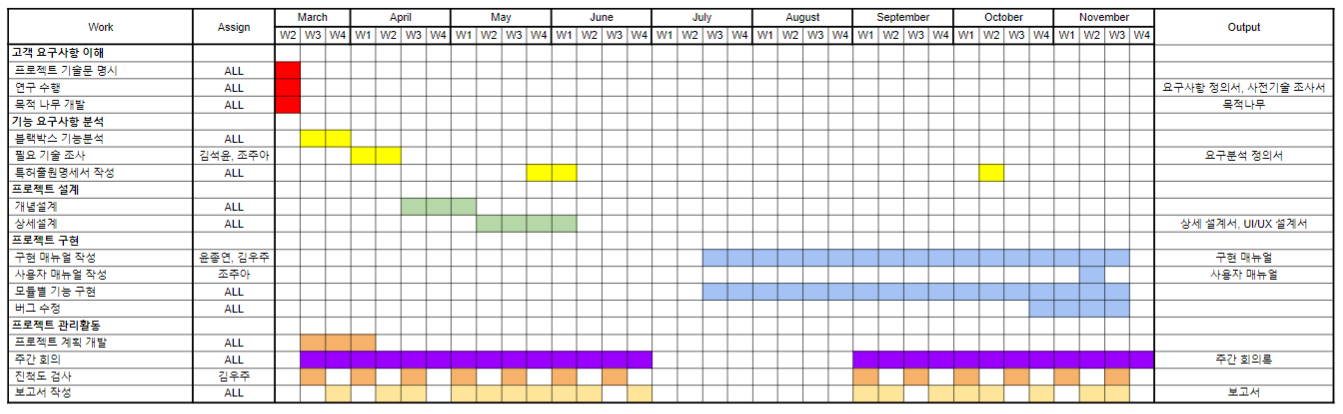
**5.3.1 Work Breakdown Structure (과업 분해)**

그림 25. Work Breakdown Structure



**5.3.2 Task Assignment and Project Timeline (과업 분담 및 과제 진행)**

그림 26. 과업 분담 및 과제 진행에 대한 Gantt-Chart.

****

본 프로젝트의 팀 구성원 평가표는 다음과 같다.

표 1. 팀 구성원 평가표

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tasks** | **Detail Description** | **Weight**  **(1-10)** | **김석윤**  **(%)** | **김우주**  **(%)** | **윤종연**  **(%)** | **조주아**  **(%)** | **Total (%)** |
| 보고서 작성 | 프로젝트 진행에 관련한 여러 문서 작성 | 5 | 15 | 25 | 15 | 45 | 100 |
| 관련 기술 조사 | 선행 기술 조사 및 필요 기술 조사 | 3 | 40 | 20 | 20 | 20 | 100 |
| 프로젝트 설계 | 개념설계 및 기능별 상세 설계 | 7 | 15 | 30 | 40 | 15 | 100 |
| 진행과정 발표 | 매 주 진행과정을 담당교수에게 보고 | 4 | 5 | 75 | 10 | 10 | 100 |
| 구현 | 프로토타입 제작 | 10 | 25 | 30 | 35 | 10 | 100 |
| **Individual Sum** |  |  | **570** | **995** | **805** | **530** |  |

**6. Anticipated Results and Contributions (예상 결과 및 기대 효과)**

RPA를 교내 IoT 디바이스 관리 업무에 도입하여 사용자가 편리하게 디바이스들의 상태를 확인하고 긴급상황에 적절한 조치를 취할 수 있도록 하였다. 매 시각 로그데이터를 분석한다는 점에서 관리의 안정성을 도모한다. 그리고 문서작성의 자동화는 인력과 시간을 절약하고, 이로 인해 향상된 ROI를 기대할 수 있을 것이다.

뿐만 아니라 현재 RPA는 금융권 기업을 대상으로 단순하고 반복적인 업무를 대체하는 용도로 많이 보급되어 있다. RPA가 적용될 수 있는 분야는 다양하지만, 아직 금융권 등 제한적인 분야에서만 활용되어 다른 활용 가능성에 대한 연구가 부족한 실정이다. 국내에 도입된 지 2~3년밖에 되지 않았지만, 이를 교내 IoT디바이스 관리환경에 도입하여 업무를 효율적으로 처리할 수 있도록 함으로써 RPA의 활용분야를 넓히는 데에 일조할 수 있을 것이다.

**7. References (참고 서적)**

*Global eProcure, Robotic process automation for supply chain management operation, 15593025 , 2017.05.11.*

*주식회사 열린기술, RPA를 활용한 플랜트 사고 시뮬레이션 데이터베이스 구축 시스템, 10-2018-0121753, 2018.10.12.*

*아미쿠스렉스 주식회사, 법률문서 자동작성장치 및 방법, 1020180099456, 2018.08.24.*

*+3시그마 관련 문서*

**8. Budget (예산)**

- 서버 비용 (Pythonanywhere 서비스를 통해 서버 호스팅)

- RPA 솔루션 시리얼 비용