중간 보고서

16011009 컴퓨터공학과 서지상, 16011060 컴퓨터공학과 장현희

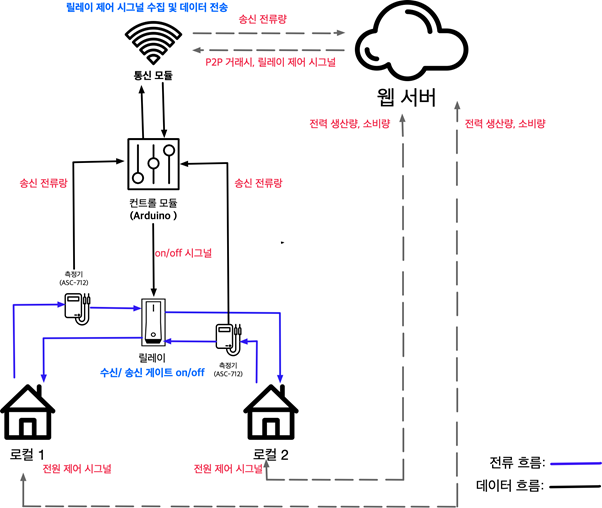
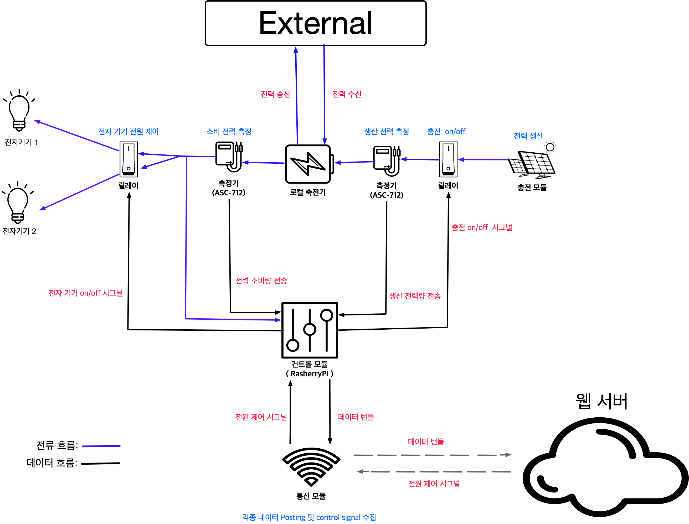
16011034 컴퓨터공학과 박찬영, 16010995 컴퓨터공학과 조성우

**개발 목표**

DC 스마트그리드는 현재 사용되고 있는 시스템의 AC-DC로 변환하는 과정 없이 DC 전력만을 사용하여 전력 손실을 줄이고자 개발되었다. 본 시스템은 각 가정에서 신재생 에너지를 생산하게 하여 공적 전기 수요량을 줄여 공적 전기를 생산할 때 발생되는 환경 오염을 줄이고자 하였다. DC 스마트그리드 방식의 전력망은 양방향 통신이 가능하므로 신재생 에너지를 생산하는 가정 간의 p2p 거래로 잉여 전력의 처리와 공적 전기 수요량 감소뿐만 아니라 경제적 이득을 보게 했다. 또한, 외부의 중앙 웹 서버와 각 가정 별 컨트롤 모듈들로 구성되며, 해당 웹 서버에 DB를 통해 데이터들을 저장 및 관리한다. 사용자는 모두 회원가입이 필수이며 각각 가정의 컨트롤 모듈을 독립적으로 가진다. 또한 사용자는 사용자가 설정한 기기들의 전원을 웹 서버를 통하여 원격제어가 가능하며 신재생 에너지의 전력 충전량과 총 사용중인 전력량, 배터리 잔량을 확인할 수 있다.

**시스템 구조**

**1. 구상도**



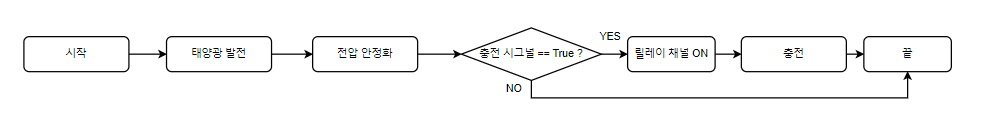
- 전체 구상도 - 로컬 구상도

**2. 상세 기능**

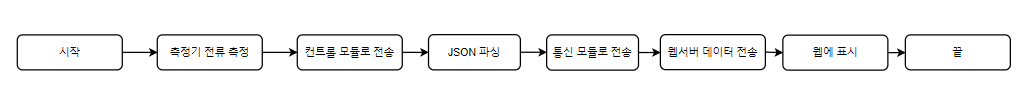
**1) 신재생 에너지 충전 제어**

- 전력 생산이 불안정한 에너지를 보완하기 위해 레귤레이터 사용

- 충전 모듈을 릴레이를 사용하여 컨트롤

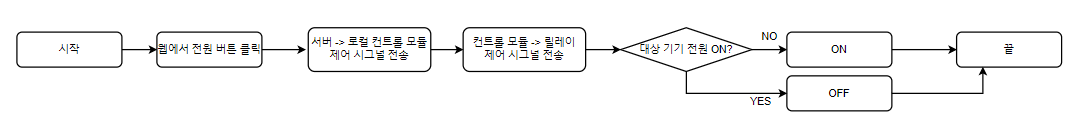
 **2) 소비량, 배터리 잔량 측정 및 표시**

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

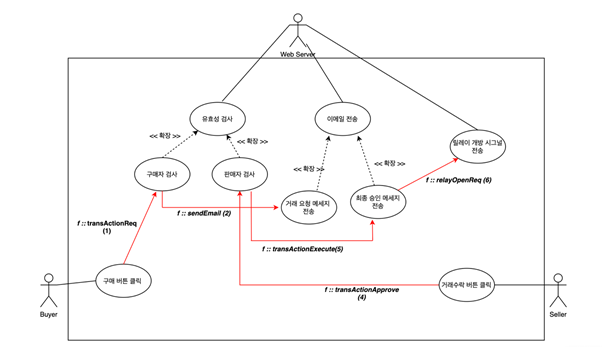


**3) 원격 ON/OFF 기능**

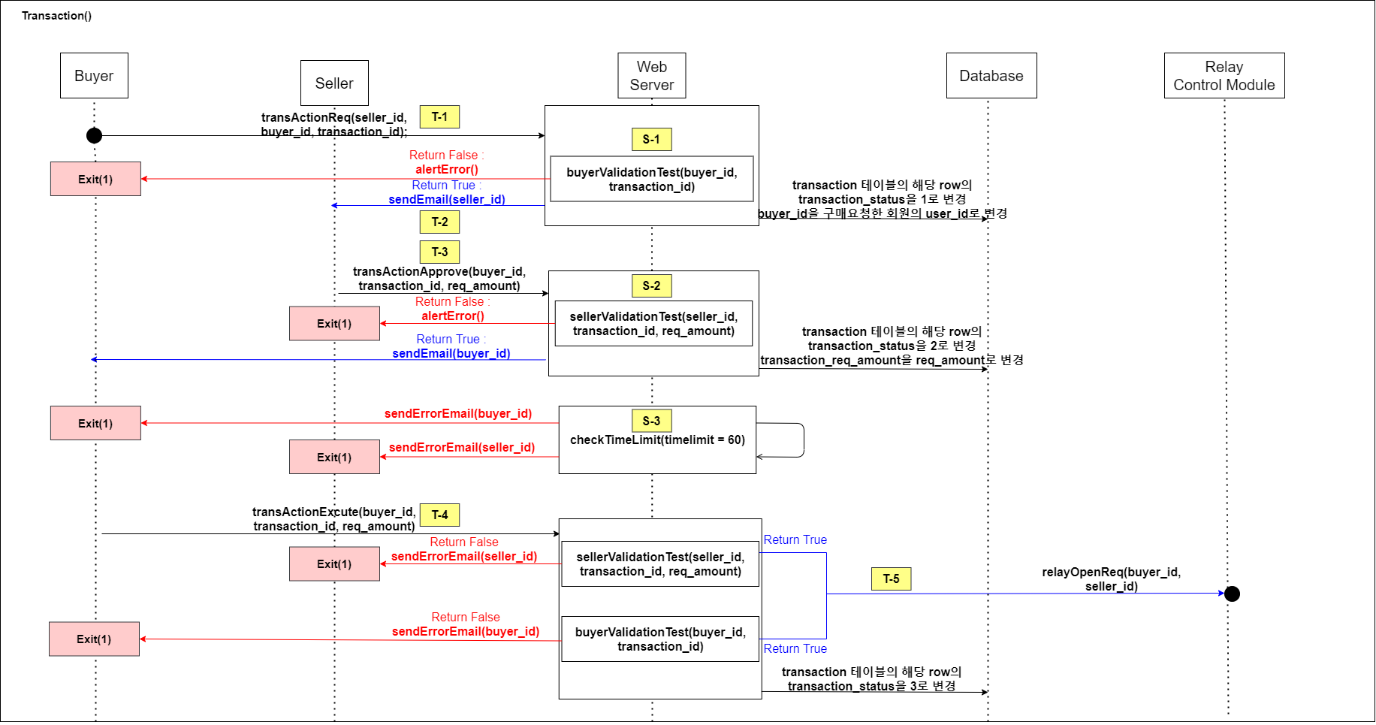
- 릴레이를 RBP에 연결하여 ON/OFF를 조정한다.

- 웹에서 ON/OFF를 선택하게 되면 RBP와 HTTP통신을 통해 릴레이를 조정한다. **4) P2P 전기 거래**

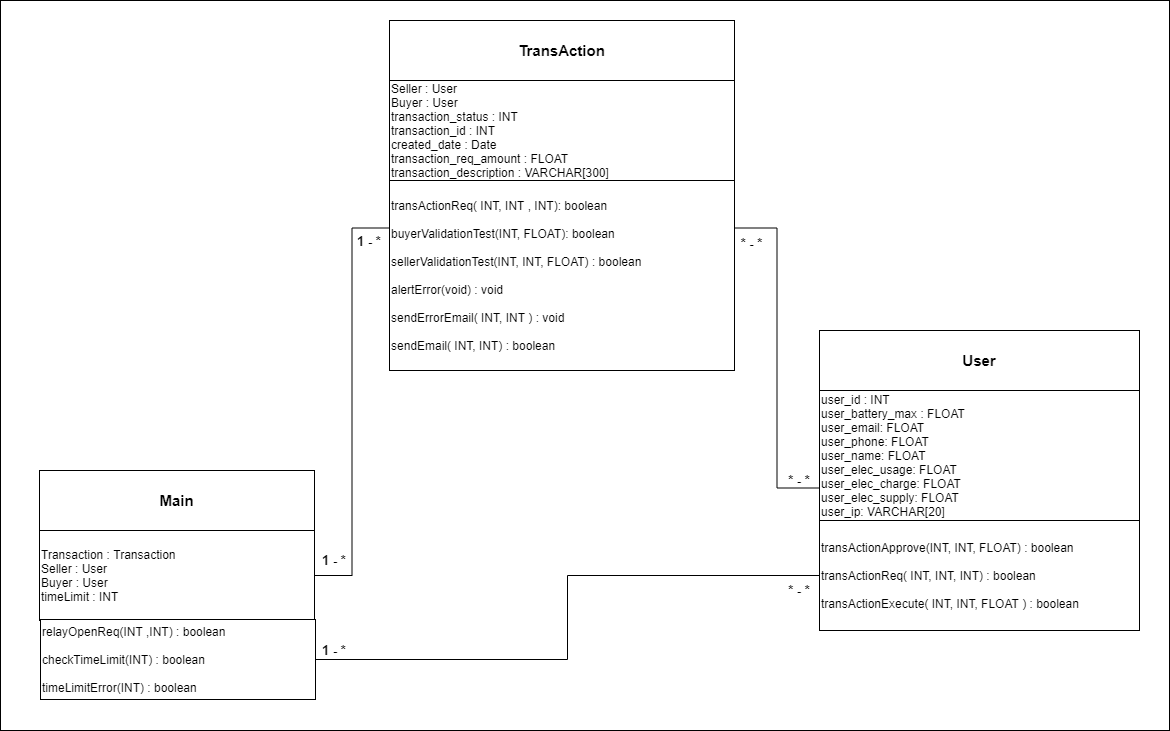
1. Usecase Diagram



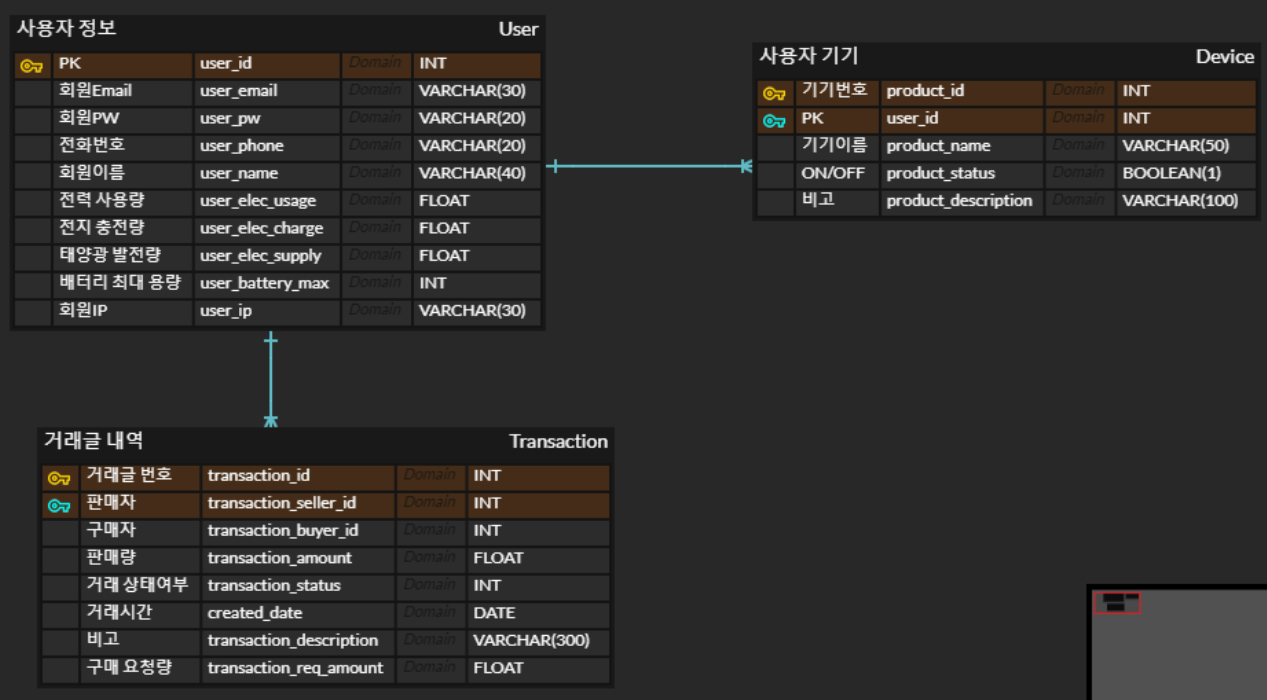
2. Sequence Diagram

****

3. Class Diagram

****

4. ER Diagram



- 등록된 판매 글에 구매자가 거래를 요청할 시, 구매자에 대한 유효성 검사가 시행되고, 통과 시, 판매자 에게 거래 요청에 대한 이메일이 전송된다. 판매자가 거래 수락을 하면, 판매자에 대한 유효성 검사가 시행된다. 통과 시, 웹 서버에서 구매자와 판매자 측에 최종 승인에 관한 메일을 전송한다. 제한시간 내에 구매자와 판매자 측 모두 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, 웹 서버에서 채널 폐쇄 신호가 전송되면, 채널을 닫는다.

**3. 팀원별 역할 분담**

- 공통 : 문서 작업

- 서지상 : BackEnd를 메인으로 담당, H/W를 서브로 담당

- 장현희 : FrontEnd를 메인으로 담당, H/W를 서브로 담당

- 박찬영 : H/W를 메인으로 담당, FrontEnd를 서브로 담당

- 조성우 : H/W를 메인으로 담당, BackEnd를 서브로 담당

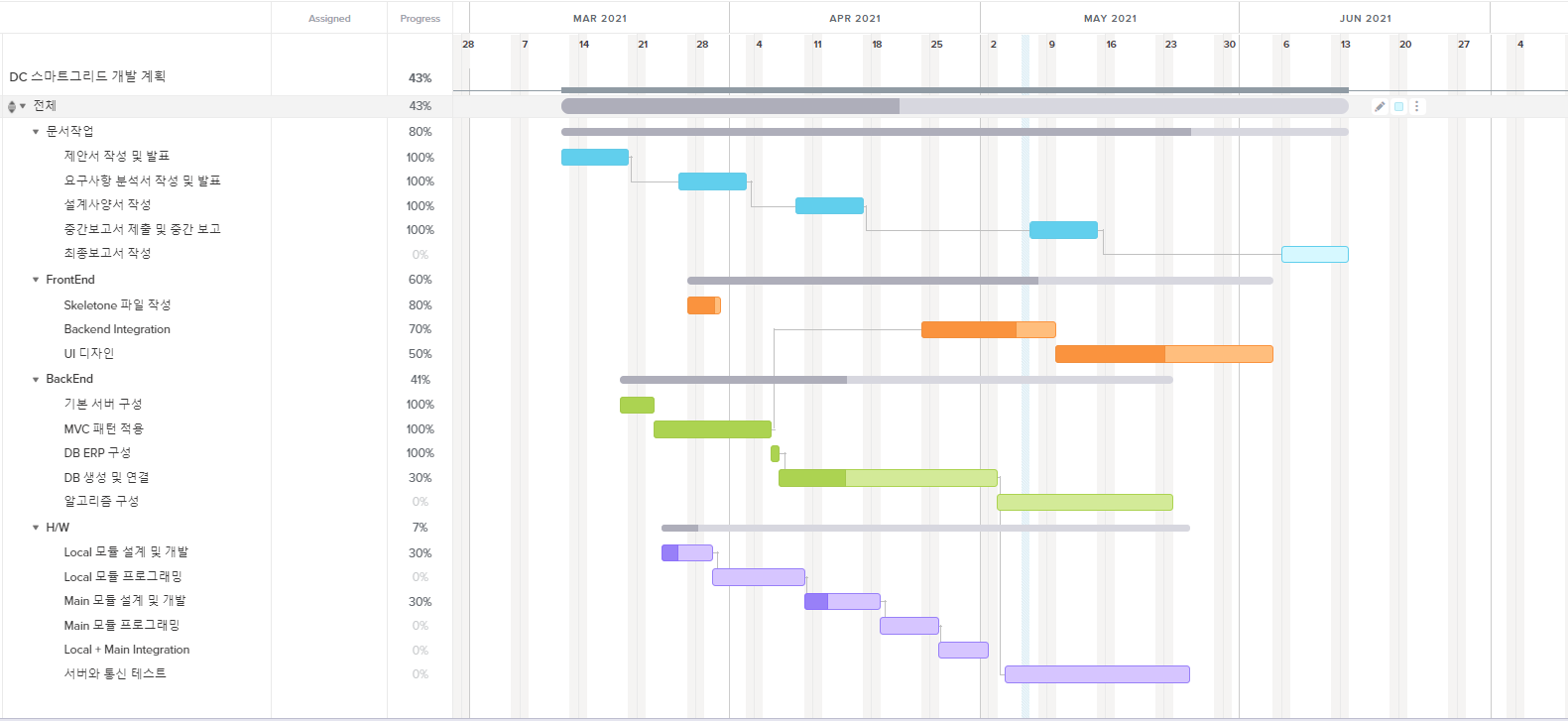
**제안서 대비 수정 내용**

**구매량 제한 변경**

수정 전) 판매자가 업로드한 전력량의 전량을 구매해야 하는 방식

수정 후) 구매자가 구매하고자 하는 전력량을 정하여 구매할 수 있는 방식

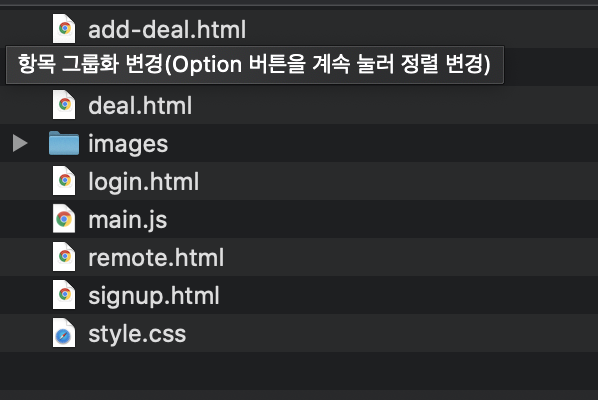
**개발 추진 계획**



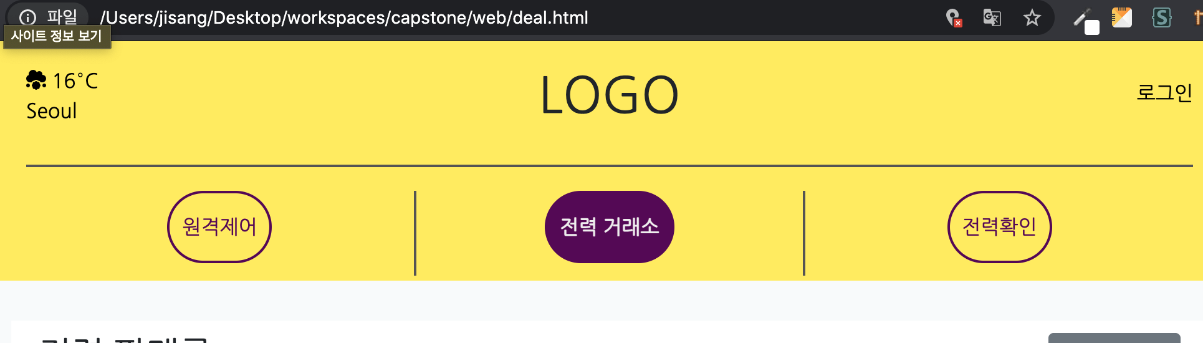
**현재 구현 내용**

**1. 뼈대 파일 작성 및 각각의 UI 기본 구상**

- 서버와 연결을 위한 기본 HTML 파일 생성



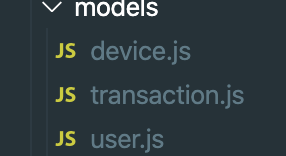
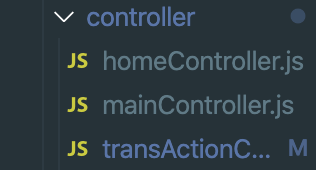
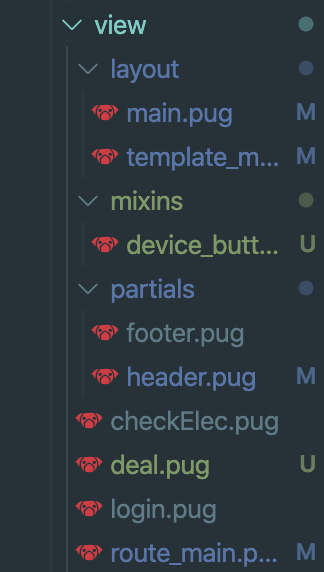
- 기본 HTML 파일



- HTML 기본 파일 UI 적용 화면

**2. MVC패턴을 적용한 기본 서버 구성**

- 서버 설계 방식으로 MVC(Model, View, Controller)패턴을 적용하여 디버깅에 용이하게 만듬

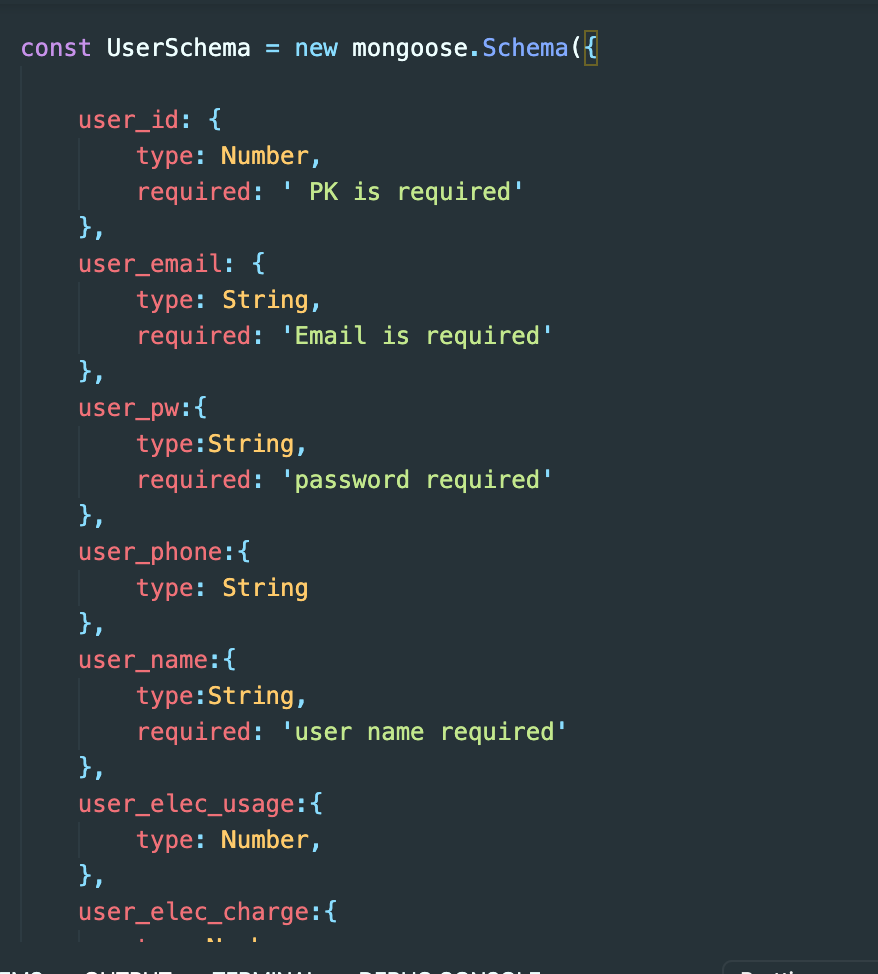
 

**3. DB 설계 및 스키마 생성**

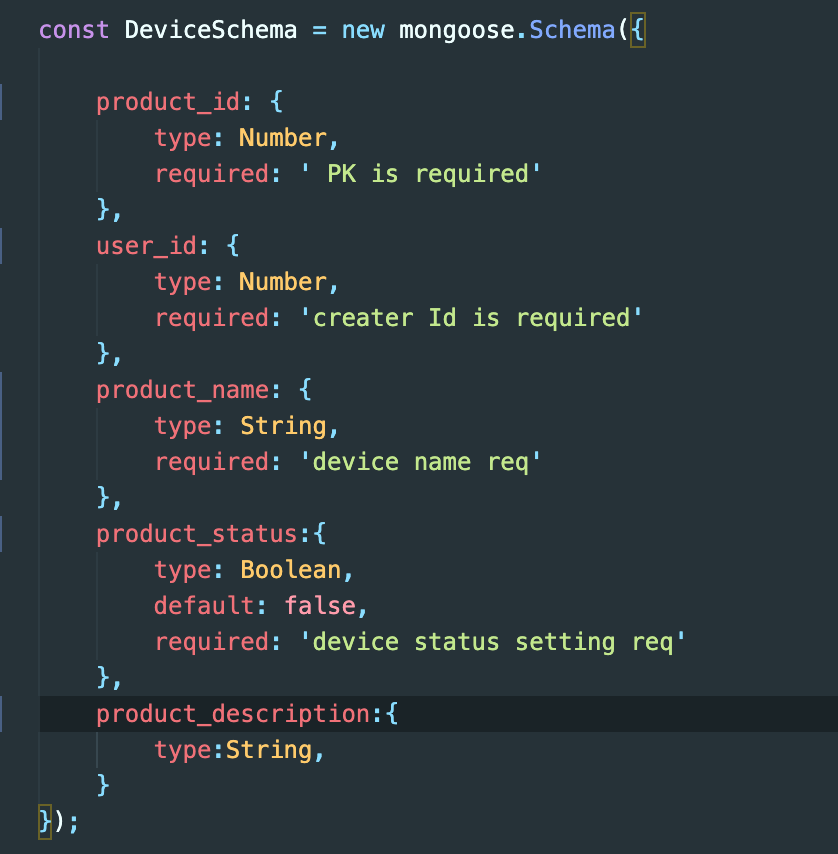
- ERD 구성도에 따라 DB 구성, 스키마 생성 및 연결



- Transaction Table 스키마



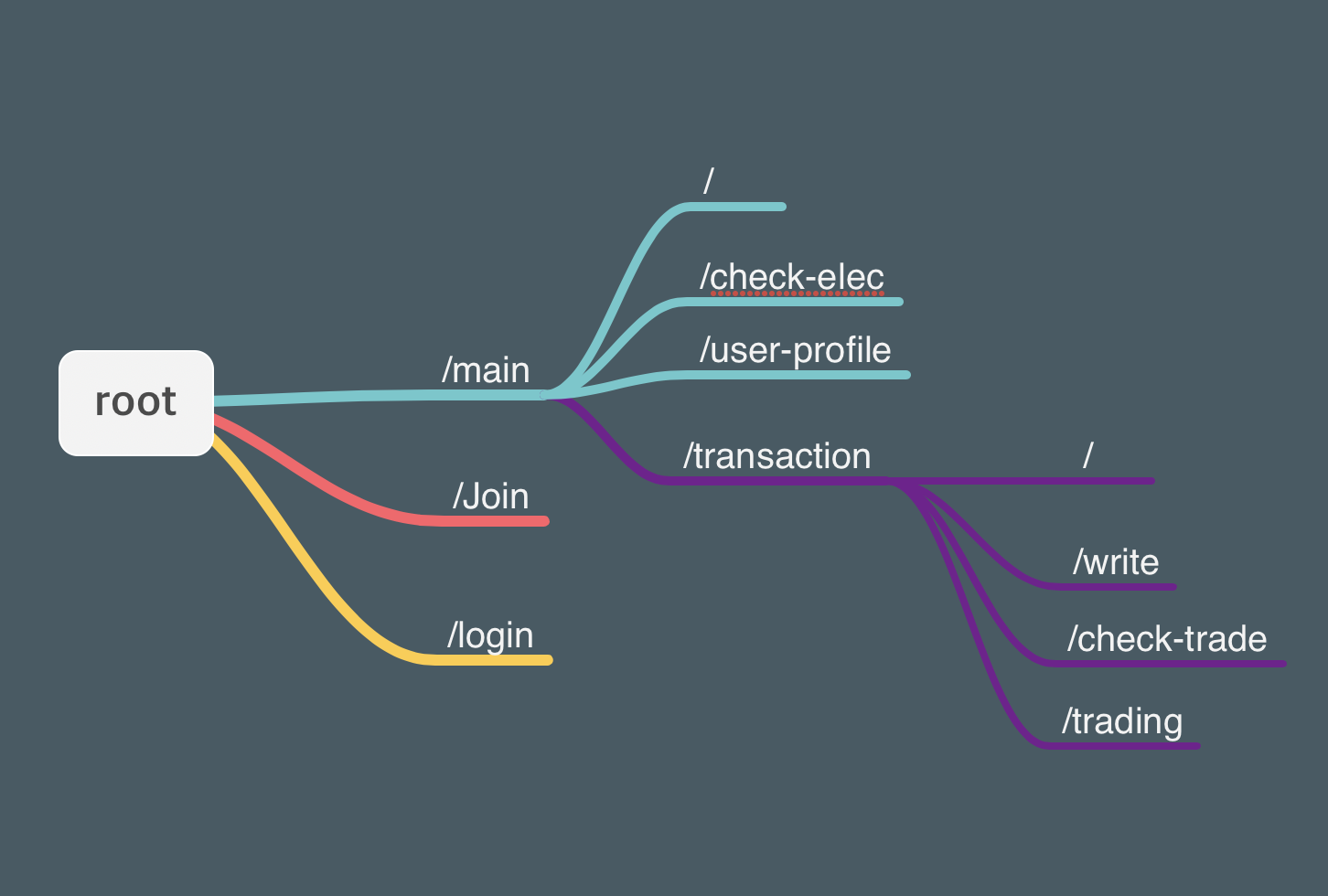
- User Table 스키마



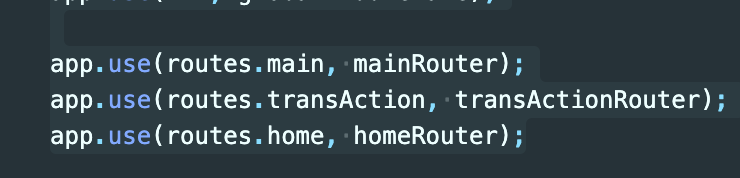
- Device Table 스키마

**4. Integration**

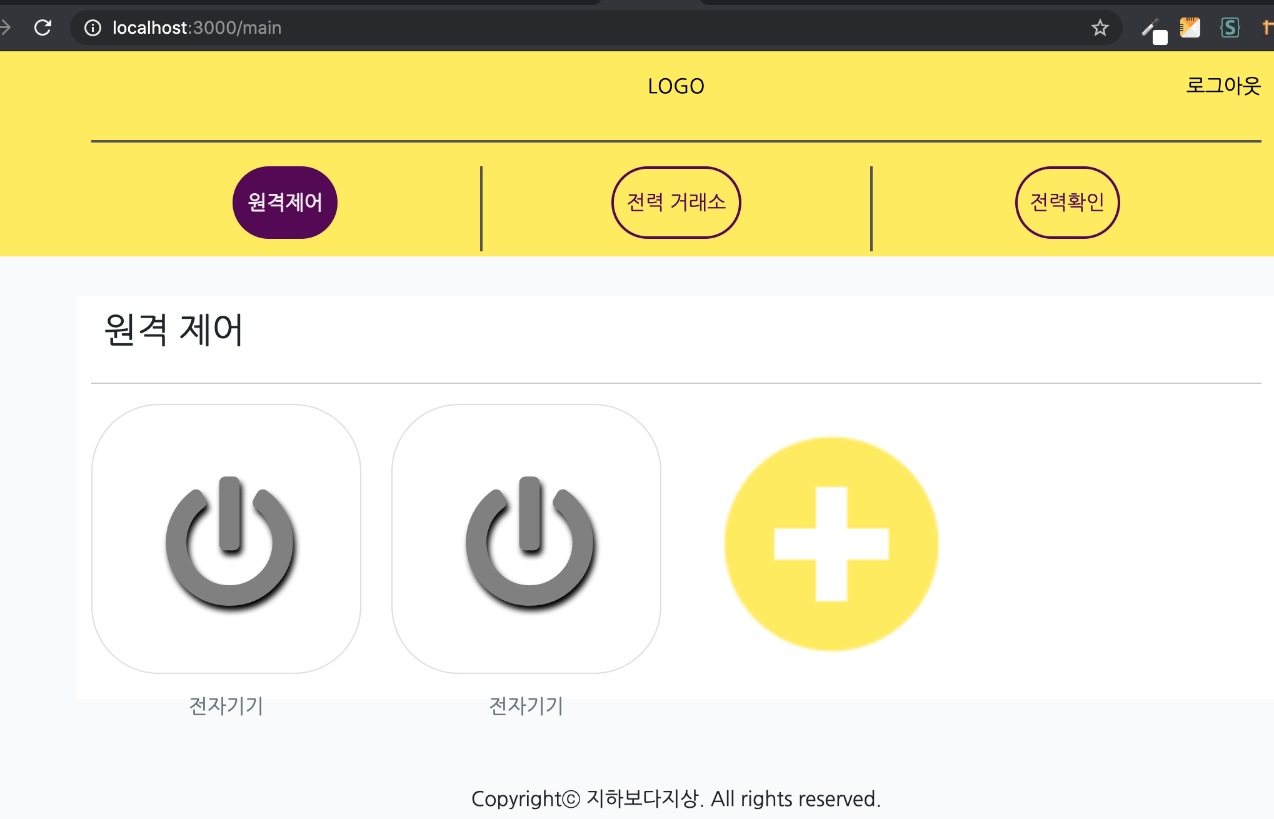
- 만들어 둔 기본 HTML 파일들을 토대로 서버와 통합

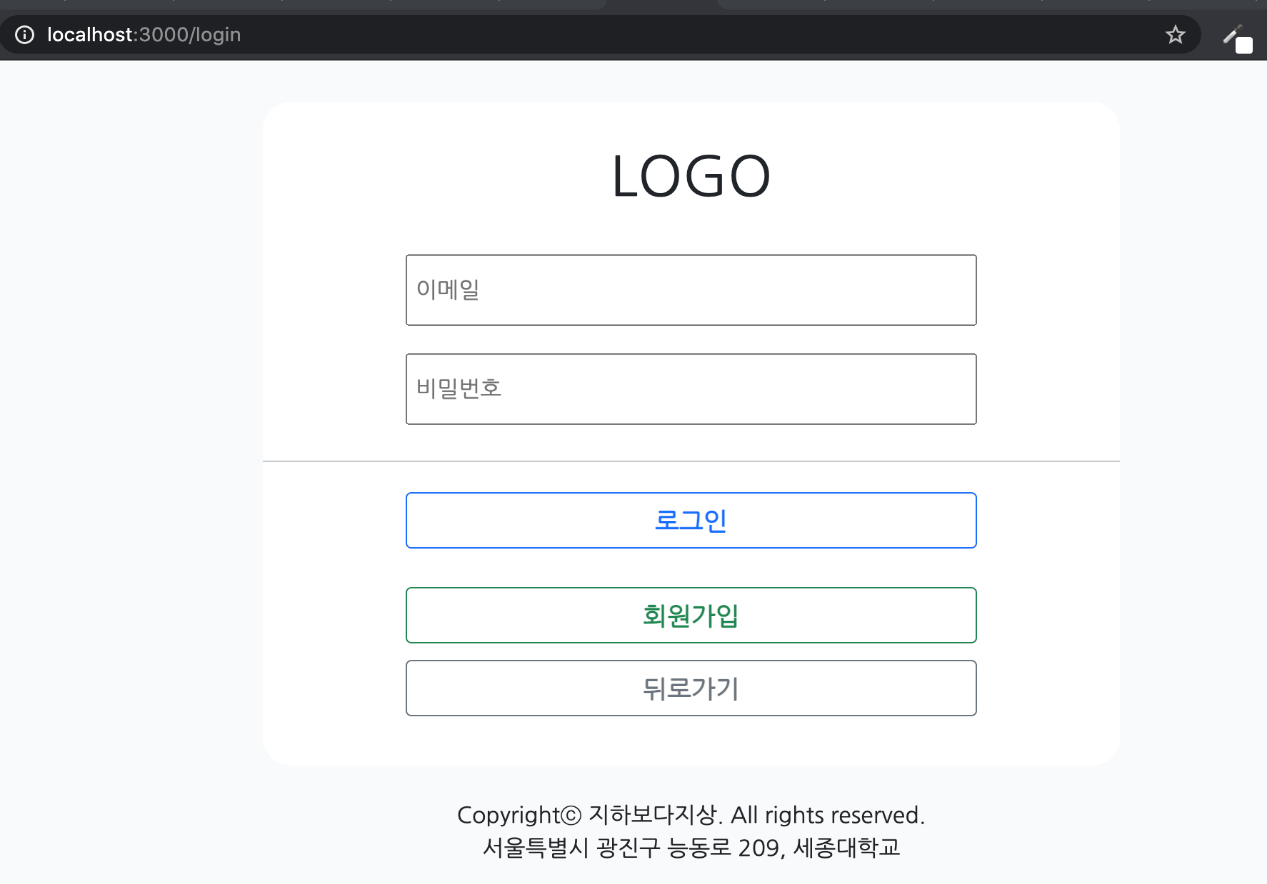
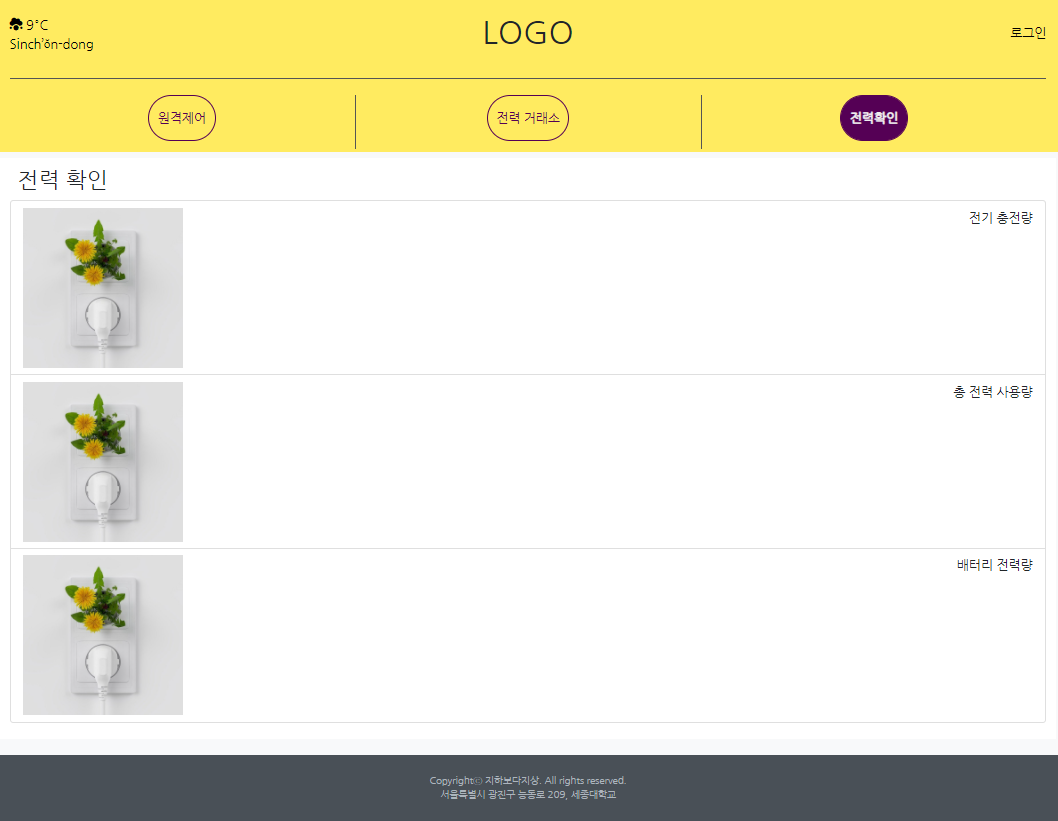


- 기본 HTML 파일 구성에 따라 뼈대가 되는 파일들을 각 라우트에 맞게 라우팅 함



- 라우팅 적용



- 각 라우터에 HTML 파일이 연결됨