**2021학년도 1학기**

**캡스톤 디자인 교과목**

**캡스톤 디자인 결과보고서**

|  |  |  |
| --- | --- | --- |
| 소 속 (전공) | 컴퓨터공학과 | |
| 팀 명 | 지하보다 지상 | |
| 지 도 교 수 | 양효식, 이수정 | |
| 팀 장 | 학번: 16011009 | 이름 : 서지상 |
| 팀 원 | 학번: 16010995 | 이름 : 조성우 |
| 학번: 16011034 | 이름 : 박찬영 |
| 학번: 16011060 | 이름 : 장현희 |
| 제 출 일 자 | 2021. 06. 11 | |

**세종대학교 공학교육센터**

**최종보고서**

**1. 개발 목표**

현재 전력 소비는 IT기기, LED조명, 디스플레이 기기 등 고밀도 DC전력을 중심으로 이루어지고 있다. DC는 특성상 송전 자체의 효율성이 AC보다 높고, 전압 변경기술의 발전으로 인해 DC의 단점이던 원거리 송전 부문에서도 DC의 우월성을 인정받고 있다. 하지만 우리나라는 현재 교류(AC)를 기반으로 한 전력 시스템을 사용하고 있어 직류(DC)를 기반으로 한 전력 시스템으로 바꿀 필요성이 있다. 또한, 전력 소비의 증가로 공급량을 맞추기 위해 많은 발전소가 지어지고 있는데, 현재 지어지고 있는 발전소들은 탄소 배출량이 많은 화석 연료를 사용하거나, 위험성이 높은 원자력 발전소 등을 사용하여 환경 오염 및 안전성 등에 대한 우려가 커지고 있다. 이러한 환경 오염 문제와 전력 부족 현상을 최소화하기 위해서는 현재의 중앙집중형 에너지 분배 시스템을 개선할 필요가 있다고 생각했다. 이에 우리는 공적으로 생산되고 있던 전기에너지를 신재생 에너지를 활용하여 개개인이 생산함으로써 발전소의 필요성을 낮춘다면, 공적 전력 소비를 줄일 수 있게 되고, 환경오염 문제 등에 대한 해결책이 될 수 있을 것이다.

우리는 앞의 결론에 의해 도출된 방안으로 DC 스마트 그리드 홈 네트워크 시스템을 개발하였다. 이는 기존의 AC-DC변환으로 인한 전력 손실을 DC-DC형태의 전력망으로 변경함으로써 에너지 효율을 2~10%정도 증가시킬 수 있을 것이고, 기존의 단방향 전력망에서 양방향 전송이 가능하게 하여 에너지 효율을 더욱 개선할 수 있다. 우리는 사용자의 접근성과 효과적인 전력 관리를 위해 웹 기반의 스마트 홈 기능을 적용하여 홈 네트워크에 연결된 전자기기의 전원(On/Off)을 원격으로 제어할 수 있고, 현재 사용하고 있는 전력량, 배터리의 잔량, 생산하고 있는 전력에 대한 정보를 제공하여 사용자가 효율적으로 전력을 관리할 수 있도록 도왔다. 또한, 소규모 홈 네트워크 그룹을 구성하여 그룹 간의 자급자족 네트워크를 구성하여 p2p 전력 거래를 가능하게 하였다. 이러한 기능들은 사용자가 자신의 전력 상황을 빠르게 파악하고 관리할 수 있게 돕고, 이를 통하여 거래할 수 있게 함으로 전력의 과소비를 줄이는 효과와 개개인의 전력을 거래함으로써 잉여 전력의 낭비를 줄이는 효과를 기대할 수 있다.

**2. 설계 사양서**

**① 요구사항 분석서**

**1) 차별성**

이번에 우리가 개발하게 될 DC 스마트그리드는 DC기반 전력 망의 사용으로 신재생 에너지에서 발생되는 DC 형태의 전기를 그대로 사용함으로써 AC에서 DC로의 전력 전환에 소모되는 손실을 줄였다. 또한 다른 기존 프로그램과 차이를 두고자 하는 부분은 개발하게 될 프로그램을 사용하게 되는 유저들의 접근성을 높여, 더욱 효율적인 전력 관리를 가능하게 하고, 소규모 그룹 간의 ‘자급자족’ 네트워크를 구성하여 p2p 전기 거래의 자유도를 높였다. 이는 기존의 전력 중개 사업을 하는 ‘Heazoom’등과 같은 회사에서 지원하고 있는 전력 생산량 및 판매 가격 이외에 현재 축전지의 축전 된 전기의 양, 총 사용하고 있는 전력량과 현재 DC 스마트그리드와 연결된 전자기기의 전원(on/off)을 관리하는 기능 등을 웹에서 제공함으로 접근성을 높였다. 자신의 전력 생산 및 사용 등에 대한 접근성이 높아진다는 의미는 자신의 전력 현 상황을 빠르게 파악하고 관리할 수 있다는 말과 같음으로 이는 전력의 과소비를 줄이는 효과를 기대할 수 있다. p2p 전기 거래의 자유화를 높이고 그 편의성을 높이기 위해 전기 거래 게시판을 개설하여 전기 거래를 희망하는 이와의 연결을 도움으로 자유도를 높였다.

**a) 전력 관리 효율 개선**

- 현재 고객이 생산하고 있는 전력량 표시

- 현재 고객의 배터리 잔량 표시

- 현재 전력 사용량 표시

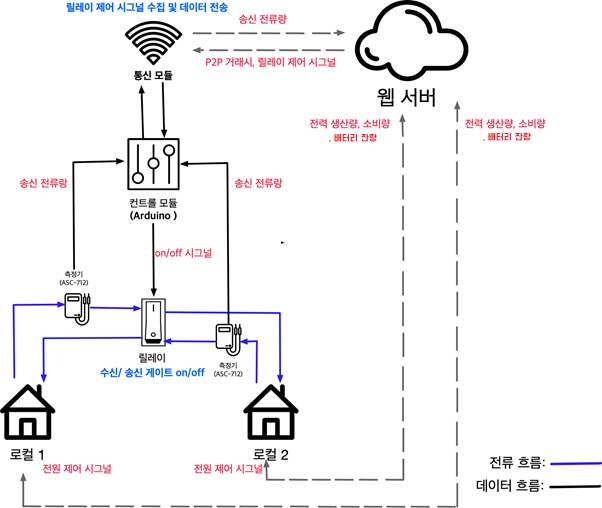
- 전원(on/off) 기능

**b) p2p 전기 거래 자유도 개선**

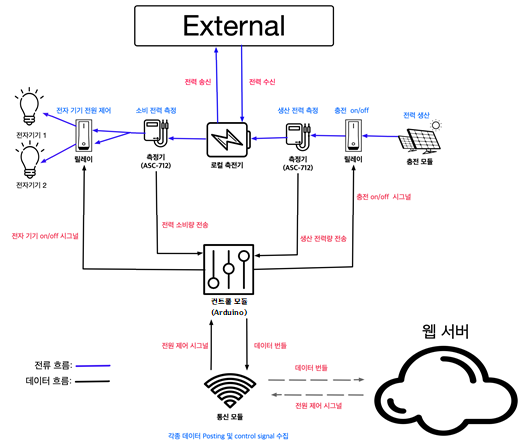
- 거래 게시판 개설

- 거래 글 작성 기능

**2) H/W 테스트셋구성도**



**[전체 하드웨어 구상도]**



**[로컬 하드웨어 구상도]**

**3) 상세개발방법**

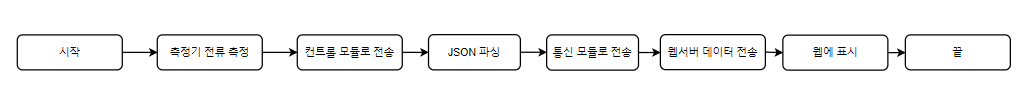
**a) 신재생 에너지 충전 제어**

- 충전 모듈을 릴레이를 사용하여 컨트롤



**b) 소비량, 배터리 잔량측정 및 표시**

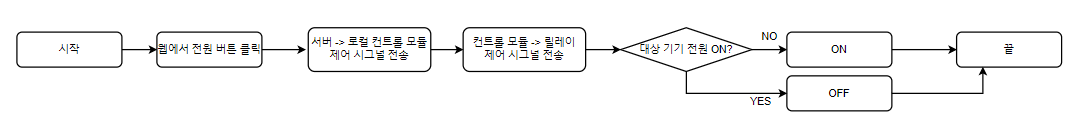
- ASC712를 사용하여 연결되어 있는 Arduino에서 총 소비량 및 배터리 잔량을 계산하고 웹 서버와 통신을 통해 각자 웹에 표시



**c) 원격 ON/OFF 기능**

- 릴레이를 Arduino에 연결하여 ON/OFF를 조정한다.

- 웹에서 ON/OFF를 선택하게 되면Arduino와 HTTP통신을 통해 릴레이를 조정한다.



**d) P2P전기 거래**

- 테스트 셋에서는 로컬이 2개로 1ch 릴레이와 ASC-712를 한 세트로 각각 로컬의 배터리와 연결한다.

- 연결되어 있는 1ch 릴레이와 ASC-712를Arduino(통신 모듈)와 연결하여 웹 서버와 통신한다.

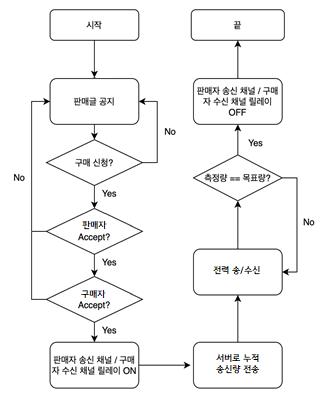
- 웹은 게시판 형식으로 거래 글을 등록하게 한다.

- 거래 요청을 하게 되면 글을 등록한 판매자에게 해당 거래 요청에 대한 수락요청 메일을 보낸다.

- 판매자가 거래 요청을 수락하게 되면, 구매자에게 최종 승인 이메일이 보내진다.

- 구매자가 최종 승인을 하게 되면, 거래 진행에 대한 안내 메일이 판매자와 구매자에게 보내진다.

- 거래는 Arduino의 통신 모듈(esp-01)과 통신하여 해당하는 릴레이를 열어주도록 한다.

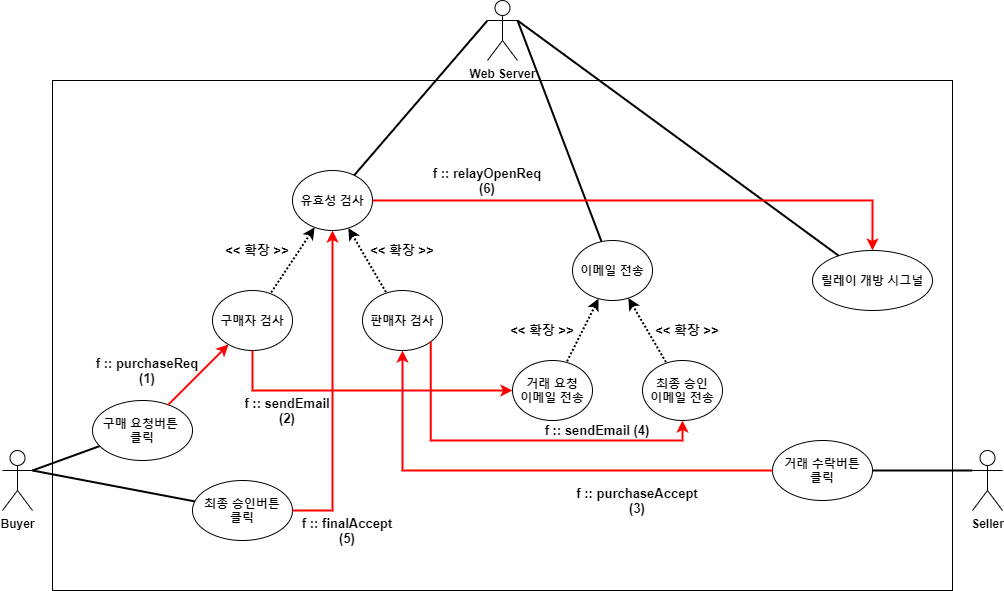


**4) 개발환경**

|  |  |  |
| --- | --- | --- |
| **H/W** | **Arduino.cc** | |
| **S/W** | **BackEnd** | **Node.js, Express** |
| **FrontEnd** | **HTML,CSS, ES6** |
| **DB** | **MongoDB, Mongoose** |

**② 유스케이스 다이어그램**

**1) 전력 거래 유스케이스**



**[전력 거래 Usecase]**

**2) Actor 목록**

|  |  |  |
| --- | --- | --- |
| **Actor** | **구분** | **설명** |
| Seller | 시스템 사용자 | 전력 판매자, 거래 글 게시자 |
| Buyer | 시스템 사용자 | 전력 구매자,거래 신청자 |
| Web Server | 제어 및 관리 | 전력 거래 중개 및 거래 과정 통괄 |

**3) P2P 전력거래 개요**

등록된 판매 글에 구매자가 거래를 요청할 시, 구매자에 대한 유효성 검사가 시행되고, 통과 시, 판매자에게 거래 요청에 대한 이메일이 전송된다. 판매자가 거래 수락을 하면, 판매자에 대한 유효성 검사가 시행된다. 통과 시, 웹 서버에서 구매자 측에 최종 승인에 관한 메일을 전송한다. 제한시간 내에 구매자가 최종 승인을 하면, 구매자와 판매자 측에 대한 유효성 검사를 시행한다. 이후, 웹 서버에서 릴레이 컨트롤 모듈로 송전 채널을 열기 위한 신호를 보낸다. 이를 수용한 컨트롤러는 채널을 열어주고 송전을 진행한 후 전력 거래가 완료되면 채널을 닫고 웹 서버에 거래 완료 신호를 보낸다.

**4) 사건 흐름**

**\* 기본 흐름 (T)**

|  |  |  |
| --- | --- | --- |
| **T** | **Action** | **System** |
| T-1 | 구매자(Buyer)가 웹 페이지에서 구매하고자 하는 전력량을 정한 후 ‘구매 신청’ 버튼을 누른다. | **purchaseReq(buyer, transaction, reqAmount)**을 호출하여 다음을 진행한다.  1. **buyerValidationTest(buyer, reqAmount)** 호출한다.  **True면**다음 과정 시행, **False**면 대안상황으로 이동한다.  **(user\_battery\_max – user\_elec\_charge > reqAmount)**  2. **transaction\_status = 1**로 변경하여. ‘구매 신청’ 상태로 만든다.  3. **transaction\_buyer\_id**에 구매자의 **user\_id**를 입력한다.  4. **buyer\_id**, 현재시간, **reqAmount** 등을 이용하여 해시값을 생성하고 **transaction\_hash**에 저장한다.  5. **sendEmail(seller, transaction)**호출하여 T-2를 이행한다. |
| T-2 | 웹 서버(Web Server)에서 판매자(Seller)에게 구매 요청 이메일을 전송한다. | **sendEmail(user, transaction)**을 호출하여 다음 중 하나를 진행한다.  1. user가 판매자(**seller**)일 경우,  **transaction**의**transaction\_seller\_id == seller**의**user\_id**  가 **True**이면 다음 이메일을 판매자에게 전송한다.  [“{**TransAction\_description**}에 구매 요청 되었습니다. 아래. 해당 거래를 아래 버튼을 통해 승인 또는 거절을 해 주시기 바랍니다.”]  2.user가 구매자(**buyer**)일 경우,  **transaction**의**transaction\_seller\_id == seller**의**user\_id**  가 **True**이면 다음 이메일을 판매자에게 전송한다.  [“{**transaction\_description**}의 거래가 수락되었습니다. 해당 거래를 아래 버튼을 통해 최종 승인 또는 거절을 해 주시기 바랍니다.”] |
| T-3 | 이메일을 받은 판매자(Seller)가 웹 페이지에서 ‘거래 수락’ 버튼을 누른다. | **purchaseAccept(transaction\_id, hash)**을 호출하여 다음을 실행한다.  1. **transaction\_hash**에 저장된 값과 **hash**를 비교하여 올바른 요청인지 확인한다.  2. **sellerVallidationTest(seller, transaction)**을 호출 하여,  1) 판매자의 잔여 배터리 용량이 판매 전력량보다 큰지 확인한다.  (**user\_elec\_charge>reqAmount)**  2) 판매자의 초당 생산량이 초당 소모량보다 큰지 확인한다.  (**user\_elec\_supply>user\_elec\_usage)**  3. 1)과 2) 모두**True**이면 3을 진행하고 그렇지 않다면 대안상황으로 이동한다.  4. **transaction\_status = 2**로 변경하여. ‘거래 수락’ 상태로. 만든다.  5. **buyer\_id**, 현재시간, **reqAmount** 등을 이용하여 해시값을 생성하고 **transaction\_hash**에 저장한다.  6. **sendEmail(buyer, transaction)**호출하여 T-2를 이행한다. |
| T-4 | 최종 승인 메일을 받은 구매자(Buyer)가 거래에 대한 최종 승인 버튼을 클릭한다. | **finalAccept(transaction\_id, hash)**을 호출하여 다음을 실행한다.  1. 구매자에게 **buyerValidationTest(…)**을,  판매자에게 **sellerValidationTest(…)**을 실행한다.  2. **buyerValidationTest(…)**에서 문제가 발생하였다면 대안 상황 S-1을 실행한다.  3. **sellerValidationTest(…)**에서 문제가 발생하였다면 대안 상황. S-2을 실행한다.  4. **transaction\_status = 3**로 변경하여. ‘거래 중’ 상태로 만든다. **sendEmail(transaction)**함수를 호출하여 구매자와 판매자에게 다음과 같은 이메일을 보낸다.  [“거래가 최종 승인되어 진행되고 있습니다.아래 링크로. 접속하여 거래 상태를 확인해 주세요.”]  5. **checkTimeLimit(timeLimit = 60)**을 호출하여 제한 시간 내(60분)에 승인이. 되지 않을 시에 대안 상황 S-3을 실행한다. |
| T-5 | 웹 서버에서 릴레이 개방 시그널을 전송한다. | 1. **isChannelOpen= True**로 바꾼다.  2. **relayOpenReq(buyer,seller)**을 호출하여 릴레이 채널 개방 시그널을(**openRelay**) HTTP프로토콜을 이용하여 JSON 형태로 전송한다. |

**\* 대안 흐름**

|  |  |  |
| --- | --- | --- |
| S-1 | buyerVaildationTest(buyer, reqAmount)  : 구매자의 부족한 배터리 용량이 구매하고자 하는 전력량을 수용할 수 없는 경우 | 1. 웹 서버에서 **alertError(buyer)**을 호출하여 구매자에게 다음과 같은 팝업 알림을 띄운다.  [“구매 불가 : 충전 가능한 배터리 용량이 구매하고자 하는 전력량 보다 적습니다.”]  2. **transaction\_status = 0**으로 바꾼 뒤 거래 프로세스를 종료한다. |
| S-2 | sellerValidationTest(seller, transaction)  1. 판매자의 배터리 충전량이 판매 전력량보다 적을 경우  2. 판매자의 초당 생산량이 초당 소모량보다 적을 경우 | 1-1). 웹 서버에서 **alertError(seller)**를 호출하여 판매자에게 다음과 같은 팝업 알림을 띄운다.  [“판매 불가 : 배터리의 잔량이 판매 전력량보다 적습니다.”]  1-2). **transaction\_status = 0**으로 바꾼 뒤 거래 프로세스를 종료한다.  2-1). 웹 서버에서 **alertError(seller)**를 호출하여 판매자에게 다음과 같은 팝업 알림을 띄운다.  [“판매 불가 : 전력 생산량이 총 소비량보다 적습니다.”]  2-2) **transaction\_status = 0**으로 바꾼 뒤 거래 프로세스를 종료한다. |
| S-3 | timeLimitError(buyer)  : 구매자가 제한시간(60분)이내로 최종 승인에 대한 응답이 없을 경우 | 1.**sendErrorEmail(seller,buyer);**를호출하여 구매자와 판매자에게 다음과 같은 이메일을 전송한다.  [“구매자가 제한시간 내에 현재 거래에 대한 최종승인을 하지 않아 자동으로 거래가 취소되었습니다. ”]  2.**transaction\_status = 0**으로 바꾼 뒤 거래 프로세스를 종료한다. |

**\*사전/사후 조건**

**- 사전 조건**

1. 로그인
2. 판매글등록

**- 사후 조건**

1. 거래액 반영
2. 거래자 및 판매자 거래 유효성 검사(실제로 목표한 송전량이 송전됐는지, 수신됐는지 확인
3. 거래 내용을DB에 저장

**③ 클래스 다이어그램 및 데이터 맵**

**1) 클래스 다이어그램**

|  |  |
| --- | --- |
| **클래스** | **설명** |
| Main | 거래 중계를 위한 클래스 |
| Web Server | DB와 Client를 연결하기 위한 클래스 |
| Seller | 판매자의 정보를 관리하는 클래스 |
| Buyer | 구매자의 정보를 관리하는 클래스 |

**a) Main**

1) 개요

Main 클래스에서는 Transaction Object, User Object에서 각 정보를 얻어와 릴레이를 컨트롤하는 아두이노와 통신을 위한 함수 **relayOpenReq(buyer\_id, seller\_id)**를 사용해 구매자와 판매자간의 거래를 진행한다. 그리고 구매자가 오랫동안 거래 승인을 하지 않는 경우를 위해 **checkTimeLimit(timeLimit)**함수를 두어 지정해둔 **timeLimit**이 넘어가게 되면 자동으로 거래를 취소시키는 **timeLimitError(buyer\_id)**함수를 사용하여 거래를 원활하게 한다

2) 연산

|  |  |  |  |
| --- | --- | --- | --- |
| **연산** | **파라미터** | **반환 값** | **설명** |
| relayOpenReq(  buyer\_id,  seller\_id  ) | INT, INT | Boolean | 구매자와 판매자의 거래를 위한 릴레이 채널을 열어주는 함수 |
| checkTimeLimit(  timeLimit  ) | INT | Boolean | 지정해둔 timeLimit을 넘겼는지 확인하는 함수 |
| timeLimitError(  buyer\_id  ) | INT | Boolean | timeLimit을 넘겼다면 현재 transaction에 대한 거래를 종료시키는 함수 |

**b) Transaction**

1) 개요

buyer의 구매 요청에 대한 buyer의 유효성 검사를 진행하고, buyer가 거래를 진행할 수 있는 상태인지를 확인한다. buyer에 대한 검사를 진행한 후 seller가 거래를 진행할 수 있는 상태인지를 확인한다. 앞의 과정에서 buyer or seller가 거래를 진행할 수 없는 상태가 된다면 거래를 진행할 수 없는 User에게 거래 불가 alert창을 띠운다. 거래를 진행하는 경우 거래 수락 or 구매 요청에 대한 확인 Email을 보내고, 최종 승인 Email을 보낸다. 이때 최종 승인 Email에 승인 응답이 없는 경우 거래 취소 Email을 보낸다.

2) 연산

|  |  |  |  |
| --- | --- | --- | --- |
| **연산** | **파라미터** | **반환 값** | **설명** |
| purchaseReq(  seller\_id,  transaction\_id,  reqAmount  ) | INT, INT, INT | x | buyer의 구매 요청에 대한 buyer의 유효성 검사 및 진행 |
| buyerValidationTest(  buyer\_id,  reqAmount  ) | INT, INT | Boolean | buyer가 사고자 하는 전력의 양이 buyer의 배터리의 총량보다 적거나 같은지를 확인 |
| sellerValidationTest(  seller\_id,  transaction\_id  ) | INT, INT | Boolean | 1. seller의 배터리 충전량이 판매 전력량보다 크거나 같은지 확인  2. seller의 초당 생산량이 초당 소모량보다 많은지 확인 |

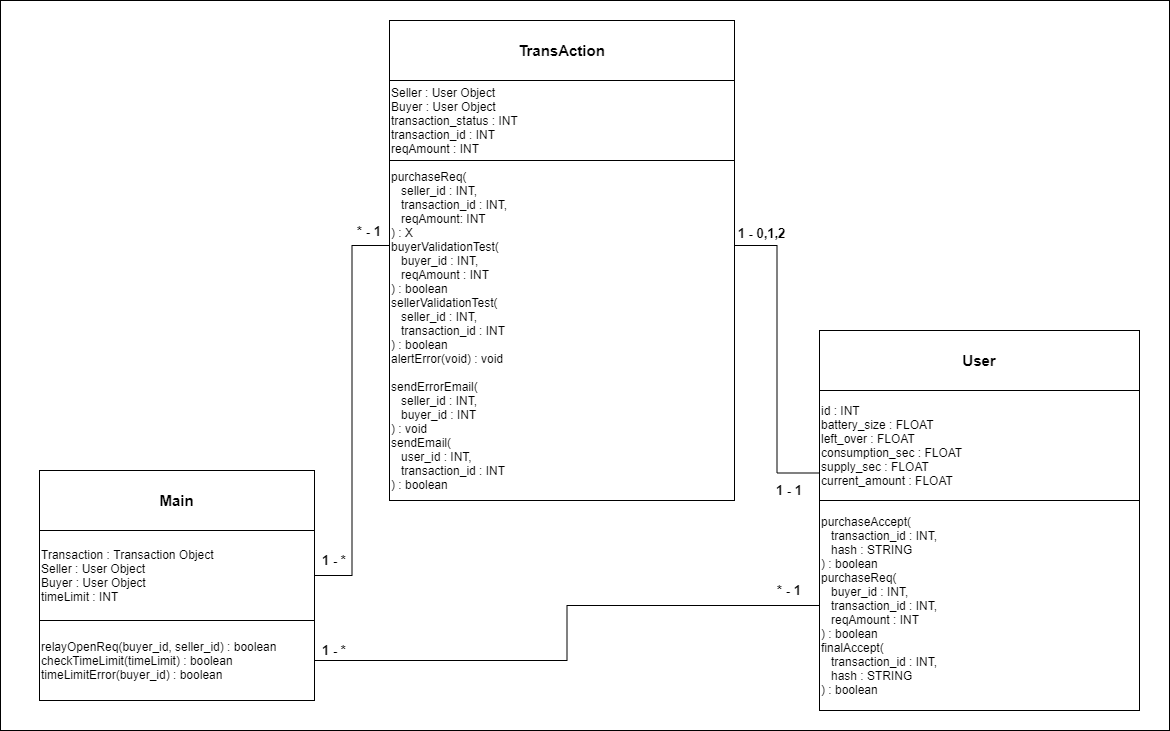
**c) User**

1) 개요

User 클래스에서는 구매를 원할 시, transActionReq메서드를 실행하여 거래 요청을 보내고, transActionApprove 메서드를 통하여 거래 수락을 하고,transActionExecute메서드를 통하여 최종 승인한다.

2) 연산

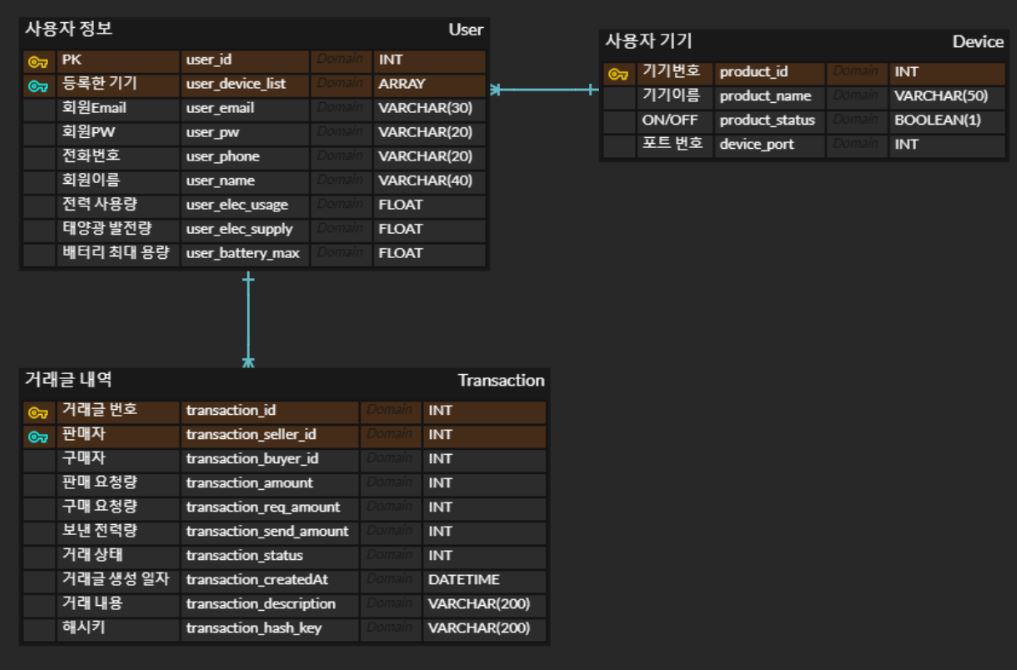
|  |  |  |  |
| --- | --- | --- | --- |
| **연산** | **파라미터** | **반환 값** | **설명** |
| purchaseReq(  buyer\_id,  transaction\_id,  reqAmount  ) | INT, INT, INT | Boolean | 거래 요청을 개시하는 함수, transaction 클래스 정보를 갱신 하고,  buyerValidation과sendEmail을 호출한다. |
| purchaseAccept(  transaction\_id,  hash  ) | INT, STRING | Boolean | 거래 수락시 시행되는 함수, sellerValidation 을 시행하고, transaction\_staus를 갱신한다. |
| finalAccept(  transaction\_id,  hash  ) | INT, STRING | Boolean | 최종 승인 통보 시, 시행되는 함수, sendEmail, sellerValidation, buyerValidation을 호출하고, transaction\_status를 갱신한다. |

****

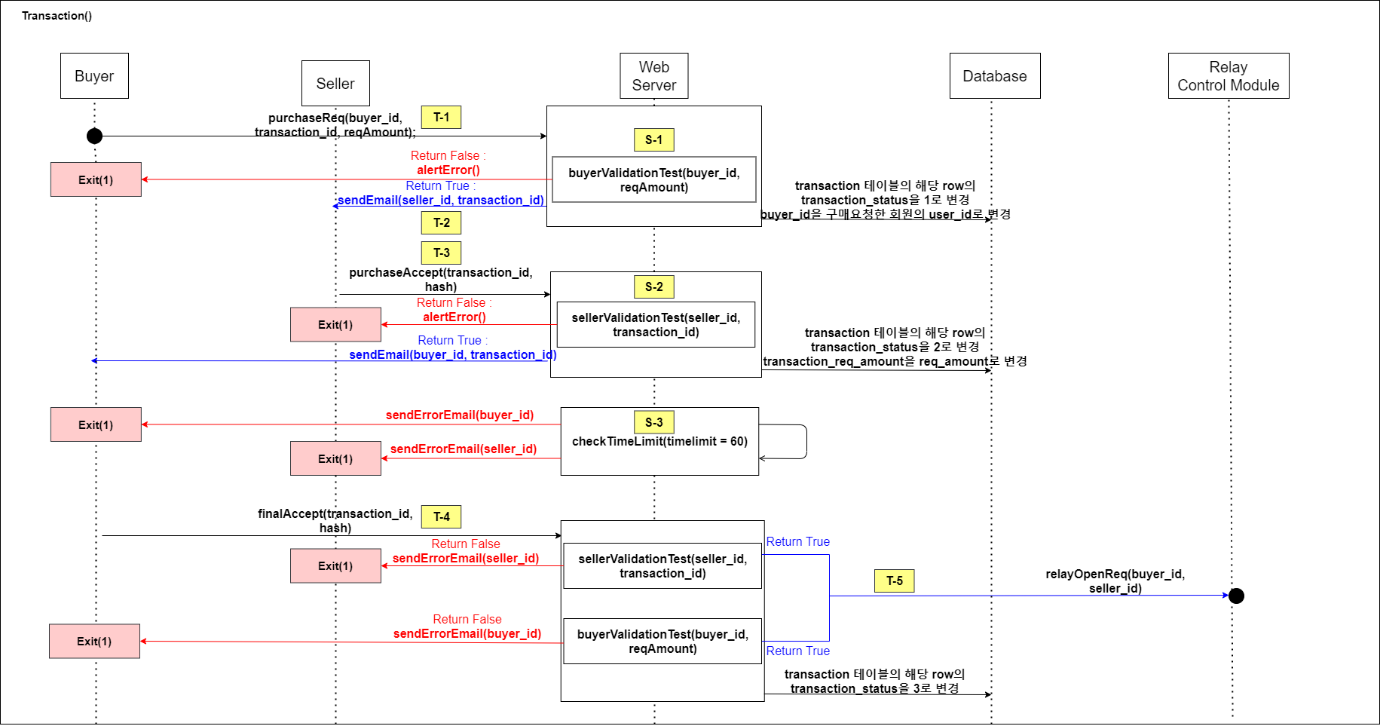
**2) 데이터 맵**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use case 명** | | | 전력 거래 | | |
| **Use case 설명** | | | 구매자와 판매자 사이의 전력 거래를 중개한다.. | | |
| **Data Name** | | | **Type** | **Size** | **Description** |
| transActionFrom  (Object) | | ID | Int | 20 | 거래 ID |
| buyer | Int | 20 | 게시글 생성자 ID |
| seller | Int | 20 | 거래 신청자 ID |
| amount | Int | 8 | 총 거래량 |
| reqAmount | Int | 8 | 구매 요청량 |
| CreateTStamp | DateTime | 4 | 거래 게시 시점 |
| endTStamp | DateTime | 4 | 거래 종료 시점 |
| startTStamp | DateTime | 4 | 거래 시작 시점 |
| User  (Object) | | ID | Int | 20 | 사용자ID |
| eCharge | Float | 8 | 현재 보유 전기량 |
| batteryMax | Float | 8 | 배터리 용량 |
| leftover | Float | 8 | 잔여 배터리량 |
| Consumption\_SEC | Float | 8 | 초당 전력 소모량 |
| Supply\_SEC | Float | 8 | 초당 전력 생산량 |
| isChannelOpen | | | BOOLEAN | 1 | 릴레이 개폐 여부 |
| timeLimit | | | Int | 4 | 응답 대기 시간 |
| openRelay  (Object) | ID | | String | 20 | 거래 ID |
| buyerID | | String | 20 | 구매자 ID |
| sellerID | | String | 20 | 판매자 ID |
| amount | | Int | 4 | 전력 거래량 |
| targetGate | | Int | 4 | 개방할 릴레이 채널 포트 번호 |

**3) DB ER-다이어그램**



**④ 시퀀스 다이어그램**



**3. 테스트 명세서**

**1) 시험 조건 및 절차**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **시험 주제** | **시험 조건** | **시험 절차** |
| 1 | 판매 글 생성 | 1) 로그인 된 상태  2) 판매 글을 작성할 수 있는 게시판이 있어야 됨 | 가)판매 글 작성하기 버튼을 클릭한다.  나)판매량을 적으면 예상 가격이 자동으로 변경되는지 확인한다.  다)내용을 작성한 뒤 등록 버튼을 클릭한다.  라) DB에 판매 글 정보가 들어갔는지 확인한다.  마)거래소에 판매 글이 올라가 있는지 확인한다. |
| 2 | 구매 요청 | 1) 판매 게시글이 작성되어 있어야 됨  2) 현재 로그인 된 회원과 판매 글을 게시한 회원은 서로 달라야 한다. | 가)구매하고자 하는 전력량을 입력한다.  나)구매 요청 버튼을 클릭한다.  다) transaction\_status의 값이 1로 바뀌는지 확인한다.  라)판매자에게 이메일이 전송된다. |
| 3 | 판매/구매  [승인or거절]  이메일 | 1) 구매자가 구매 요청해야 함  2) DB에 각 회원의 이메일이 저장되어 있어야 한다.  3) 서버는 이메일을 전송 및 응답 받을 수 있어야 함 | 가)판매자가 구매 요청 메일을 확인하고 승인 버튼을 누른다.  나)구매자에게 승인 이메일이 전송되었는지 확인한다.  다)구매자가 최종 승인 이메일을 확인하고 승인 버튼을 누른다.  라)구매자와 판매자 모두 승인되었는지 확인한다.  마)구매자와 판매자에게 거래 진행 상태 확인 이메일을 보낸다.  바)이메일이 전송되었는지 확인하고, DB가 올바르게 수정되었는지 확인한다. |
| 4 | 판매자 유효성 검사 | 1) 판매자가 판매 승인 이메일에서 판매를 승인해야 한다. | 가)판매자가 이메일을 승인한다.  나)판매자가 조건에 부합하는지 유효성 검사를 실시한다.  다)유효성 검사가 올바르게 진행되는지 확인한다.  라)유효성 검사 조건에 부합하면,구매자에게 승인 이메일을 전송한다. |
| 5 | 구매자 유효성 검사 | 1) 판매자가 구매 요청에 대해 수락해야 한다.  2) 판매자가 유효성 검사에 통과해야 한다.  3) 구매자에게 이메일이 도착해야 한다. | 가)구매자가 이메일을 승인한다.  나) 구매자와 판매자에 대한 유효성 검사를 실시한다.  다)유효성 검사가 올바르게 진행되는지 확인한다.  라)구매자와 판매자 모두 유효성 검사 조건에 부합하면,구매자와 판매자에게 거래 진행 상태 확인 이메일을 전송한다. |
| 6 | 실시간 거래 확인 | 1) 판매자와 구매자에 대한 유효성 검사가 통과되어 거래가 진행되어야 한다. | 가)전력 거래 진행  나)내 거래 확인하기 버튼을 클릭한다.  다)현재 거래 진행 상태가 잘 반영 되어있는지 확인한다. |
| 7 | 전력 전송 | 1) transaction\_status= 3으로 변경 되어야 한다.  2) 각 로컬들이 동일한 릴레이 채널에 연결되어 있어야한다. | 가)판매자와 구매자의 승인을 통해 transaction\_status = 3 인 ‘거래 중’로 변경한다.  나)웹 서버와 Arduino의 통신으로 각 로컬이 연결되어 있는 릴레이 채널을 열어준다.  다)릴레이 채널이 제대로 열리는지 측정기르 통해 확인한다.  라)측정기로 알아낸 값들을 웹 서버로 전송한다.  마)전송된 값을 바탕으로 웹 서버에서 계산하여 DB에 저장한다.  바)계산된 값과 실제 오차를 확인한다. |

**2) 결과 및 예외처리**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **예상 결과** | **시험 결과** | **예외 처리** |
| 1 | 판매 글 등록 시 DB에 판매 글 정보가 저장 되고, 거래소에 판매 글이 등록 된다. | 예상 결과와 동일한 결과 | -판매 글 등록자의 배터리 잔량보다 많게 판매량을 작성하면 판매 글 등록을 거부한다.  -판매량이 0보다 같거나 작을 경우 판매 글 등록을 거부한다. |
| 2 | transaction\_status = 1로 변경되어 ‘구매 신청’ 상태가 되고, 판매자에게 이메일이 전송된다. | 예상 결과와 동일한 결과 | -올바르지 않은 입력이 구매 전력량에 입력되면 구매 요청을 거부한다.  -구매 전력량이 판매량보다 많거나 충전 가능한 용량을 넘어설 때,구매 요청을 거부하고,다음과 같은 팝업 알림을 띄운다.  [“구매 불가 : 충전 가능한 배터리 용량이 구매하고자 하는 전력량 보다 적습니다”]  -transaction\_status = 0인 ‘구매 신청 대기’로 변경하고,거래 프로세스를 종료한다. |
| 3 | 시험 조건이 달성되었을 때, 구매자와 판매자에게 거래 진행 상태 확인 이메일이 보내진다. | 예상 결과와 동일한 결과 | -구매자와 판매자의 제대로 이루어지지 않았을 때 거래가 진행되지 않게 한다.  -거절 버튼을 눌렀을 경우,거래 상대에게 거절 이메일을 보내고 transaction\_status = 0 인‘구매 신청 대기’로 변경한다. |
| 4 | 판매자가 유효성 검사의 조건에 부합한다면 구매자에게 승인 이메일을 전송한다. | 예상 결과와 동일한 결과 | -판매자의 배터리 충전량이 판매 전력량보다 적을 경우, 다음과 같은 팝업 알림을 띄운다.  [“판매 불가 : 배터리의 충전량이 판매 전력량보다 적습니다”]  -판매자의 초당 생산량이 초당 소모량보다 적을 경우,다음과 같은 팝업 알림을 띄운다.  [“판매 불가 : 전력 생산량이 총 소비량 보다 적습니다.”]  -transaction\_status = 0으로 변경하고 거래 프로세스를 종료한다. |
| 5 | 구매자와 판매자가 유효성 검사 조건에 부합한다면, 구매자와 판매자에게 거래 진행 상태 확인 이메일이 전송된다. | 예상 결과와 동일한 결과 | -구매자의 구매 전력량이 판매량보다 많거나 충전 가능한 용량을 넘어설 때,구매 요청을 거부하고,다음과 같은 팝업 알림을 띄운다.  [“구매 불가 : 충전 가능한 배터리 용량이 구매하고자 하는 전력량 보다 적습니다”]  -판매자의 배터리 충전량이 판매 전력량보다 적을 경우, 다음과 같은 팝업 알림을 띄운다.  [“판매 불가 : 배터리의 충전량이 판매 전력량보다 적습니다”]  -판매자의 초당 생산량이 초당 소모량보다 적을 경우,다음과 같은 팝업 알림을 띄운다.  [“판매 불가 : 전력 생산량이 총 소비량 보다 적습니다.”]  -transaction\_status = 0으로 변경하고 거래 프로세스를 종료한다. |
| 6 | 현재 거래 진행 상태가 반영되어 페이지를 Rendering한다. | 예상 결과와 동일한 결과 | - 거래가 진행되고 있지 않을 시,기본 페이지를 Rendering한다. |
| 7 | 로컬 간의 릴레이 채널이 열리며 전력 전송이 시작되고, 측정기를 통해 알아낸 값으로 웹 서버에서 계산하여 해당 값들을 DB에 저장한다. | 예상 결과와 동일한 결과 | - H/W상의 고장의 경우 웹서버에서 거래량이 10분간 늘지 않으면 거래를 종료한다. |

**4. 요구사항 대비 시스템 구현 내용**

|  |  |  |
| --- | --- | --- |
|  | **완성도** | **부족한 점** |
| 신재생 에너지 충전 제어 | **60%** | 신재생 에너지 충전을 제어하는 릴레이가 망가져 현재 제어가 되지 않고 있다. |
| 소비량, 배터리 잔량 측정 및 표시 | **100%** | **-** |
| 원격 ON/OFF 기능 | **95%** | 원격 ON/OFF를 실행하면 즉각적으로 반응하지 않고 1sec~2sec정도의 텀이 있다. (알고리즘 개선 필요) |
| P2P 전력 거래 | **80%** | - 판매 승인이나 최종 승인에 제한 시간을 설정하지 않았다. |

**5. 개발 추진 내역**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **개발 내용** | **개발 기간(2021.03.07 ~ 2021.06.13)** | | | | | | | | **계획** | | | | |  | | | |
| **실적** | | | | |  | | | |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | | **9** | **10** | **11** | **12** | | **13** | **14** | **15** |
| **1. 문서 작업** |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
| 제안서 작성 |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| 유스케이스 작성 |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
| 클래스,시퀀스 다이어그램 작성 |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
| 중간 보고서 작성 |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
| 테스트 절차서 작성 |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
| 최종 보고서 작성 |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  | | | | | | | | | | | | | | | | | |
| **2. Login 기능** |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
| User DB 생성 |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
| Login 기능 개발 |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
| 접근 권한 설정 |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
| Login페이지 UI/UX 개발 |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  | | | | | | | | | | | | | | | | | |
| **3. 전력 거래 기능** |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
| Transaction DB 생성 |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
| 게시판 추가,수정,삭제 기능 개발 |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
| 이메일 보내기 및 응답 처리 기능 개발 |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
| 유효성 검사 기능 개발 |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
| External Arduino와 HTTP통신으로 거래 통제 기능 개발 |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
| 전력 거래 게시판 페이지 UI/UX 개발 |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  | | | | | | | | | | | | | | | | | |
| **4. 실시간 전력 내용 확인 기능** |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
| Local Arduino와 통신 |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
| 통신을 통한 측정값 내용 객체 전달, 표시 |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
| 실시간 전력 내용 확인 페이지 UI/UX 개발 |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  | | | | | | | | | | | | | | | | | |
| **5. 원격 제어 기능** |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
| Device DB 생성 |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
| 기기 추가,수정, 삭제 기능 |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
| Local Arduino와 HTTP 통신으로 기기 제어 신호 전송 |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
| 원격 제어 페이지 UI/UX 개발 |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  | | | | | | | | | | | | | | | | | |
| **6. 배포** |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
| Port Fowarding |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  | | | | | | | | | | | | | | | | | |
| **7. H/W 개발** |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
| 1 |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
| Local H/W조립 |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
| Local Arduino 원격 제어 기능 구현 |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
| Local Arduino 실시간 전력 내용 기능 구현 |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
| External H/W 조립 |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
| External Arduino 전력 거래 기능 구현 |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
| 선 정리 |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |

**6. 개발 프로그램**

**개요**

사용자가 등록한 기기에 한하여, 실시간으로 전력 데이터 (소비 전력, 생산 전력, 에너지, 잔량)등을 확인하고, 커뮤니티 내의 전력 거래를 가능케 하는 웹 플랫폼이다.

1. **전력 거래**

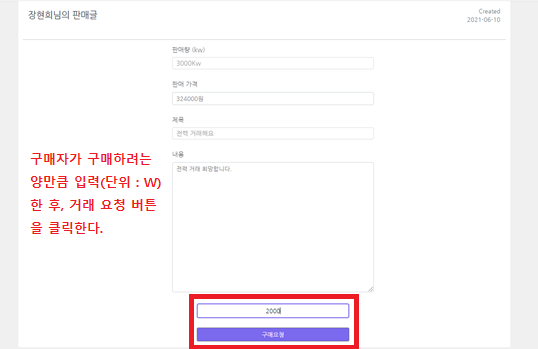
**1] 개요**

해당 플랫폼에 가입한 유저들에 한하여, 전력 거래를 진행한다. 거래의 신청은, 구매자가 판매자 글에 거래를 신청하는 방식으로 이루어진다. 구매자가 거래를 신청했을 경우, 판매자와 구매자에게 거래 승인에 관한 이메일이 전송되고, 구매자와 판매자 모두 수락 시, 거래가 진행된다.

**2] 작동 예시**



**[1-그림 1 :: 게시판 확인]**



[ 1-그림 2 :: 구매량 입력]

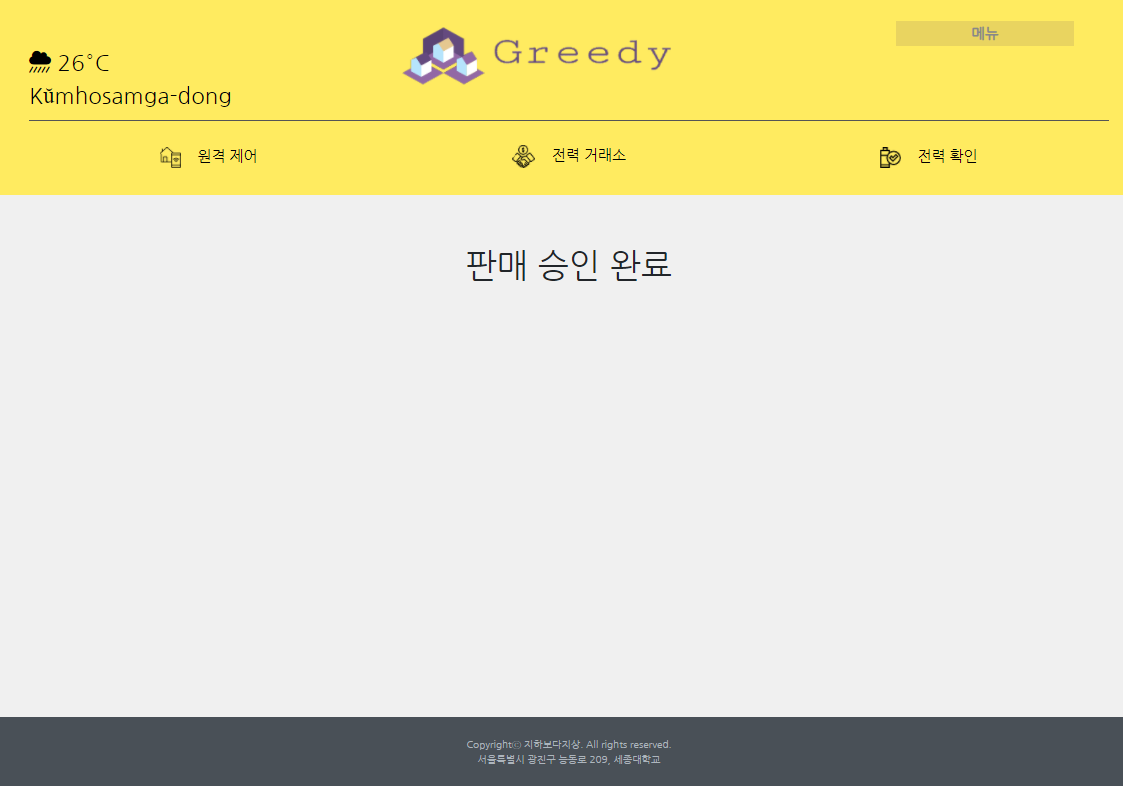
텍스트이(가) 표시된 사진

자동 생성된 설명

[ 1-그림 2-1 :: 구매 요청]



[ 1-그림 3 :: 판매자가 받는 판매 승인 E-mail]



[ 1-그림 3-1 :: 판매 승인 시]

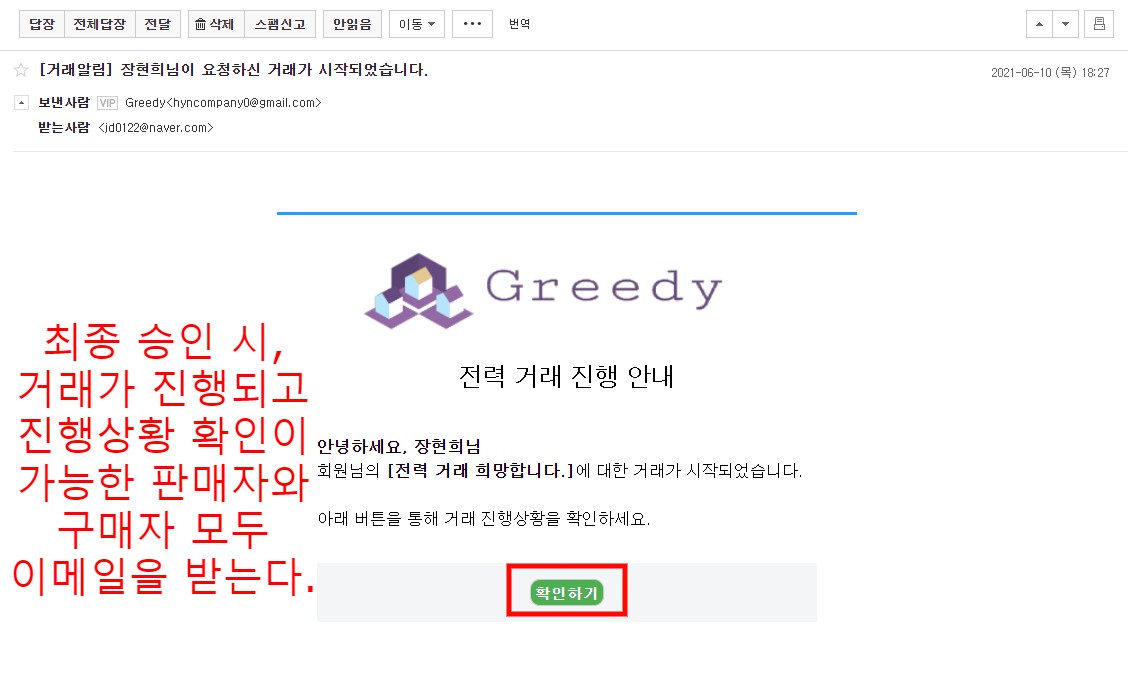


[ 1-그림 4 :: 최종 구매 승인에 관한 E-mail]

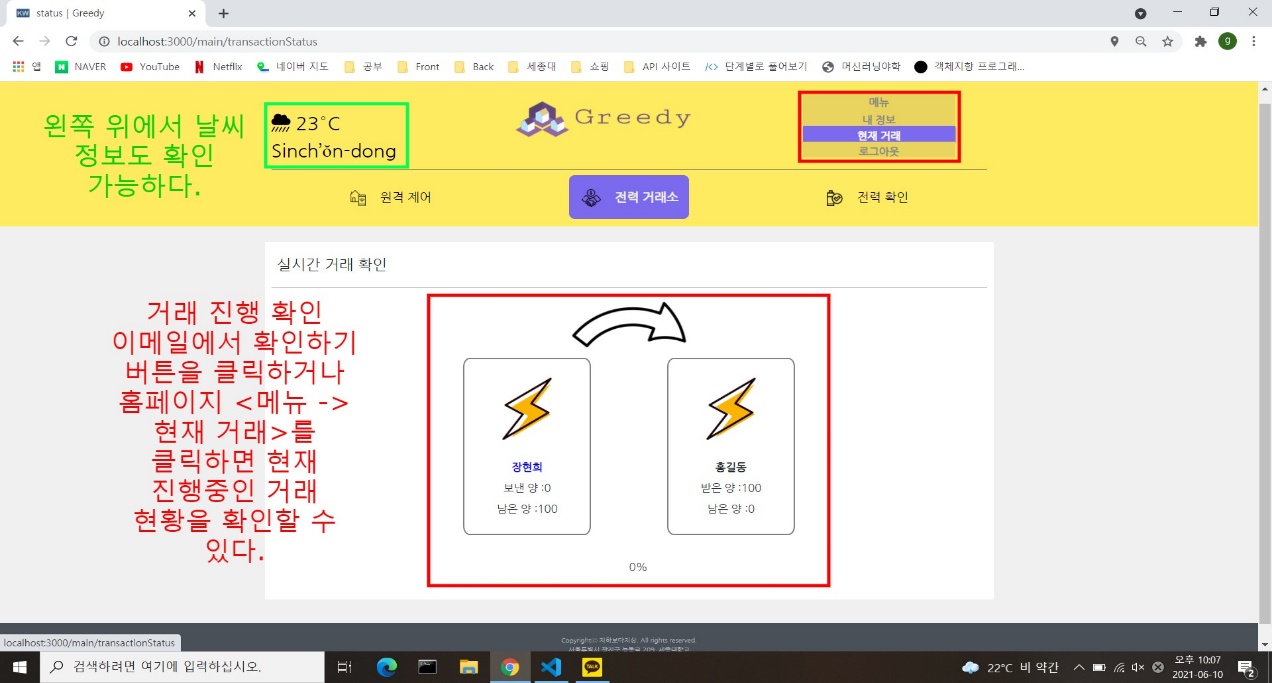
텍스트이(가) 표시된 사진

자동 생성된 설명

[ 1-그림 4-1 :: 최종 구매 승인시]



[ 1-그림 5 :: 전력 거래 시작 후 판매자와 구매자에게 가는 Email]



[ 1-그림 6 :: 실시간 거래 확인 페이지]

1. **실시간 전력 확인**

**1] 개요**

등록된 기기들에 한하여,실시간 에너지 소비량 , 에너지 생산량 , 배터리 잔량을 웹에 표기한다.각각 항목들의 단위는 mA/s, mA/s , % 이다.

**2] 작동 예시**



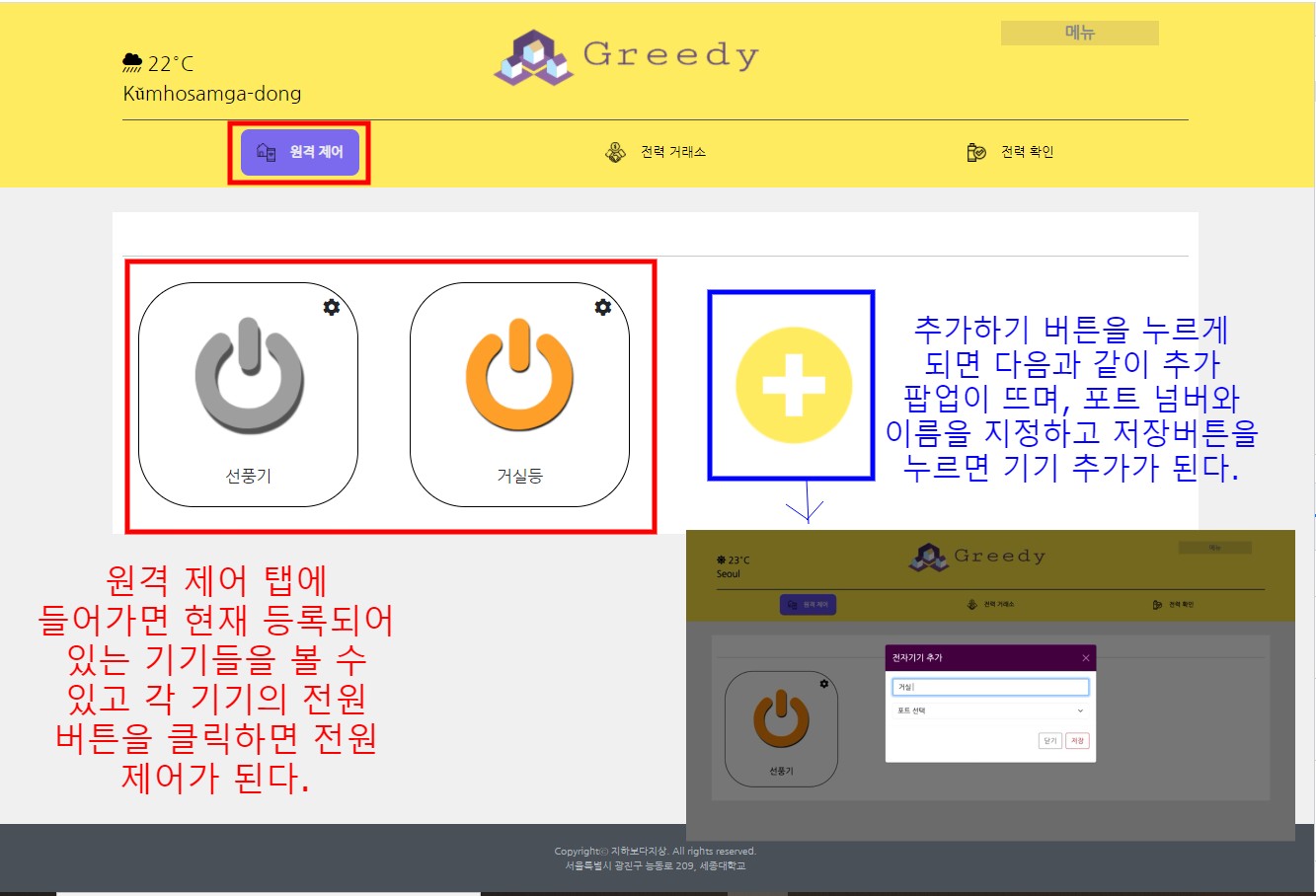
[ 2-그림 1 :: 실시간 전력 확인 ]

1. **원격 제어**

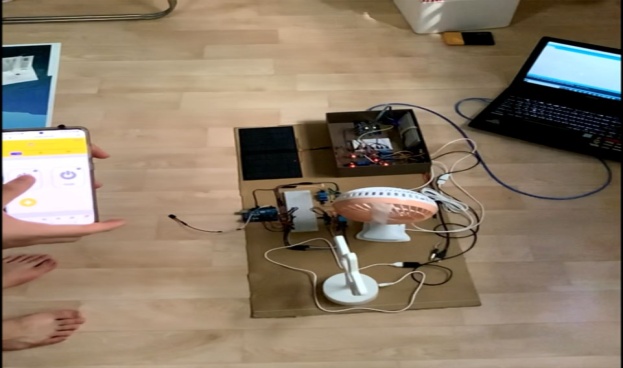
**1] 개요**

‘원격 제어’ 페이지에 생성된 기기 전원 버튼을 클릭함으로써,기기의 동작 상태를 제어할 수 있다.주황색으로 점멸 된 경우,현재 ‘사용 중’이란 상태(status = 1)이고, 회색일 경우, ‘사용 안함’ 상태(status = 0)이다.

**2] 작동 예시**



[ 3-그림 1 :: 원격 제어 페이지 ]

[ 3-그림 2 :: 전원 ON ] [ 3-그림 3 :: 전원 OFF ]

**7. 설계 구성 요소**

|  |  |  |
| --- | --- | --- |
| **설계 구성 요소** | **목표 설정** | 한국 전력 홈페이지를 통해 현재의 비즈니스 모델이 어떻게 구성되는지, 해외 스마트그리드 시스템의 사례를 통하여 장단점이 무엇인지 찾아보았다. 이를 바탕으로 Capstone 수업 내에서 금전적으로 불가능한 부분들을 감안하여 초기 모델 제작을 목표로 설정하였다. |
| **합성** | 회로의 구성, 배터리 잔량 계산, 예상 거래 시간 등을 계산하기 위해, 전기학 지식을 활용하였다. 단위 전력당 가격을 매기는 것을 위해 경제학적 관점으로 접근해 보는 것을 시도 해보았다. 전체적인 UI 색상 및 디자인을 위해 색채학의 도움을 받았다. |
| **분석** | 목표 설정 단계에서 수집한 여러 데이터들에 대해 현 상황에서 불가능 할 것 같은 부분들은 배제하고, Capstone Design수업 내에서 얻은 피드백을 통하여, 4가지 요구 사항을 구성하여 각 항목들에 대한 하위 레벨로 분할하여 문제 해결의 실마리를 잡았다 |
| **제작** | Agile 방식 중 하나인 Extreme Programming방식으로 프로젝트를 진행하였다. 크게 SW팀과 HW팀으로 나눈 후, 2명씩 배치하였으며, 프로젝트 진행 중 HW 부품 조달 과정에 문제가 발생하여, 초기에는 4인 모두 SW팀으로 배치하여, FrontEnd팀과 Backend팀으로 나눠 진행하였다. HW 수령 후, 원래 방식대로 진행하였다. 버전 관리는 Github를 이용하였다. |
| **시험** | SW작성 전 예상할 수 있는 여러 가지 예외 사항에 대한 대처방안을 구현 및 테스트 하여 전체적인 시스템 안정성을 높였다. |
| **평가** | 평가는 테스트 절차서 및 유스케이스를 참고하여 현재 구현된 항목들과 구현되지 않은 항목들을 직접 시연을 통하여 평가한다. |
| **제한 조건** | **산업 표준** | 스마트 그리드 시스템은 현재 개발 및 연구 중이므로, 표준으로 지정된 문서가 존재하지 않는다. |
| **경제성** | 해당 시스템은 사용자가 자신의 전력 상황을 빠르게 파악하고 관리할 수 있도록 돕고, 이를 통하여 거래할 수 있게 함으로 전력의 과소비를 줄이는 효과와 개개인의 전력을 거래함으로써 잉여 전력의 낭비를 줄이는 효과를 기대할 수 있다. |
| **안정성** | 해당 시스템이 각종 상황들(최대 보유 전력량 보다 많이 판매하는 경우, 올바르지 않은 입력, 올바르지 않은 경로 접근 등)에 대해 예외 처리함으로써, 안정성을 높였다. |
| **미학** | 전원 제어와 같은 버튼들을 직관적으로 이해할 수 있도록 아이콘을 활용하였다. 또한 배터리 잔량을 mAh나 Wh단위를 사용하지만, 직관적인 이해를 위해 ‘%’ 로 표기하였다. |
| **사회 영향** | 사이버 윤리에 대해 저촉할 가능성이 있는 부분은 지속적인 관리를 통해 해결할 예정이다. |

**8. 향후 개선 계획**

**1) 신재생 에너지 충전 제어**

 - 릴레이 재구매를 통해, 기존에 진행할 수 없었던, 충전 제어 기능을 완성한다.

**2) 가격 산출 알고리즘 정의**

- 현재 우리나라에서 사용하고 있는 누진세 방식을 스마트그리드 시스템에서의 가격 측정 방식으로 사용하는 건, 규모의 경제의 차이에 있어서 부적합하다.

- 생산량, 수요량 등과 같은 데이터를 지속적으로 조사하여, 스마트 그리드 시스템에 맞는 가격 산출 알고리즘을 정의한다.

**3) 기상청 정보의 특이사항 알림**

- 기상청 정보 API를 통해 유저에게 날씨 정보를 알려주었지만, 특정 날씨에 대한 알림을 추가하여 사용자가 더 빠르게 기상 정보를 접할 수 있도록 한다.

**4) 기기 별 전력 사용량 정보 제공**

- 현재 방식은 총 전력 사용량만 제공하고 있다.

- 기기 별 전력 사용량 정보 제공을 통하여, 사용자가 어떤 기기를 제어할지 의사 결정에 도움을 준다.

**9. 개발과정에서의 문제점**

|  |  |
| --- | --- |
| **문제점** | **해결** |
| **예산 및 공간 부족으로 태양광 패널이 전력 사용량에 비해 매우 부족함.** | **보조배터리를 완충한 상태에서 테스트를 진행함.** |
| **2개의 Wifi모듈 중 하나가 인식을 제대로 하지 못함.** | **외부 아두이노에 대한 통신 Test(원격 기기 제어)와**  **내부 아두이노에 대한 통신 Test(전력 거래)를**  **따로 진행하여 해결함.** |
| **케이블이 너무 난잡하여 개발 과정에서 혼란이 야기되어 개발이 늦어짐.** | **종이박스와 테이핑을 통해 선 정리를 하여 해결함.** |
| **전력 거래 시 문자를 사용하여 알림을 하고 싶었지만 예산이 부족함.** | **이메일(SMTP)을 사용하여 알림을 대체함.** |

**1) 서지상**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **학습**  **성과** | **수행준거** | **평가 기준** | **미달** | **기본** | **양호** | **우수** |
| **0점** | **1점** | **2점** | **3점** |
| PO3 | 컴퓨팅 분야의 문제를 정의하고 모델링할 수 있다. | - 보고서의 내용 및 결과가 과제의 목적과 부합하는가?  - 작성된 보고서의 품질이 우수한가? |  |  |  | O |
| PO4 | 컴퓨팅 분야의 문제를 해결하기 위해 최신 정보, 연구 결과, 프로그래밍 언어를 포함한 적절한 도구 등을 활용할 수 있다. | - 소프트웨어기술 및, 최신 기기 등을 효과적으로 활용하였는가? |  |  |  | O |
| PO5 | 사용자 요구사항과 현실적 제한조건을 고려하여 하드웨어 또는 소프트웨어 시스템을 설계할 수 있다. | - 문제를 해결하는 과정에서 발생하는 제한조건을 명확히 이해하고 이를 기반으로 프로젝트 계획 수립 및 수행을 하였는가? |  |  |  | O |
| PO6 | 컴퓨팅 분야의 문제를 해결하는 과정에서 팀 구성원으로서 팀 성과에 기여할 수 있다. | - 팀 과제 시 파트너십 향상을 위하여 수행한 활동 등이 적절하였는가?  - 팀 과제 수행이 원활하게 이루어 졌는가? |  |  |  | O |
| PO7 | 컴퓨터공학 전공자로서 다양한 환경에서 효과적으로 의사소통할 수 있다. | - 주어진 시간 내에 발표가 잘 이루어졌는가?  - 발표 내용이 잘 전달되었는가? |  |  |  | O |

**2) 장현희**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **학습**  **성과** | **수행준거** | **평가 기준** | **미달** | **기본** | **양호** | **우수** |
| **0점** | **1점** | **2점** | **3점** |
| PO3 | 컴퓨팅 분야의 문제를 정의하고 모델링할 수 있다. | - 보고서의 내용 및 결과가 과제의 목적과 부합하는가?  - 작성된 보고서의 품질이 우수한가? |  |  |  | O |
| PO4 | 컴퓨팅 분야의 문제를 해결하기 위해 최신 정보, 연구 결과, 프로그래밍 언어를 포함한 적절한 도구 등을 활용할 수 있다. | - 소프트웨어기술 및, 최신 기기 등을 효과적으로 활용하였는가? |  |  |  | O |
| PO5 | 사용자 요구사항과 현실적 제한조건을 고려하여 하드웨어 또는 소프트웨어 시스템을 설계할 수 있다. | - 문제를 해결하는 과정에서 발생하는 제한조건을 명확히 이해하고 이를 기반으로 프로젝트 계획 수립 및 수행을 하였는가? |  |  |  | O |
| PO6 | 컴퓨팅 분야의 문제를 해결하는 과정에서 팀 구성원으로서 팀 성과에 기여할 수 있다. | - 팀 과제 시 파트너십 향상을 위하여 수행한 활동 등이 적절하였는가?  - 팀 과제 수행이 원활하게 이루어 졌는가? |  |  |  | O |
| PO7 | 컴퓨터공학 전공자로서 다양한 환경에서 효과적으로 의사소통할 수 있다. | - 주어진 시간 내에 발표가 잘 이루어졌는가?  - 발표 내용이 잘 전달되었는가? |  |  |  | O |

**3) 박찬영**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **학습**  **성과** | **수행준거** | **평가 기준** | **미달** | **기본** | **양호** | **우수** |
| **0점** | **1점** | **2점** | **3점** |
| PO3 | 컴퓨팅 분야의 문제를 정의하고 모델링할 수 있다. | - 보고서의 내용 및 결과가 과제의 목적과 부합하는가?  - 작성된 보고서의 품질이 우수한가? |  |  |  | O |
| PO4 | 컴퓨팅 분야의 문제를 해결하기 위해 최신 정보, 연구 결과, 프로그래밍 언어를 포함한 적절한 도구 등을 활용할 수 있다. | - 소프트웨어기술 및, 최신 기기 등을 효과적으로 활용하였는가? |  |  |  | O |
| PO5 | 사용자 요구사항과 현실적 제한조건을 고려하여 하드웨어 또는 소프트웨어 시스템을 설계할 수 있다. | - 문제를 해결하는 과정에서 발생하는 제한조건을 명확히 이해하고 이를 기반으로 프로젝트 계획 수립 및 수행을 하였는가? |  |  |  | O |
| PO6 | 컴퓨팅 분야의 문제를 해결하는 과정에서 팀 구성원으로서 팀 성과에 기여할 수 있다. | - 팀 과제 시 파트너십 향상을 위하여 수행한 활동 등이 적절하였는가?  - 팀 과제 수행이 원활하게 이루어 졌는가? |  |  |  | O |
| PO7 | 컴퓨터공학 전공자로서 다양한 환경에서 효과적으로 의사소통할 수 있다. | - 주어진 시간 내에 발표가 잘 이루어졌는가?  - 발표 내용이 잘 전달되었는가? |  |  |  | O |

**4) 조성우**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **학습**  **성과** | **수행준거** | **평가 기준** | **미달** | **기본** | **양호** | **우수** |
| **0점** | **1점** | **2점** | **3점** |
| PO3 | 컴퓨팅 분야의 문제를 정의하고 모델링할 수 있다. | - 보고서의 내용 및 결과가 과제의 목적과 부합하는가?  - 작성된 보고서의 품질이 우수한가? |  |  |  | O |
| PO4 | 컴퓨팅 분야의 문제를 해결하기 위해 최신 정보, 연구 결과, 프로그래밍 언어를 포함한 적절한 도구 등을 활용할 수 있다. | - 소프트웨어기술 및, 최신 기기 등을 효과적으로 활용하였는가? |  |  |  | O |
| PO5 | 사용자 요구사항과 현실적 제한조건을 고려하여 하드웨어 또는 소프트웨어 시스템을 설계할 수 있다. | - 문제를 해결하는 과정에서 발생하는 제한조건을 명확히 이해하고 이를 기반으로 프로젝트 계획 수립 및 수행을 하였는가? |  |  |  | O |
| PO6 | 컴퓨팅 분야의 문제를 해결하는 과정에서 팀 구성원으로서 팀 성과에 기여할 수 있다. | - 팀 과제 시 파트너십 향상을 위하여 수행한 활동 등이 적절하였는가?  - 팀 과제 수행이 원활하게 이루어 졌는가? |  |  |  | O |
| PO7 | 컴퓨터공학 전공자로서 다양한 환경에서 효과적으로 의사소통할 수 있다. | - 주어진 시간 내에 발표가 잘 이루어졌는가?  - 발표 내용이 잘 전달되었는가? |  |  |  | O |