



# BÁO CÁO ĐỒ ÁN CUỐI KÌ

## LẬP TRÌNH WEB 2

Giáo viên: Đỗ Nguyên Kha

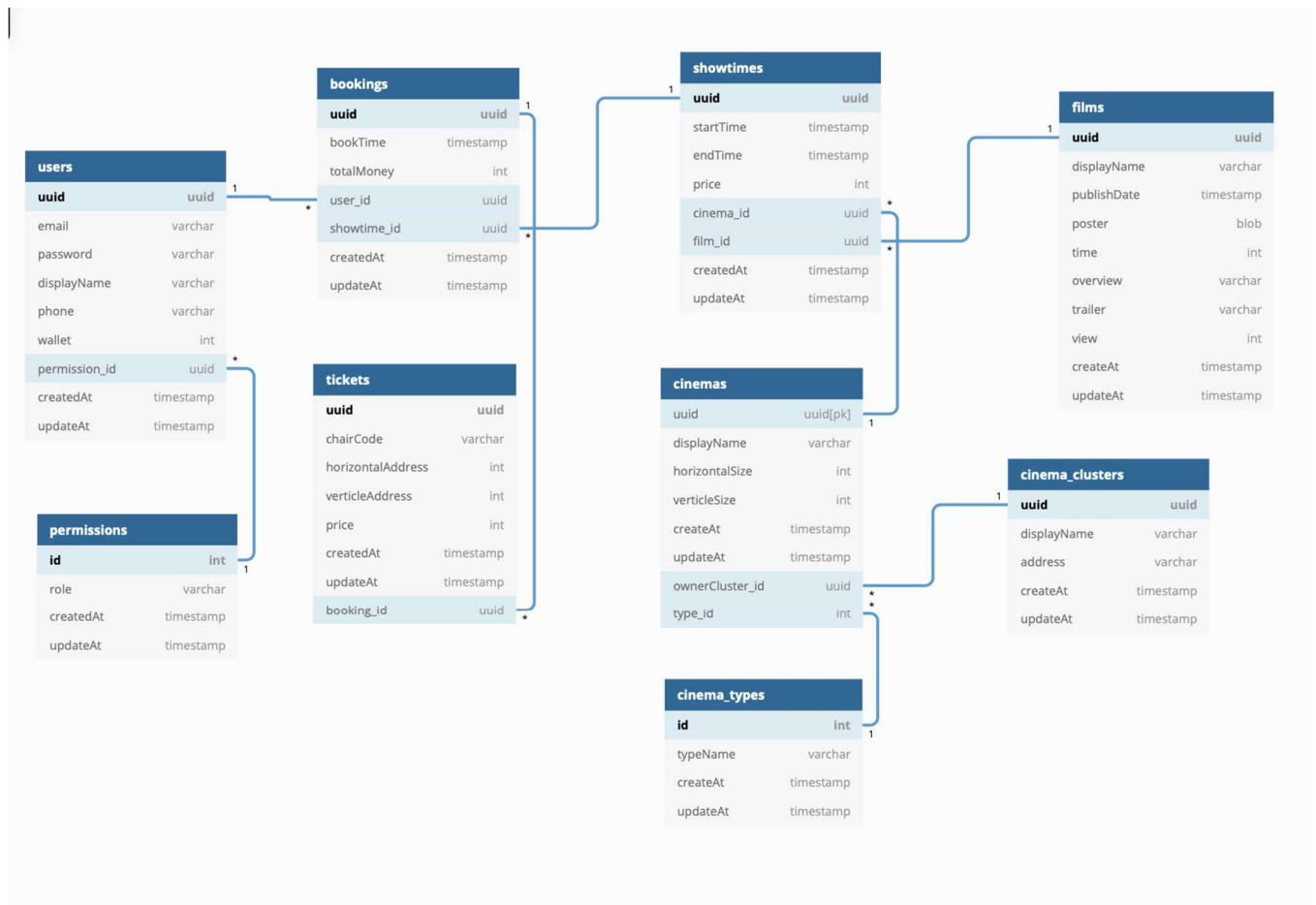


## I. Thành viên

STT	Mã số sinh viên	Họ và tên
1	18600280	Lưu Hữu Tình
2	18600287	Nguyễn Minh Trọng Trí
3	18600351	Nguyễn Đăng Khoa
4	18600266	Võ Văn Minh Thông

## II. Thiết kế

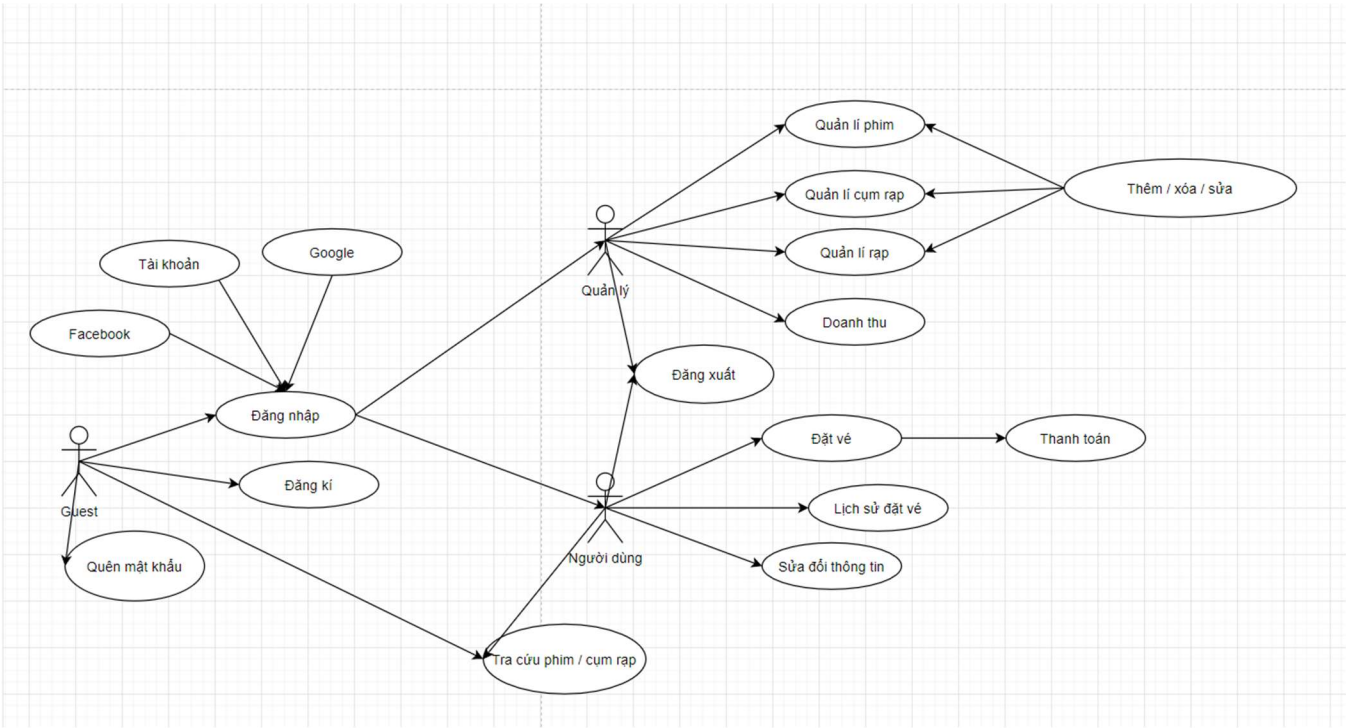
### a. Database



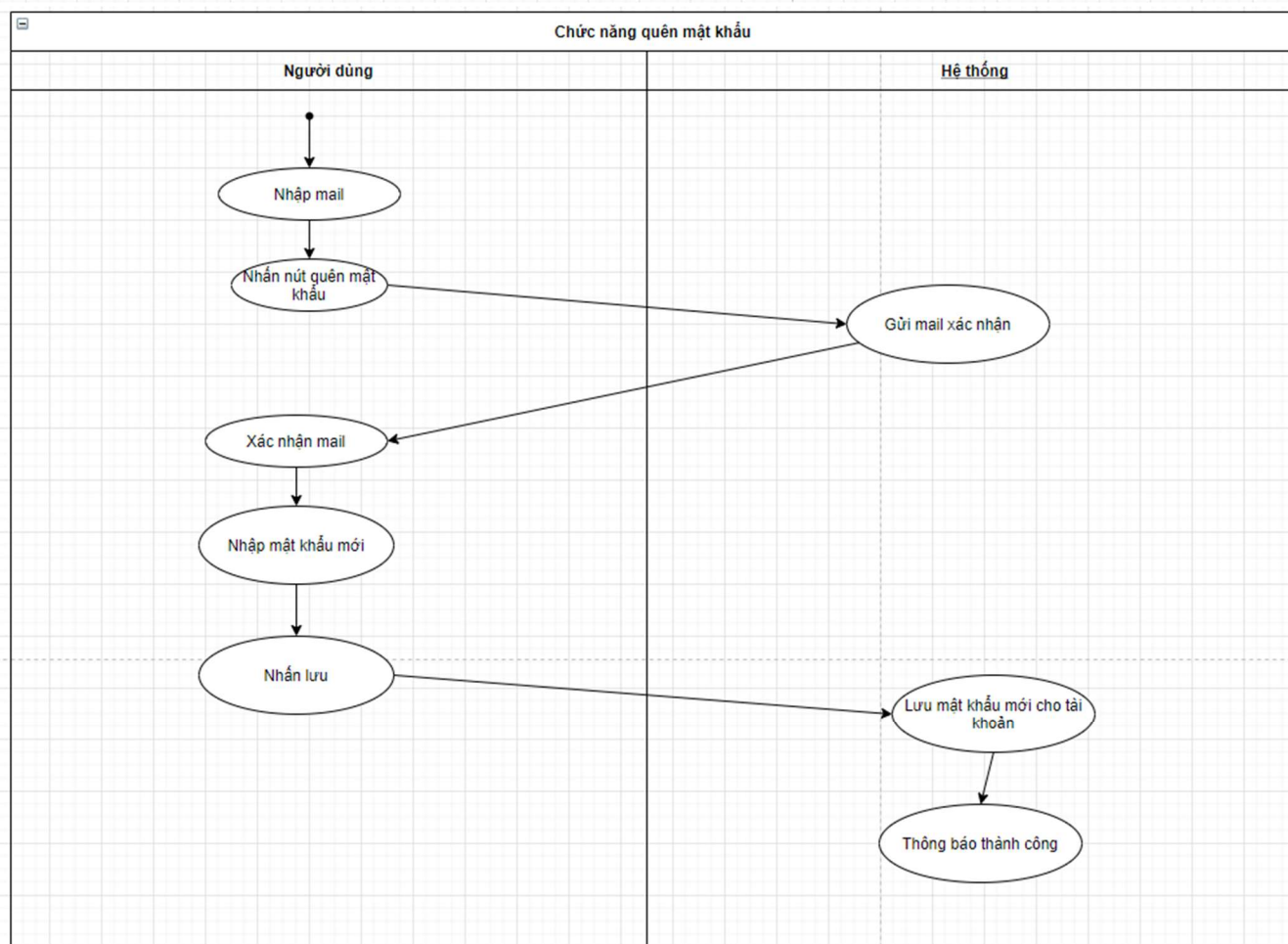
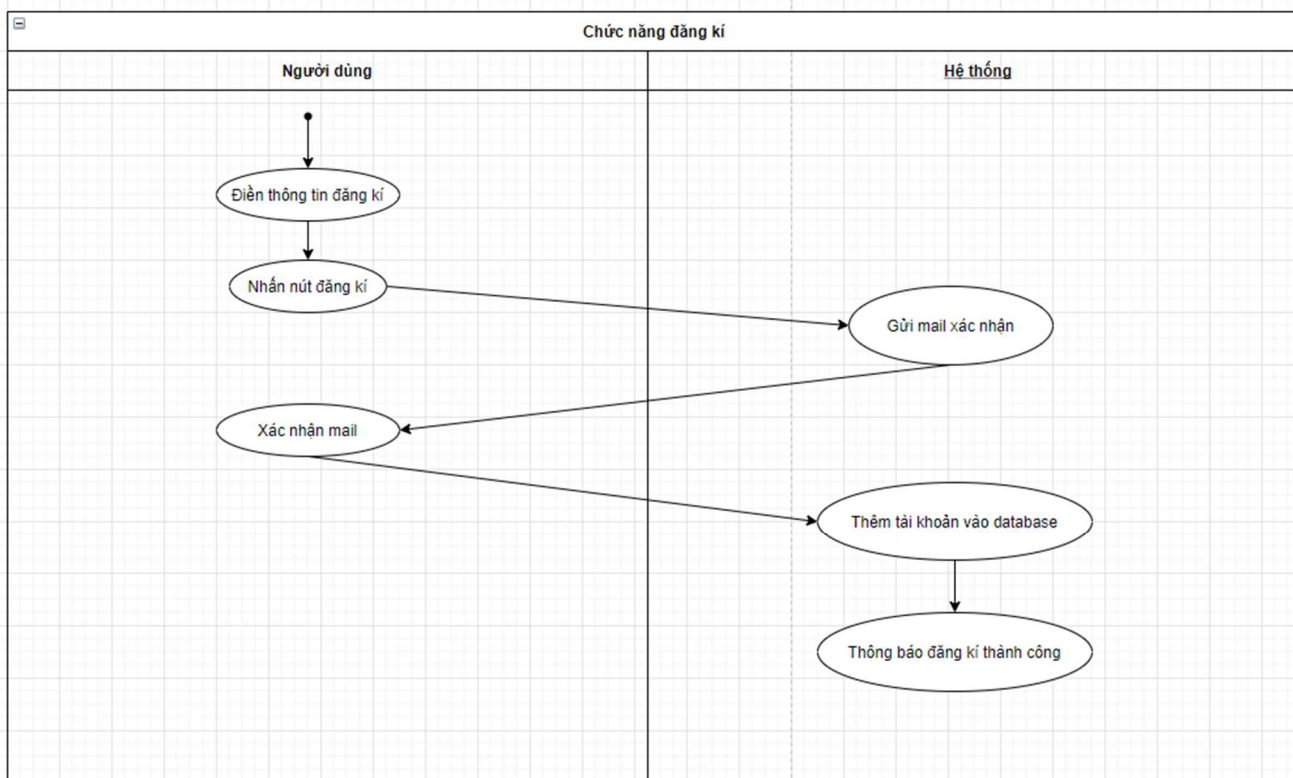
- + Một phim ( film ) bao gồm các thông tin cơ bản: Tên, trailer, poster,...
- + Người dùng ( user ) sẽ được phân quyền ( permission ) → khách / người quản lí
- + Mỗi cụm rạp ( cinema\_cluster ) bao gồm nhiều rạp ( cinema ), các rạp được phân loại ( cinema\_type )
- + Một suất chiếu ( show ) sẽ có giá vé, chiếu 1 bộ phim trong khoảng thời gian nhất định. Mỗi suất chiếu thuộc về 1 rạp.
- + Cho phép người dùng đặt vé ( booking ), mỗi lần đặt có thể bao gồm nhiều vé ( ticket )
- + Tài khoản admin: **admin@gmail.com**, password: **admin**

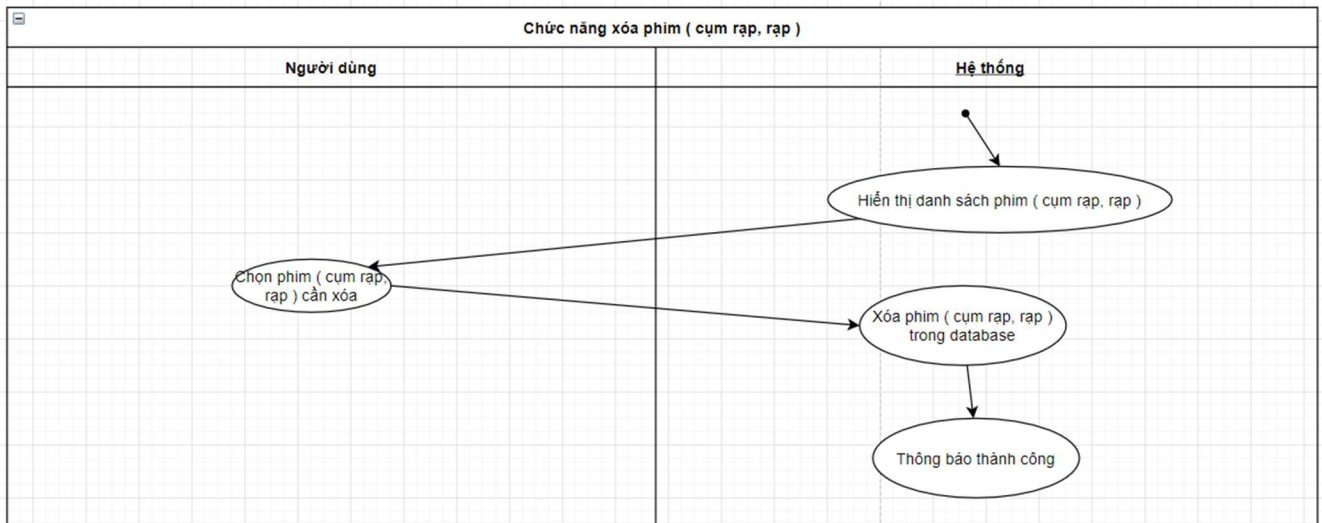
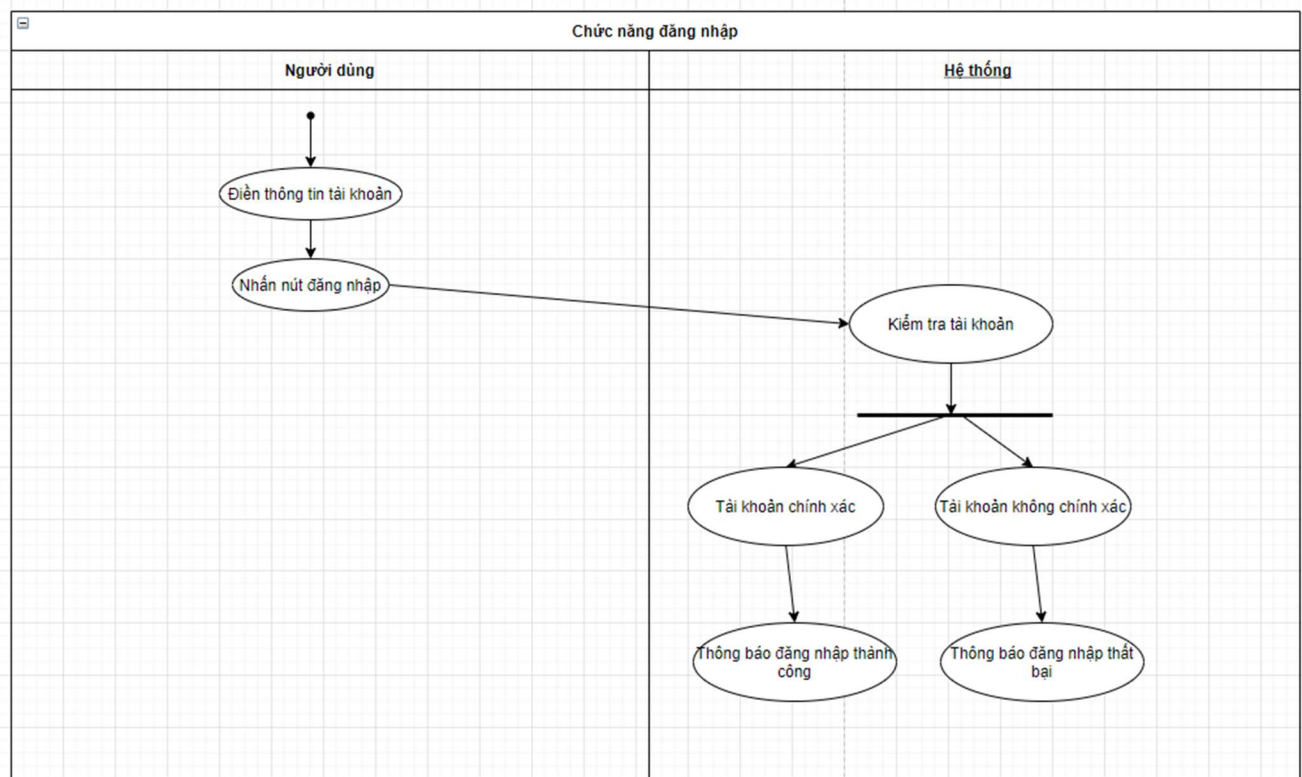


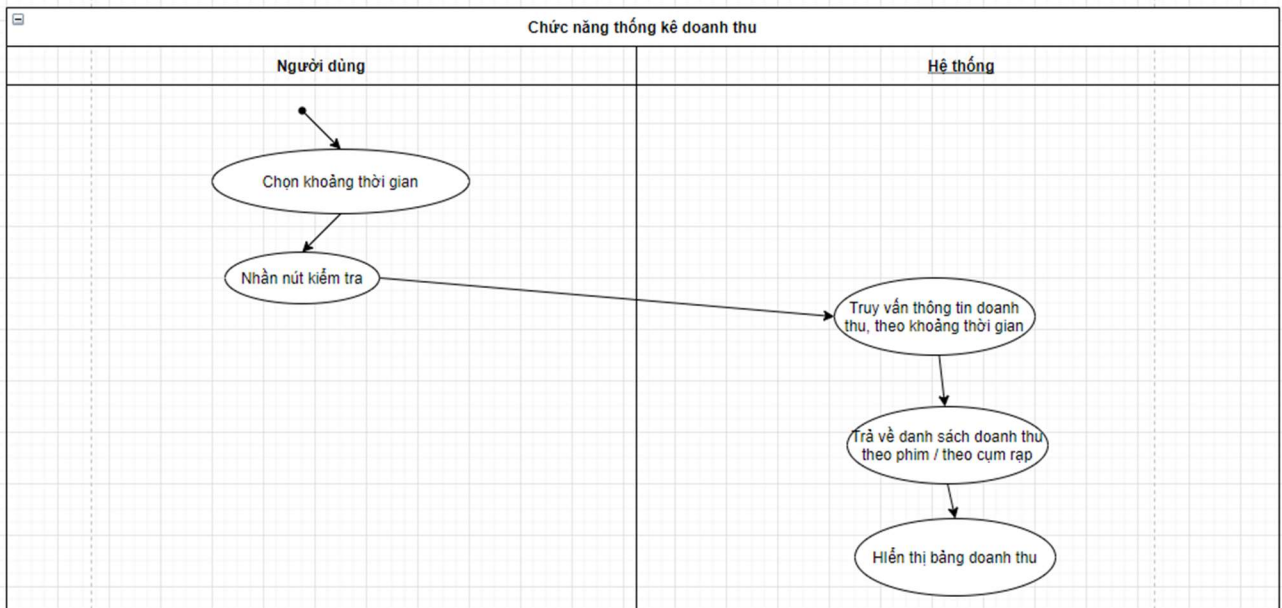
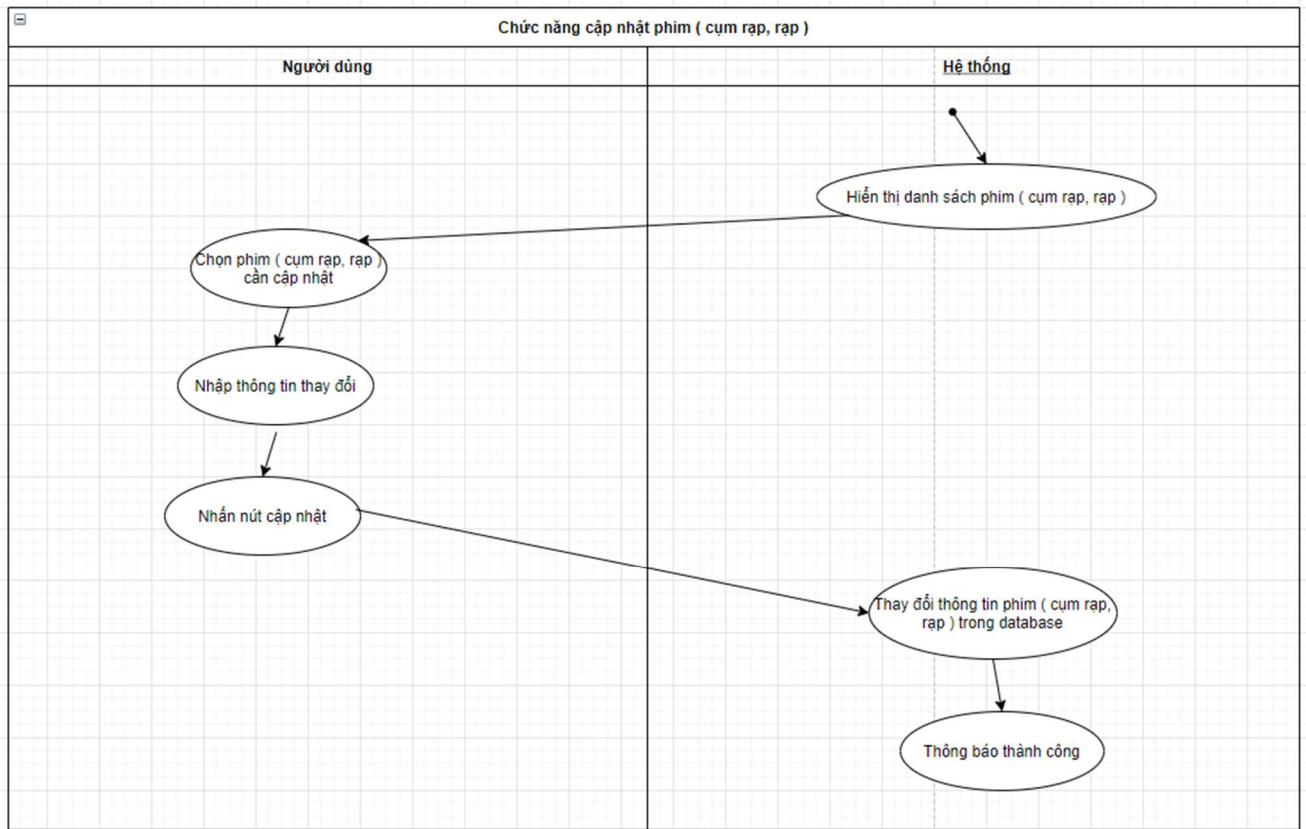
b. Use Case



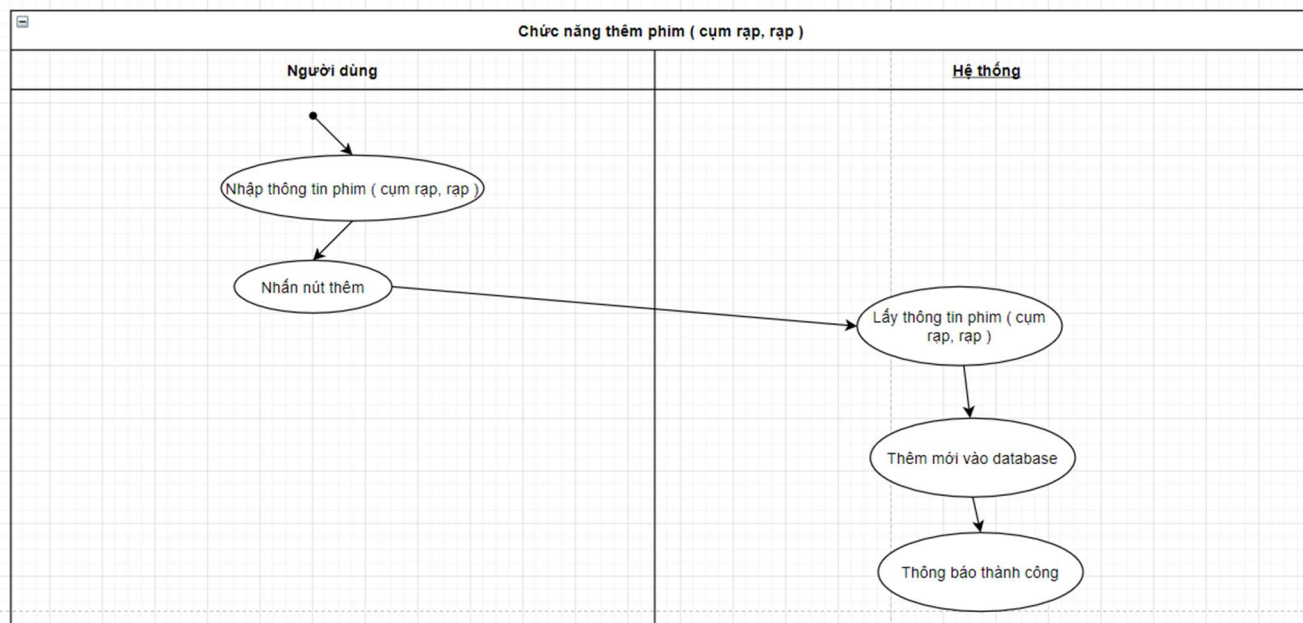
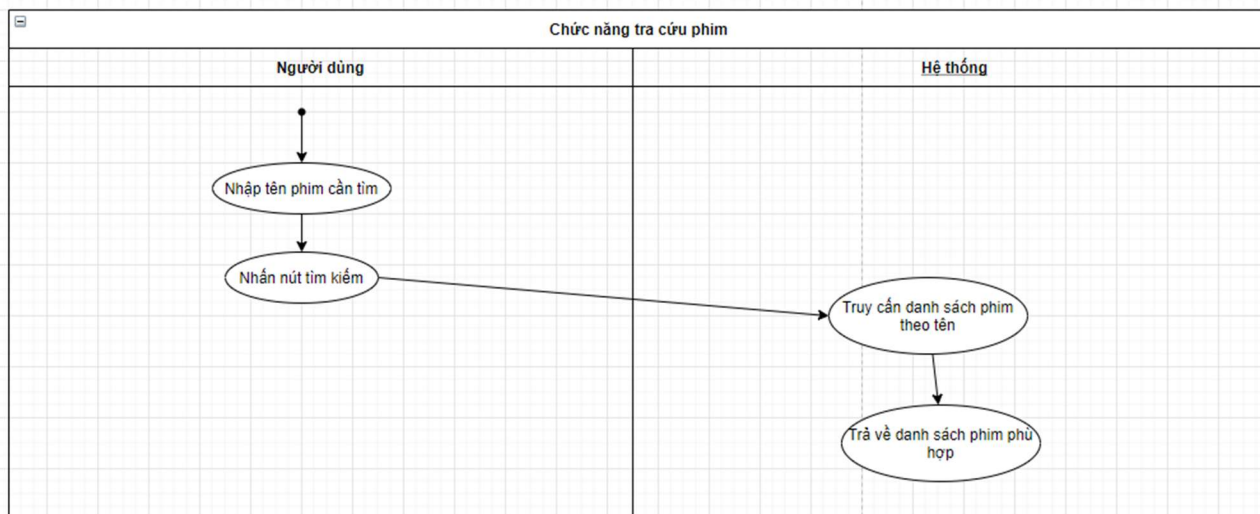
### c. Activity diagram



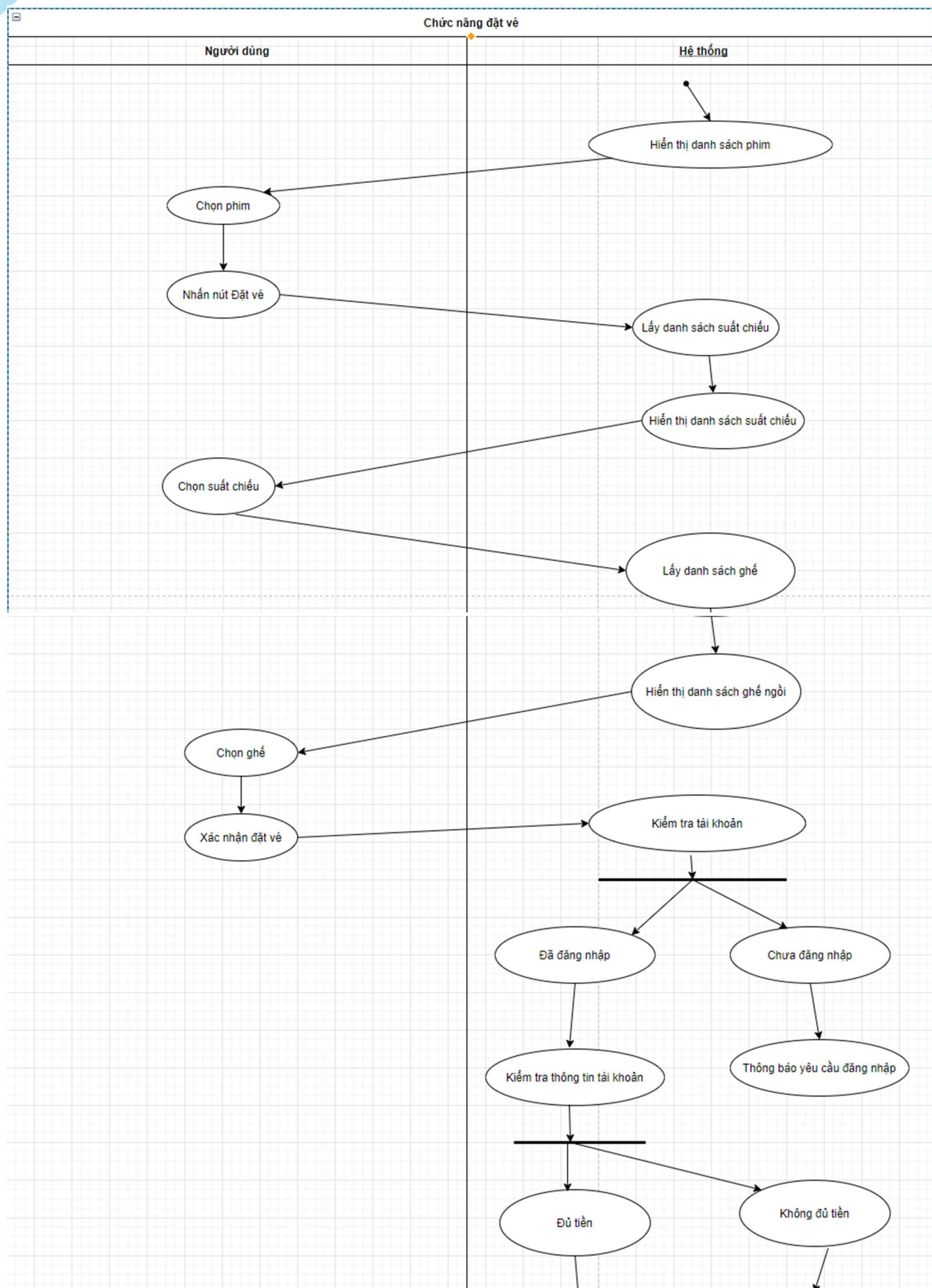


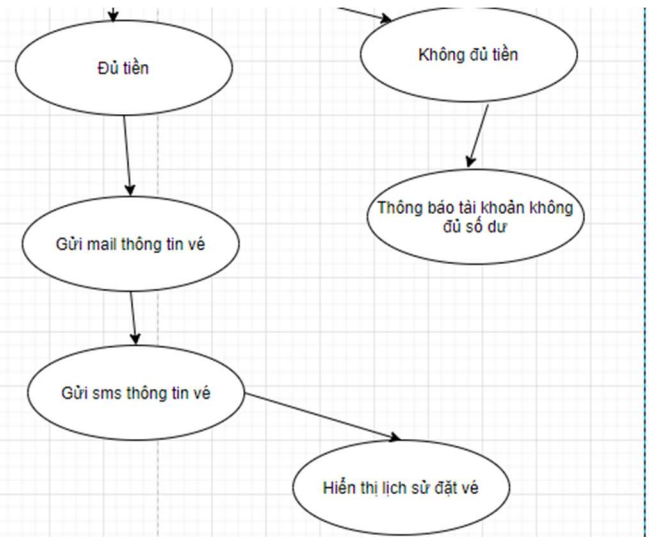












### III. Cài đặt ( Hướng dẫn chạy web )

#### a. Cài đặt các dependency trong file package.json

```
{ } package.json 1 X
Lập trình Web 2 > DoAn > git new > booking-cinema-tickets > v3
12   license: "ISC",
13   "dependencies": {
14     "bcrypt": "^5.0.1",
15     "body-parser": "^1.19.0",
16     "cookie-parser": "^1.4.5",
17     "cookie-session": "^1.4.0",
18     "cors": "^2.8.5",
19     "dotenv": "^10.0.0",
20     "ejs": "^3.1.6",
21     "express": "^4.17.1",
22     "express-ejs-layouts": "^2.5.0",
23     "express-session": "^1.17.2",
24     "fs": "0.0.1-security",
25     "http": "0.0.1-security",
26     "jquery": "^3.6.0",
27     "multer": "^1.4.2",
28     "nodemailer": "^6.6.2",
29     "nodemon": "^2.0.7",
30     "passport": "^0.4.1",
31     "passport-facebook": "^3.0.0",
32     "passport-google-oauth20": "^2.0.0",
33     "pg": "^8.6.0",
34     "pg-hstore": "^2.3.3",
35     "prettier": "^2.3.0",
36     "request": "^2.88.2",
37     "sequelize": "^6.6.2",
38     "twilio": "^3.65.0"
39   }
40 }
```

#### b. Cấu hình kết nối database

```
JS database.js M X
Lập trình Web 2 > DoAn > git new > booking-cinema-tickets > v3 > WebPrograming2_Final-Project > src > config > JS database.js > ...
3 | const sequelize = new Sequelize( process.env.DATABASE_URL || "postgres://postgres:postgres@localhost:5432/final_project",
4 |   {
5 |     dialect: 'postgres',
6 |     dialectOptions: {
7 |       /*ssl: {
8 |         rejectUnauthorized: false,
9 |       }*/
10 |    },
11 |   }
12 | );
13
14 module.exports = sequelize;
15
```

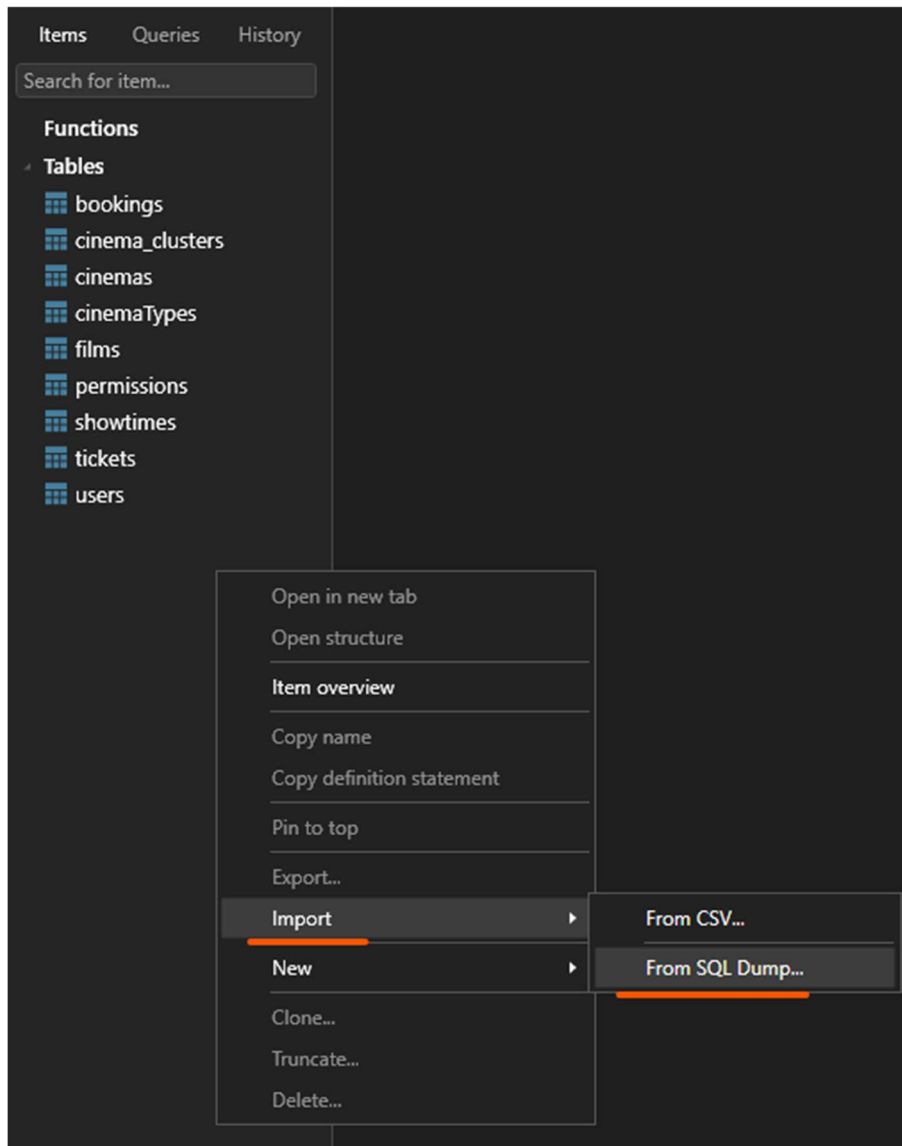
### c. Chạy npm start, lấy cấu trúc bảng

```
PS D:\Vhọc kì II, 2020-2021\Lập trình Web 2\DoAn\git new\booking-cinema-tickets\v3\WebPrograming2_Final-Project\src> npm start

> Final_Project@1.0.0 start D:\Vhọc kì II, 2020-2021\Lập trình Web 2\DoAn\git new\booking-cinema-tickets\v3\WebPrograming2_Final-Project
> nodemon src/app.js

[nodemon] 2.0.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node src/app.js`
body-parser deprecated undefined extended: provide extended option src\routers\User\auth\loginWithFacebook.js:47:23
express-session deprecated undefined resave option; provide resave option src\routers\User\auth\loginWithFacebook.js:48:12
express-session deprecated undefined saveUninitialized option; provide saveUninitialized option src\routers\User\auth\loginWithFacebook.js:48:12
body-parser deprecated undefined extended: provide extended option src\routers\User\auth\loginWithGg\auth.js:19:23
express-session deprecated undefined resave option; provide resave option src\routers\User\auth\loginWithGg\auth.js:20:12
express-session deprecated undefined saveUninitialized option; provide saveUninitialized option src\routers\User\auth\loginWithGg\auth.js:20:12
Executing (default): SELECT 1+1 AS result
Executing (default): SELECT 1+1 AS result
Executing (default): SELECT 1+1 AS result
Executing (default): SELECT 1+1 AS result
Executing (default): SELECT 1+1 AS result
Executing (default): SELECT 1+1 AS result
Executing (default): SELECT 1+1 AS result
Executing (default): SELECT 1+1 AS result
Executing (default): CREATE TABLE IF NOT EXISTS "permissions" ("id" SERIAL , "role" VARCHAR(255) NOT NULL, "createdAt" TIMESTAMP WITH TIME ZONE NOT NULL, "updatedAt" TIMESTAMP WITH TIME ZONE NOT NULL, PRIMARY KEY ("id"));
Executing (default): SELECT i.relname AS name, ix.indisprimary AS primary, ix.indisunique AS unique, ix.indkey AS indkey, array_agg(a.attname) AS column_indexes, array_agg(a.attname) AS column_names, pg_get_indexdef(ix.indexrelid) AS definition FROM pg_class t, pg_class i, pg_index ix, pg_attribute a WHERE t.oid = ix.indrelid AND i.oid = ix.indexrelid AND a.attrelid = t.oid AND t.relkind = 'r' and t.relname = 'permissions' GROUP BY i.relname, ix.indexrelid, ix.indisprimary, ix.indisunique, ix.indkey ORDER BY i.relname;
Executing (default): CREATE TABLE IF NOT EXISTS "users" ("uid" UUID NOT NULL , "email" VARCHAR(255) NOT NULL, "password" VARCHAR(255) NOT NULL, "displayName" VARCHAR(255) NOT NULL DEFAULT 2, "phone" VARCHAR(255) NOT NULL, "wallet" INTEGER NOT NULL DEFAULT 0, "createdAt" TIMESTAMP WITH TIME ZONE NOT NULL, "updatedAt" TIMESTAMP WITH TIME ZONE NOT NULL, "permission_id" INTEGER REFERENCES "permissions" ("id") ON DELETE CASCADE ON UPDATE CASCADE, PRIMARY KEY ("uid"));
Executing (default): SELECT i.relname AS name, ix.indisprimary AS primary, ix.indisunique AS unique, ix.indkey AS indkey, array_agg(a.attname) AS column_indexes, array_agg(a.attname) AS column_names, pg_get_indexdef(ix.indexrelid) AS definition FROM pg_class t, pg_class i, pg_index ix, pg_attribute a WHERE t.oid = ix.indrelid AND i.oid = ix.indexrelid AND a.attrelid = t.oid AND t.relkind = 'r' and t.relname = 'users' GROUP BY i.relname, ix.indexrelid, ix.indisprimary, ix.indisunique, ix.indkey ORDER BY i.relname;
```

### d. Trong table plus, import dữ liệu từ file: \database\insert\_Data.sql



e. Chạy web: localhost:3000/

