

시 스템 규 격 서

(System Specification)

제목 : 스마트팩토리 통합 관리 시스템
(과목명 : IoT프로젝트2)

참여자 : 허정운
신비
이유비
최소희

1. 개요

1.1. 기획동기

구분	현장자동화	공장운영	기업자원관리	제품개발	공급사슬관리
고도	IoT/IoS 기반의 CPS화				인터넷 공간 상의 비즈니스 CPS 네트워크 협업
	IoT/IoS화	IoT/IoS(모듈)화 빅데이터 기반의 진단 및 운영			
중간2	설비제어 자동화	실시간 공장제어	공장운영 통합	시뮬레이션과 일괄 프로세스 자동화	다품종 개발 협업
중간1	설비데이터 자동집계	실시간 의사결정	기능 간 통합	기술 정보 생성 자동화와 협업	다품종 생산 협업
기초	실적집계 자동화	공정물류 관리(POP)	관리 기능 중심 기능 개별 운용	서버를 통한 기술/납기 관리	단일 모기업 의존
ICT 미적용	수작업	수작업	수작업	수작업	전화와 이메일 협업

기존의 작업 데이터 수집은 수기로 이루어져 왔기 때문에 누락 및 오입력이 될 가능성이 높다. 따라서 입력된 데이터가 정확하지 않을 수 있으며 사람이 직접 하기 때문에 시간적, 인적 자원을 이용해야 하는 문제를 해결하고자 스마트 팩토리라는 개념이 도입되었다. 스마트 팩토리의 도입 목적은 경제성과 생산성 향상에 있다. 따라서, 디지털화 된 데이터를 가지고 시간과 장소에 구애받지 않으면서 인적 자원까지 아낄 수 있다면 작업 및 업무 프로세스의 효율성이 상당히 증가할 것이고, 이를 통해 생산성을 향상시킬 수 있다고 생각하여 해당 주제를 선정하였다.

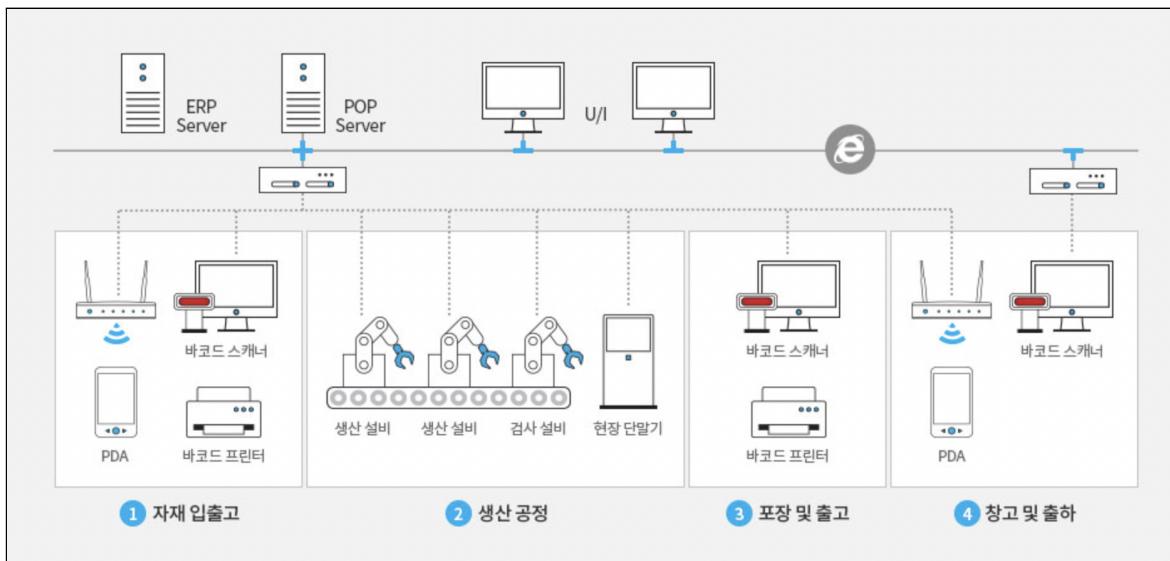
1.2. 분석

부록1-1) 특허 분석

부록1-2) 선도기업 분석

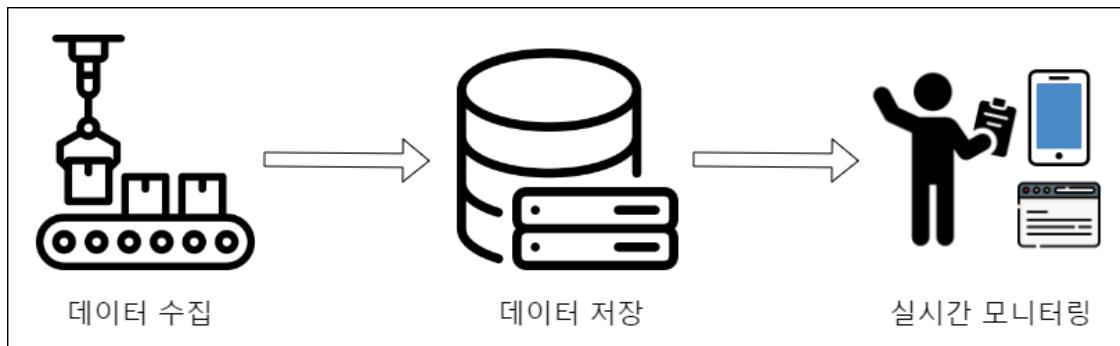
부록1-3) 정부 지원 및 규제 참조

1.3. 분석 결과



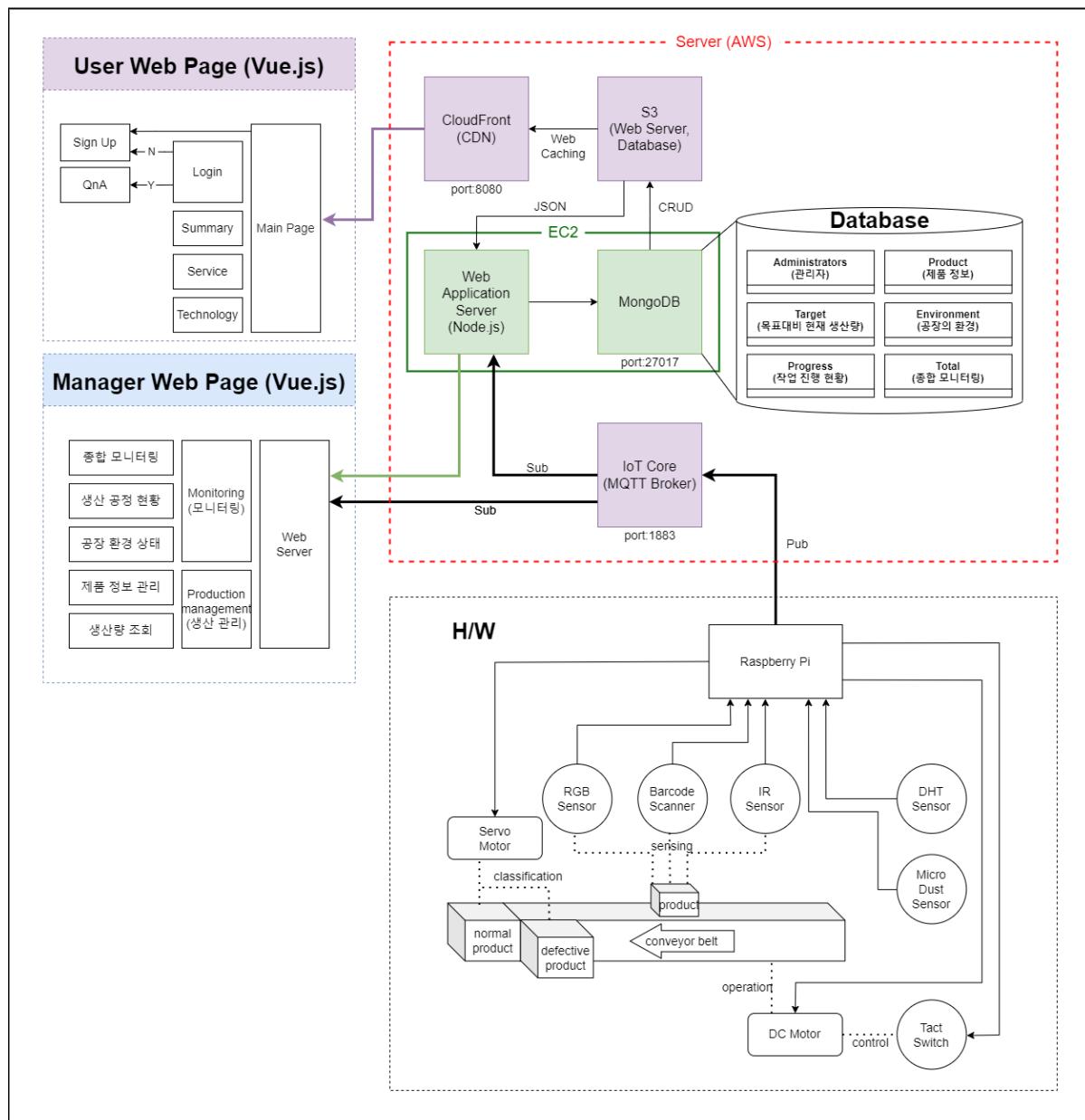
스마트팩토리는 제품의 기획부터 판매까지 모든 생산과정을 ICT(정보통신)기술과 융합해 최소 비용과 시간으로 고객 맞춤형 제품을 생산하는 사람 중심의 첨단 지능형 공장이다. 선도 기업이 추구하는 공통적인 방향성은 ‘효율적인 생산 관리’, ‘생산성 향상’, ‘품질 향상’, ‘에너지 절약’, ‘안전성 향상’, ‘보안성 향상’이다.

1.4. 목표



위와 같은 스마트팩토리에 들어가는 모든 기술을 우리가 구현하기에는 어려움이 있다고 판단하였다. 따라서, 생산 관리 부분에 초점을 맞추어 현실적으로 구현 가능하며, 효율적인 시스템이 필요한 공정에 적용할 관리 시스템을 만들고자 한다. 전산화된 생산 공정의 데이터를 수집하여 관리자에게 제공하는 시스템은 기존 스마트 팩토리에서 흔하게 적용되어 있기에, 단순히 수집된 데이터를 제공하는 것을 뛰어넘어 데이터를 기반으로 공장을 더욱 효율적으로 관리 가능한 솔루션을 제공하고자 한다. 해당 솔루션 제공을 위해 일일 생산량, 주간 생산량, 월간 생산량 등을 계산하여 모니터링에 반영하고, 추후 인공지능 기술을 접목하여 제품의 불량 원인 추적 및 해결 방법을 제시할 수 있도록 한다.

2. 전체 시스템 구조



부록2-1

The screenshot displays the TIMIROOM MES system interface, which includes several key modules:

- 종합 모니터링**: A dashboard showing real-time status of services, current production volume, and product count.
- 생산 공정 관리**: A module for managing the production process, showing current status and historical data.
- 공장 환경 상태**: A module for monitoring factory environment conditions like temperature, humidity, and dust levels.
- 생산량 조회**: A module for querying production volume and yield.
- 제품 정보 관리**: A module for managing product information, including codes and details.

부록1-4

2.1. 개발 환경

• Server

A. 환경구성

- 사양 : AWS(EC2/t2.micro), Ubuntu 20.04 LTS

B. 개발언어

- Node.js : Javascript 백엔드 프레임워크

• Web

A. 개발언어

- 프레임워크 : Vue.js
- Web Component : HTML, CSS, Javascript

• H/W

A. 환경구성

- 사양: 라즈베리파이4B 8GB
- OS: RasbianOS(Linux/64bit)

B. 개발언어

- Python

• DB

A. 개발언어

- MongoDB
- MongoDB Compass(UI)

B. 서버와의 연동

- node.js

2.2. 주요수행내용

2.2.1. 하드웨어

a. 실시간 데이터 수집

- 공장 환경 데이터 수집: 온습도 센서, 미세먼지 센서를 통해 수집한 데이터 서버로 전송
- 제품 데이터 수집: 바코드 스캐너 모듈을 통해 읽어들인 바코드 정보, 색상 감지 모듈로 판별한 정상 제품 여부 서버로 전송

b. 프로토타입

- 컨베이어 벨트: DC모터를 이용한 컨베이어 벨트 가동, 택트스위치 이용 또는 웹의 요청에 따른 컨베이어 벨트 작동과 중지 제어
- 물체 감지: 적외선 송수신 센서를 통한 물체 감지
- 제품 분류: 색상 감지 모듈로 판별한 정상 제품 여부에 따라 서보모터 각도 제어를 통한 정상 제품과 불량 제품 구분

2.2.2. 서버

- Web Application Server
- 하드웨어에서 수집한 데이터를 DB로 저장
- 저장한 DB를 App/Web에서 호출 시 통합된 데이터로 변환하여 응답
- 로그인 및 회원가입 -> 계정 관리

2.2.3. DB

- 관리자 정보 DB에 저장, 로그인 시 관리자 권한 확인
- 입고 생산품의 바코드 스캔 시 제품 정보 DB로 저장
- 작업 현황 관련 데이터 저장과 설비 가동 상태에 따른 데이터 처리
- 관리자로부터 목표 생산량 입력 받아 저장, 현재 생산량 수집
 - >> 목표대비 생산량의 작업 실적 정보 제공
- 센서로부터 공장의 환경 데이터 수집
- 종합 모니터링 지원하기 위한 데이터 및 통계 결과 저장

2.3.4. Web

- 관리자 로그인 및 회원가입 기능
- 온도, 습도, 미세먼지 변화 모니터링
- 컨베이어 벨트 on/off 기능
- 작업장 내 불량 내역, 작업 현황과 실적 확인
- 제품의 일련번호 품목코드 데이터 저장 및 확인
- 생산 과정 내 오류 발생 모니터링 및 알림 기능

3. 테스트 시나리오

3.1. 하드웨어

분류		내용		결과
데이터 수집	공장 데이터	온도	온습도 센서로 현재 온도 측정 확인	정상 측정 시 값을 temp에 담아 mqtt 전송
		습도	온습도 센서로 현재 습도 측정 확인	정상 측정 시 값을 humid에 담아 mqtt 전송
		미세먼지 농도	미세먼지 센서로 현재 미세먼지 농도 측정 확인	정상 측정 시 값을 dust에 담아 mqtt 전송
	제품 데이터	감지	적외선 센서로 제품 감지	제품 감지 시 바코드 스캔
		정보	바코드 스캐너로 제품의 바코드를 인식	정상 인식 시 값을 num, item, prod_date에 담아 서버 전송
		불량 여부	색상 감지 모듈로 색상 감지	색상 감지 시 white(255, 255, 255)일 경우 정상 제품(0), 아닐 경우 불량 제품(1)으로 판별하여 is_defect에 담아 mqtt 전송
프로토 타입	컨베이어 벨트	작동	DC모터 단방향 가동 확인	
		제어	1) 택트 스위치 push 시 DC모터 작동/중지 확인 2) 웹에서 가동/중지 버튼 클릭 시 DC모터 작동/중지 확인	1) 가동 상태를 state에 담아 mqtt 전송 2) state가 1일 때 작동, 0일 때 중지
	제품 분류	1) is_defect가 1일 때 서보모터 30도 2) 0일 때 5도로 제어 확인		
통신	MQTT	연결	라즈베리파이 부팅 후 MQTT 연결 상태 확인	
		전송	데이터 전송 확인	temp, humid, dust, num, item, prod_date, state, is_defect 값 전송
		수신	데이터 수신 확인	state 값 수신

3.2. 서버

분류	내용	결과
데이터 저장	1) 데이터를 하드웨어(Publish)에서 지정된 Topic으로 JSON 형태로 메시지 발행 2) 해당 Topic을 IoT Core Broker Server를 통해 Subscribe하고 있는 WAS에서 데이터를 받아 S3에 저장	데이터가 지정된 테이블에 입력됐는지 상호 비교
데이터의 호출 및 표시	1) 데이터를 Web에서 get 요청 2) DB가 담긴 S3로부터 JSON 형태로 호출 3) Web으로 return	호출한 값과 Web에 표시된 데이터가 일치하는지 확인
회원가입	1) client가 회원가입 요청 시 데이터를 post로 JSON 형태로 cors 통신을 통해 넘겨 받는다. 2) 데이터를 쿼리문을 통해 DB에 insert 3-1) 일치하는 데이터가 없을 경우, 성공 메시지를 return 3-2) key 혹은 id가 중복 될 경우, 에러 메시지를 return	web에서 입력한 데이터, 서버에서 받은 데이터, DB에 저장된 데이터가 일치하는지 확인
로그인	1) client가 로그인 요청 시 web에서 get 요청 2) 일치하는 데이터가 있는지 쿼리문을 통해 탐색 3-1) 일치하는 데이터가 있는 경우, 성공 메시지를 return 3-2) 일치하는 데이터가 없는 경우, 에러 메시지를 return 3-3) 비밀번호가 다를 경우, 에러 메시지를 return	요청한 아이디로 로그인이 되는지 확인(web)
MQTT	1) MQTT Broker(IoT Core)에서 발행되는 모든 메시지를 저장 2) Publish에서 지정 Topic으로 발행되는 메시지를 확인 3) Subscribe에서 메시지가 정상적으로 받아지는지 확인	Pub과 Sub, Broker를 확인하여 데이터가 모두 일치하는지 확인

3.3. DB : 데이터 관리

분류	내용	결과
관리자 관리	1) 회원가입 시 관리자 정보 저장 2) 로그인 시 관리자 정보 제공	Admin 테이블에 관리자가 입력한 정보 정상적으로 저장
제품 관리	1) 제품 바코드 스캔 시 제품 정보 저장 2) 제품 불량 여부 수집	바코드에 저장된 제품 데이터 Product 테이블에 정상적으로 저장
작업 현황 관리	1) 설비 가동 상태 수집 2) 현재 생산량 수집 3) 현재 불량 제품 개수 수집	1) 설비가 켜진(on) 상태이면 Progress 테이블의 설비 가동 상태(state) 필드가 true, 꺼진(off) 상태이면 설비 가동 상태(state) 필드가 false로 설정 2) 제품이 인식되면 Progress와 Total 테이블의 현재 생산량(prod_vol) 값 1 증가 3) 인식된 제품이 불량이면 Progress와 Total 테이블의 불량 제품 개수(defect_cnt) 값 1 증가
환경 관리	공장 현장의 온도, 습도, 먼지 농도 수집	Environ 필드에 공장 환경 데이터 실시간 저장
통계 관리	1) 목표 생산량 대비 현재 생산량을 계산한 달성을 데이터 제공 $\text{now(달성을)} = \text{prod_vol}(\text{현재 생산량}) / \text{tar_vol}(\text{목표 생산량}) * 100$ 2) 현재 생산량 대비 현재 불량 제품 개수를 계산한 불량률 데이터 제공 $\text{defect_rate(불량률)} = \text{defect_cnt(불량 개수)} / \text{prod_vol(현재 생산량)} * 100$	1) 실시간으로 수집되는 생산량 및 불량 개수 데이터 정상적으로 반영 2) 통계 결과 정상적으로 도출 및 저장 3) 현재 생산량, 불량 개수의 변화에 따라 통계 결과 정상적으로 실시간 업데이트

3.4. Web : 실시간 모니터링

분류	내용	결과
로그인	1) 아이디, 비밀번호 입력창 2) 회원가입	회원가입의 사용자 정보가 서버로 전달되는 것을 확인
종합 모니터링	설비 명, 작업자, 가동 상태, 생산 목표량, 불량 수량, 현재 생산량, 달성을, 불량률을 한 화면에서 모니터링을 제공하며, DB와 서버에서 정보를 가져와 보여준다	서버에서 DB의 정보 이동, 모든 정보 화면 표시 확인
생산 공정	설비 가동 상태, 생산 목표량, 품목 번호, 불량품, 생산량의 현황을 볼 수 있는 웹 화면 제공	설비 가동 상태가 정확한지 확인
공장 환경 상태	온도, 습도, 미세먼지의 변화를 시각적으로 제공	온도, 습도, 미세먼지의 실시간으로 변화하는 상태 그래프 확인
생산량 조회	제품수량, 불량수량, 품목 번호의 정보를 제공	1) 서버에서 DB의 데이터를 가져온다 2) 가져온 정보가 정확한지 확인
제품 정보 관리	품목 번호, 품목, 생산 일자의 정보	모든 정보가 화면에 표시되도록 한다

3.5. 이벤트 기준 전체 시스템

이벤트	절차		결과
회원가입	1) Web에서 입력하여 서버로 회원가입 요청 2) JSON 형태로 DB의 Admin 테이블에 저장		Admin 테이블에 관리자 저장
로그인	1) 관리자가 Web에서 로그인 요청 2) 서버가 저장되어 있는 데이터와 대조 3) 일치 시 메인 페이지로 이동		if 관리자 정보 존재: 메인 페이지로 이동 else: 로그인 실패
제품 생산	정상	1) 제품의 바코드 스캔 후 서버로 전송 2) DB의 Product 테이블에 저장 3) Progress, Total 테이블의 전체 생산량(prod_vol) 값 1 증가	1) Product 테이블에 제품 정보 저장 1.1) is_defect = false 2) Progress 및 Total 테이블의 전체 생산량(prod_vol) 값 1 증가
	불량	1) Product 테이블의 불량 여부(is_defect) 값 1 설정 2) Progress, Total 테이블의 불량 제품 개수(defect_cnt) 값 1 증가	1) Product 테이블에 제품 정보 저장 1.1) is_defect = true 2) Progress 및 Total 테이블의 전체 생산량(prod_vol) 값 1 증가 3) Progress 및 Total 테이블의 불량 제품 개수(defect_cnt) 값 1 증가
모니터링	1) DB에 저장되어 있는 데이터를 Web으로 전송 2) Web에 표시되는 데이터가 일치하는지 교차 확인 3) 실시간 데이터 변화가 정상적으로 적용되는지 확인		1) 관리자 화면에 제품 정보, 생산 작업 현황, 작업 실적, 공장 환경, 요약 데이터가 설계 화면과 같이 표시됨 2) 불량 제품 개수, 현재 생산량, 달성을, 불량률과 같은 실시간 변화하는 데이터 정상적으로 적용됨

4. 자체 평가 기준

개요) 스마트 팩토리라는 주제에 걸맞게 공장 내 업무 효율성 증대와 관리자의 편의성에 중점을 두고 고안한 작품이기 때문에 전체 구조를 이해하고 시스템을 설계, 제작하는 것이 중요하다. 따라서 기본적인 객관성을 비롯한 각 파트 간 원활한 소통, 제작 시스템에 대한 이해도와 완성도를 자체 평가의 기준으로 두었다.

1) 기획 작품에 관한 사전 조사가 충분했는가. (10점)

산업과 연계되는 분야를 다루는 작품인 만큼 도메인 지식을 최대한 갖추기 위해 충분한 사전 조사는 필수적이므로 모든 팀원이 성실히 임해야 할 것이다.

2) 각 파트 간의 협력과 협업이 잘 이루어졌는가. (10점)

각자 수행해야 하는 작업뿐만 아니라 전체적인 구조에 대한 동일한 이해와 목적이 중요시되어야 한다. 따라서 활발한 회의와 협력적인 태도가 필요할 것이다.

3) 이 작품에서 각 파트 간의 통신과 연결을 원활하게 구현하였는가. (15점)

각 시스템의 통신·연결이 원활하게 이루어져야 전체 시스템의 완성도를 높일 수 있다. 따라서 비중을 조금 더 두었다.

4) 이 작품은 오차(오류) 없이 작동하는가. (5점)

하드웨어 부분, 로직 및 데이터 처리 과정에서 작동이나 계산 오류, 데이터 오차를 최소화해야 한다. 가장 기본적인 항목이라고 할 수 있다.

5) 작품은 처음 기획한 의도로 완성되었는가. (15점)

초기 기획한 목적을 바탕으로 끝까지 작품을 제작하였는가 평가하는 항목이다. 개인적으로 수정하는 사항이나 목적을 바라보는 관점의 차이 없이 작품을 제작해야 한다.

6) 이 작품이 실제 현장에 적용될 경우, 기존 시스템과의 괴리감 없이 연동 가능한가. (10점)

현재 상용화되고 있는 시스템이기도 하며, 실제 현장의 문제점을 해결하기 위한 작품이기 때문에 공장 현장에 적용될 경우의 연동성에 대해 고려해야 한다.

7) 이 작품이 상품화 될 경우, 실제 사용자의 편의성과 현장의 효율성을 높일 수 있는가.

(5점)

이 시스템의 목적이지만, 실제 상품화가 되지 않으면 구체적으로 일기 어려운 항목이므로 비중을 적게 두었다.

8) 이 작품은 경제성이 있는가. (5점)

실제 산업에 적용되는 시스템을 설계한 것이므로, 해당 작품이 상품화 될 정도의 경제성이 있는가를 고려해야 한다.

9) 각 개인의 파트를 정확히 이해했으며, 설계할 수 있는 역량을 갖추게 되었는가. (15점)

전체 시스템을 위해 각자 담당한 파트를 견고하게 완성하는 것이 중요하다. 따라서 담당한 파트를 구체적으로 이해하고 습득하는 능력이 요구된다.

10) 개인의 개발 환경을 충분히 공부했으며, 맡은 업무를 성실히 수행하였는가. (10점)

하드웨어/소프트웨어 개발을 진행하기 위해서는 본인의 개발 환경에 대한 이해가 기본 바탕이 되어야 한다. 또한 맡은 업무를 정해진 기간에 맞춰 수행해야만 원활한 협업이 가능하다.