# ITE2038 Assignment#3

#### Node Application

프로그램은 다음과 같은 개발 스택을 가지고 있습니다.

* Node.js
  + Express
  + sequelize
* MySQL *(JSON Column을 사용하기 위해서 5.8 이상의 버전이 필요합니다.)*

프로그램을 실행하기 위해서는 먼저 의존성 패키지를 설치해야 합니다.

npm install

을 통해 node packages를 설치할 수 있습니다.

의존성이 모두 설치되고 데이터베이스에 접속할 수 있는 상태가 되면 아래의 명령을 통해 웹 어플리케이션을 실행할 수 있습니다. 실행된 웹 어플리케이션은 브라우저에서 localhost:80 으로 접속할 수 있습니다.

npm start

#### Database Initialize

MySQL DB를 사용합니다. MYSQL을 설치하고 접속 정보를 `config.json`에 작성해야 합니다. `config.example.json`을 복사해서 DB에 접속할 수 있도록 작성하면 됩니다. Database 이름은 \*\*ITE2038\*\*을 기본으로 사용합니다.

\*프로그램이 MySQL DB에 접속할 때 Permission Denied문제가 발생할 수 있습니다. 그런 경우에는 다음 명령을 MySQL에서 실행하세요.\*

ALTER USER 'user'@'localhost' IDENTIFIED WITH mysql\_native\_password BY 'password'

`init.sql`에는 Application에 필요한 Database를 초기화 하는 Table과 Relation에 대한 정보가 담겨있습니다.

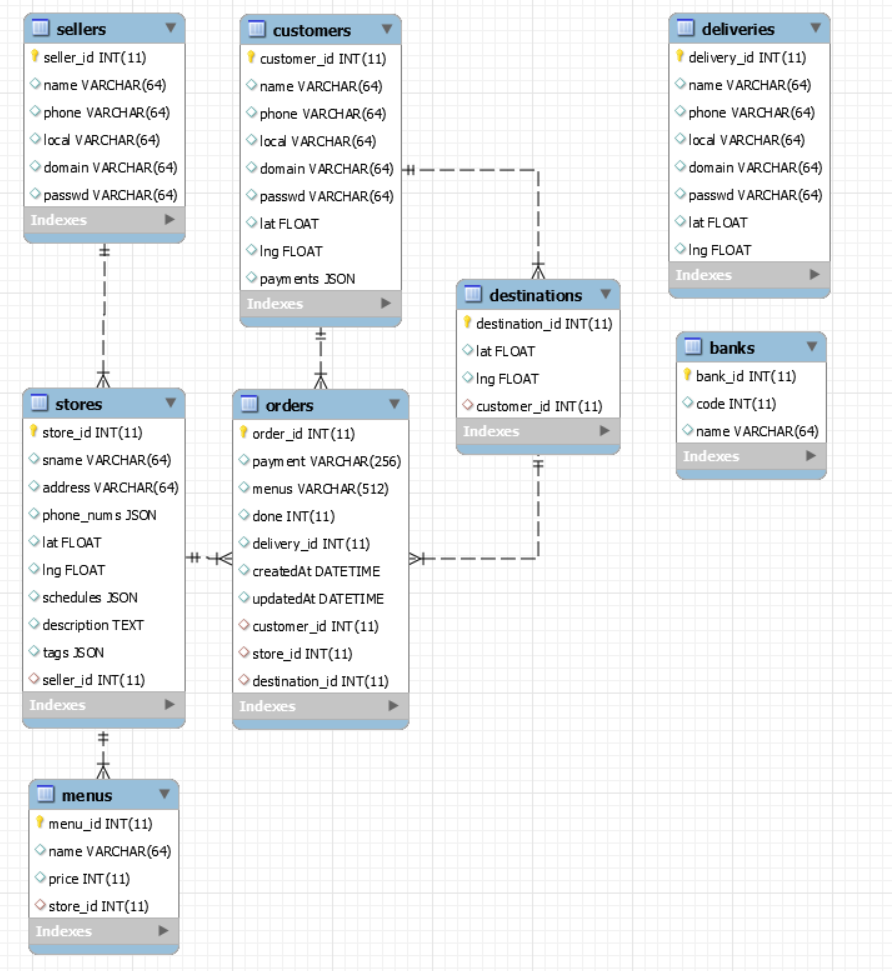
`example.sql`에는 테스트에 필요한 예시 정보가 담겨있습니다. 아래와 같은 명령을 통해 database에 삽입할 수 있습니다. Example.sql은 예제로 주어진 DB자료를 기반으로 조금씩 가져와서 만든 자료입니다. Bank, Customer, Deliver, Seller, Store, Menu들을 집어넣습니다. 테스트용 계정으로 각각 customer, seller, delivery @ hanyang.ac.kr 계정으로 비밀번호 123456으로 되어있습니다.

mysql -u root -p < init.sql

mysql -u root -p < example.sql

만약 example.sql을 mysql에 삽입하는데 에러가 걸린다면 코드 페이지 문제일 수 있습니다. 윈도우 사용중이라면 chcp65001 입력하여 코드페이지를 바꿔주세요.

## 구현



Database는 다음과 같이 구성되어 있습니다.

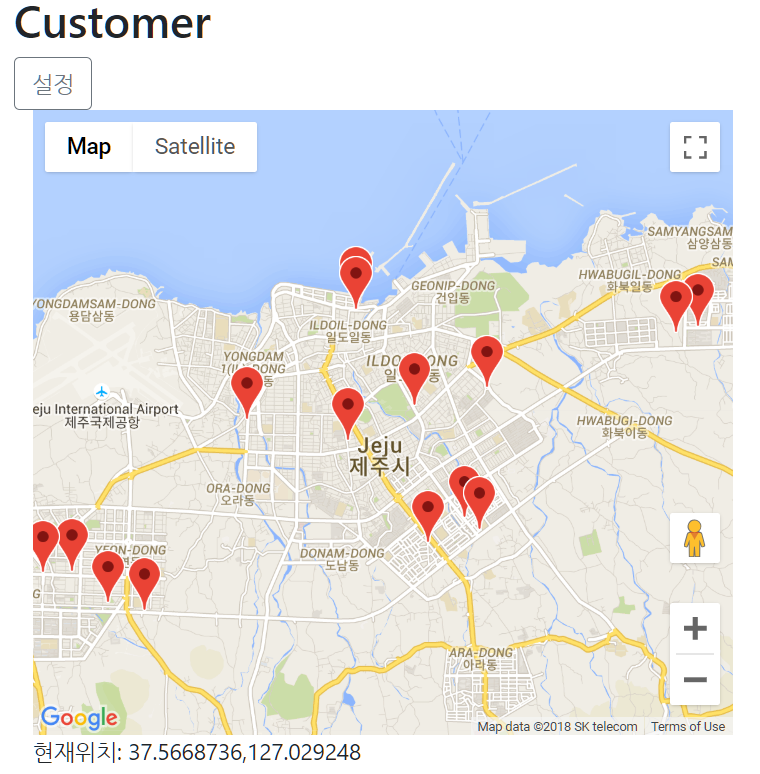
구현은 Node.js의 Sequelize(ORM)을 사용하여 구현하였습니다. 이는 app/models 모듈에서 확인할 수 있으며 각각의 모델은 DB에서 테이블을 가지게 됩니다. 각 모델에는 associate를 통해 모델관의 연관을 정의하는데 다음과 같습니다.

* Seller has many Store.
* Customer has many Order, Destination
* Store belongs to Seller
* Store has many Order, Menu
* Menu belongs to Store
* Order belongs to Customer, Store, Destination
* Destination belongs to Customer

이를 controller에서 관리하는데 기본적인 index(show all instances), show(show specific instance by id), destroy(delete instance), create, update 등은 서로 같습니다.

Routes에 정의된 API들을 통해서 모델을 다룰 수 있습니다. 예를 들어 GET:/seller/3 은 seller\_id가 3인 Seller를 선택해 보여줍니다. POST:/order는 새 Order를 parameter의 값으로 만들게 됩니다. 기본적으로 bank, customer, delivery, destination, menu, order, seller, store가 있고 paramter를 통해 where 조건을 붙여줄 수 있습니다. (예를들어 /order?store\_id=3&customer\_id=1을 하면 customer의 id가 1이고, store\_id가 3인 order들만 가져옵니다.)

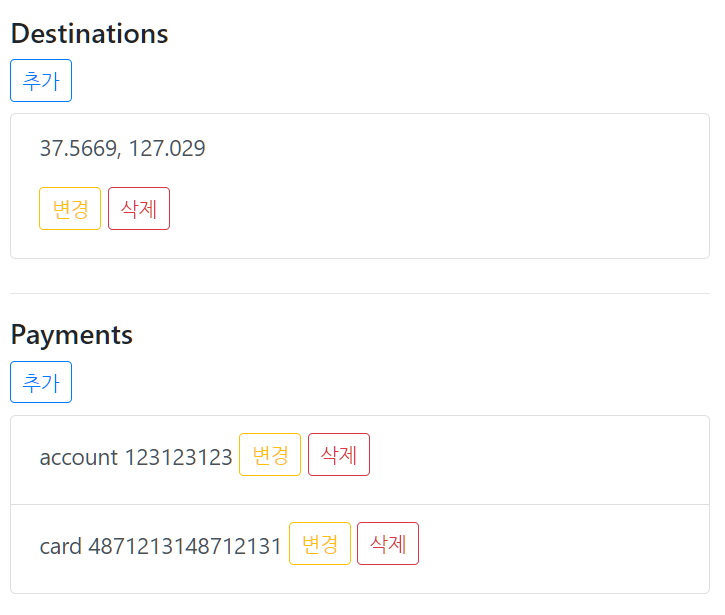
## 실행화면



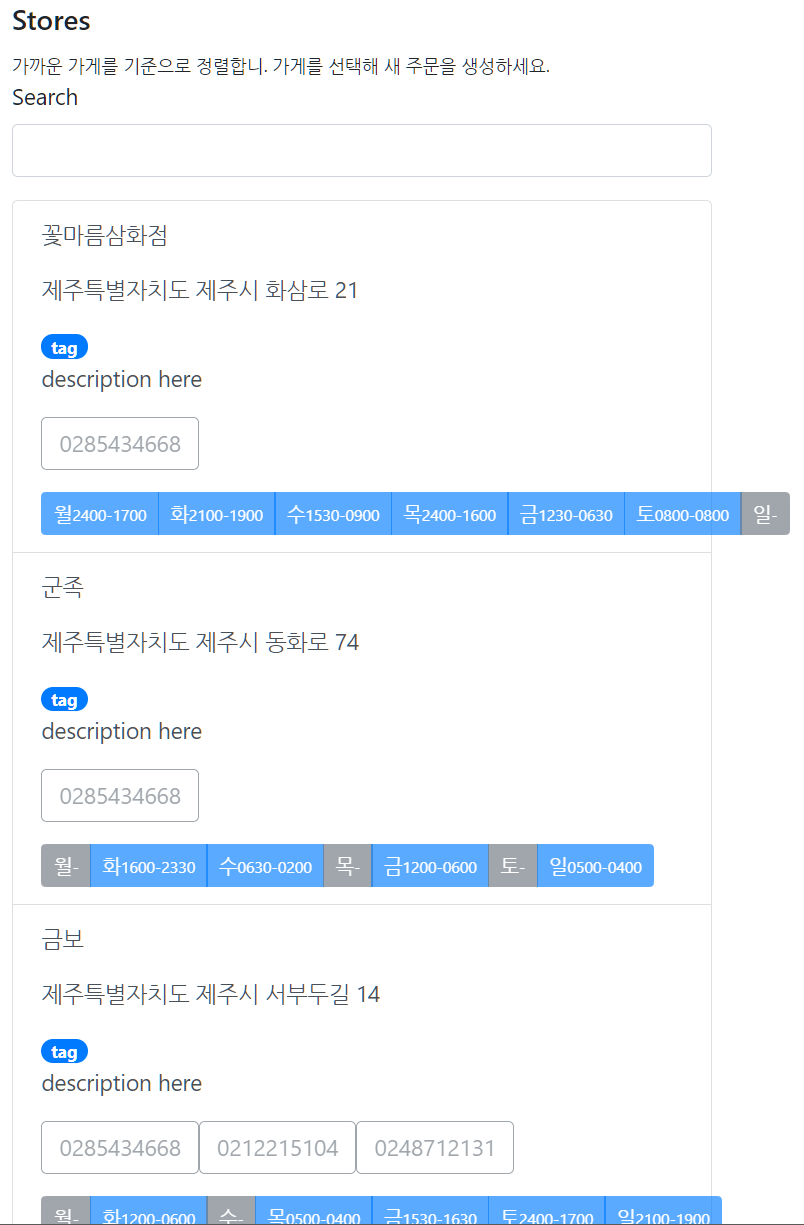
Customer화면에서는 인근 Store가 마커로 지도에 표시됩니다. 지도는 처음 시작될 때 자신의 위치를 기준으로 생성됩니다. 설정 버튼을 눌러 자신의 정보를 편집할 수 있습니다. (샘플 가게 위치가 제주도에 있어 지도를 축소해야 볼 수 있습니다.)



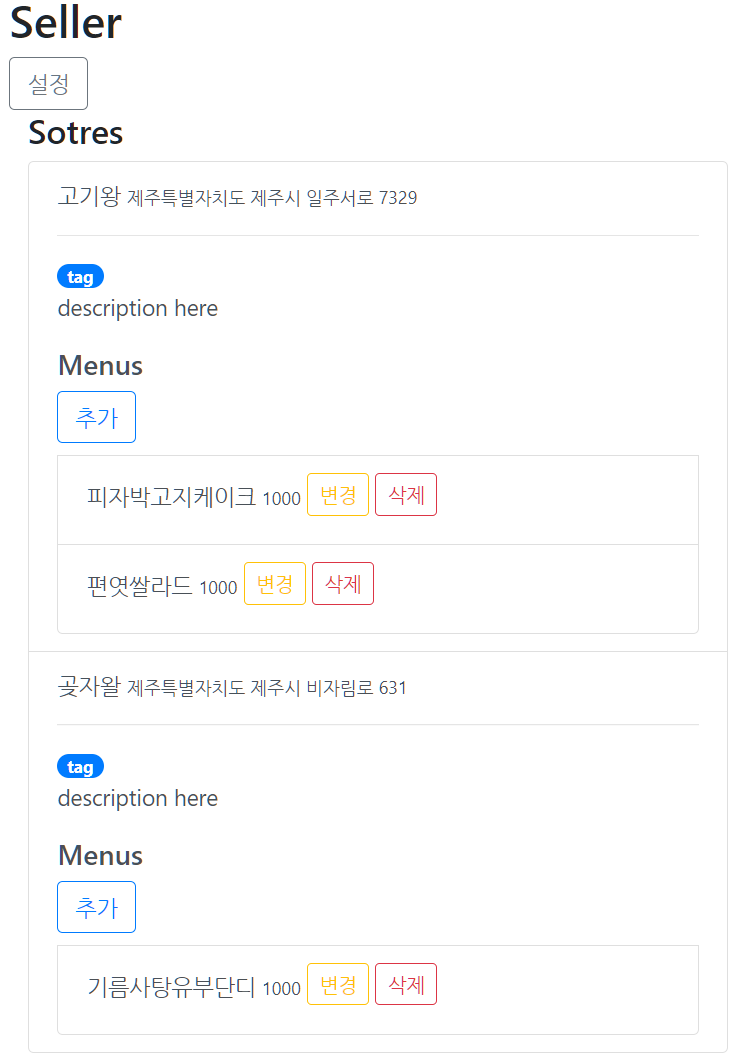
주문 목록에서는 준비중, 배달중, 취소됨, 완료됨이 표시됩니다.



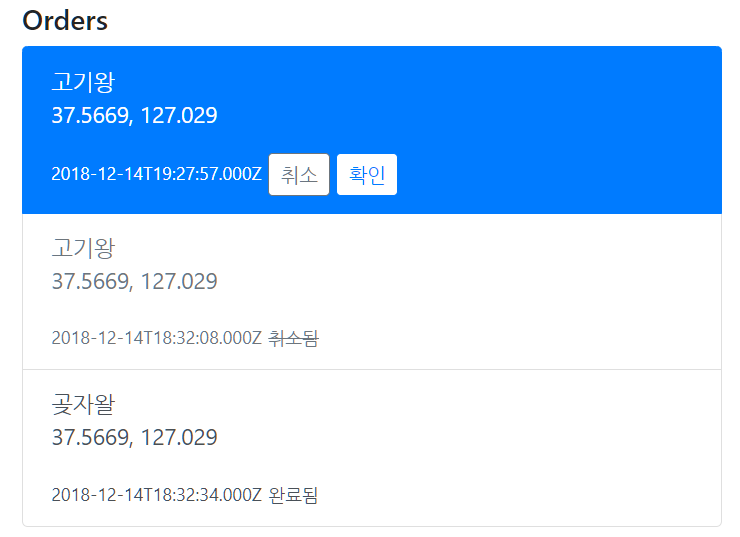
주소록과 결제수단을 추가/변경/삭제 할 수 있습니다.



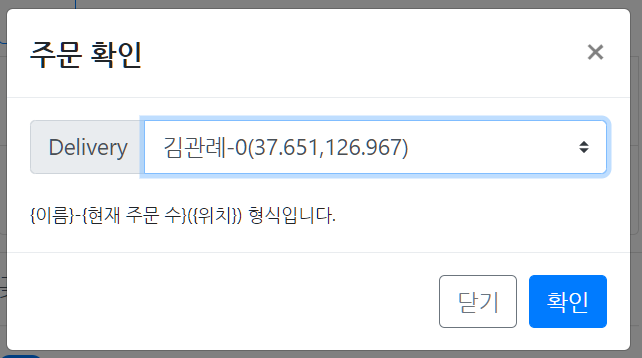
가게 목록에서는 운영하는 요일을 확인할 수 있고 전화번호와 태그, 설명 그리고 주소를 볼 수 있습니다.



Seller는 Store를 관리할 수 있습니다. 각 Store의 Menu를 추가, 변경, 삭제 할 수 있으며 태그와 설명, 주소를 확인할 수 있습니다.



주문을 취소하거나 확인할 수 있고,



주문을 확인하면서 어떤 Delivery를 선택할 지 정할 수 있습니다. 가까운 순서대로 정렬하여 Stock이 4개 이하인 배달부만 표시됩니다.



Delivery는 현재 자신에게 할당된 배달을 확인할 수 있습니다.