**2020 캡스톤 디자인 I**

**중간 자문평가 중간보고서**

8조 눈치보이조

|  |  |
| --- | --- |
| **팀원** | 20143033 김도훈 (조장) |
| 20143030 곽지훈 |
| 20143037 김상원 |
| 20143056 명석현 |
| 20143116 홍령기 |
| 20163653 소가위 |

1. 어플리케이션

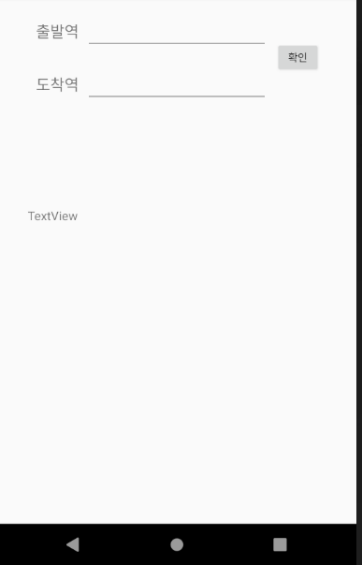
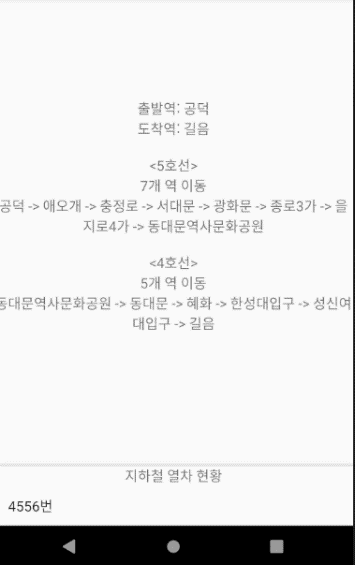
* 회원가입과 로그인, 임산부 인증이 구현 완료되었습니다.

현재는 이메일과 비밀번호를 이용한 회원가입만 지원하며 임산부 인증의 경우 임산부라면 모두 발급받아야하는 국민행복카드를 이용할 예정이나 프로토타입의 한계상 DB에 임의의 이름과 카드번호를 저장하여 그와 비교하는 방식으로 인증이 진행됩니다.

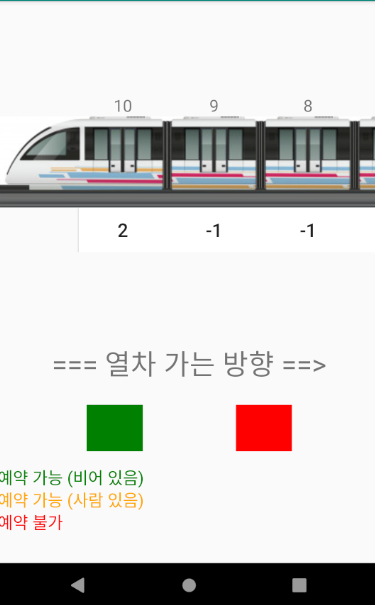
이후 노약자석에 대해서는 인증과정 없이 좌석현황만을 보여줄 수 있게 만들 계획입니다.

* 지하철 경로 찾기가 구현 완료되었습니다.

ODSay API의 지하철 경로검색 조회 레퍼런스를 사용하였으며 출발역과 도착역을 입력함으로써 경로가 출력됩니다. 이후 UI 업데이트 시 소요시간과 빠른 환승정보 등을 추가할 예정입니다.

* 좌석 예약이 부분 구현되었습니다.

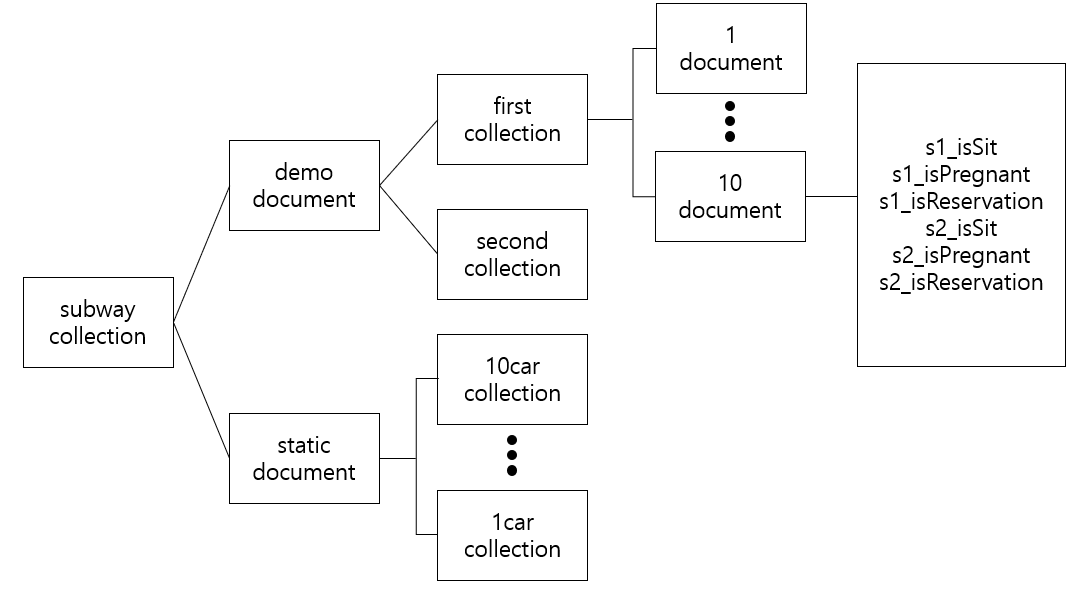
  

* 경로 화면의 아래의 바를 드래그하면 탑승역의 전체 시간표를 볼 수 있을 예정입니다. 원하는 시간을 클릭하면 해당 열차의 좌석현황을 볼 수 있으며 예약이 가능합니다.

현재는 예약과 동시에 이동 중의 레이아웃이 출력되지만 DB와의 연동이 완료되면 예약 후 탑승 버튼을 누르면 해당 레이아웃으로 이동하여 목표 역에 도착하기 까지의 시간을 계산, 도착하게 되면 환승할 시간표를 재선택하는 레이아웃 또는 도착 레이아웃을 출력할 계획입니다.

2. 서버 및 DB

* Firestore에 사용자 정보를 저장할 user collection, 산모 인증을 위해 카드번호와 이름을 저장해둘 pregnant\_init collection, 예약 및 좌석 정보를 저장할 subway collection을 만들었습니다.
* 이 중 subway collection의 경우 당초 계획은 노선별, 시간표별로 열차마다 DB를 모두 만들 계획이었으나 API로부터 해당 열차를 특정해낼 단서를 찾지 못했으며 하드웨어 개수의 한계 때문에 실제 하드웨어와 연동하여 센서값을 저장할 demo document와 이외의 노선의 좌석 정보를 저장할 static document로 분리하여 DB를 설계하였습니다.
* DB 설계는 다음과 같습니다.



최상위 subway collection 아래에 demo용 db인 document와 고정값 db인 static document이 있습니다.

demo document는 직접 하드웨어와 연동하며 특정 노선에 대해서만 이용합니다. 환승하는 경우를 위해 first, second로 나누었으며 해당 컬렉션 안에는 각 칸의 document와 field가 저장됩니다.

isSit은 로드셀 센서의 값을 통해 착석 여부를 판별하는 boolean type field입니다.

isPregnant는 BLE를 통해 저장되는 산모 여부를 판별하는 boolean type field입니다.

isReservation은 어플을 통해 예약 여부를 판별하는 boolean type field입니다.

static document 하위 DB 또한 같습니다.

* 라즈베리파이의 센서값을 MQTT 브로커를 통해 Firestore에 저장하기 위하여 express 서버를 구축하였습니다.

3. 하드웨어

* 당초 계획은 아두이노를 이용하는 것이었으나 안정성 문제로 라즈베리파이로 대체하였습니다.