ĐẠI HỌC BÁCH KHOA HÀ NỘI

Trường Công nghệ thông tin và Truyền thông

A red and yellow logo

AI-generated content may be incorrect.

Tài liệu thiết kế phần mềm

(Software Design Document – SDD)

Phiên bản 1

Kĩ thuật phần mềm

Đề tài: Quản lý chung cư BlueMoon

Nhóm 10

Nguyễn Văn Thành Đạt - 20225606

Nguyễn Mạnh Thái Hà - 20225621

Nguyễn Thanh Tân - 20225923

Nguyễn Minh Quân – 20235814

Nguyễn Trường Sơn - 20230097

*Hà Nội, ngày 10 tháng 05 năm 2025*

**Mục lục**

Mục lục 1

1 Giới thiệu 2

1.1 Mục đích 2

1.2 Phạm vi 2

1.3 Từ điển thuật ngữ 2

1.4 Tài liệu tham khảo 2

2 Thiết kế kiến trúc 3

2.1.1 Lựa chọn kiến trúc phần mềm 3

2.1.2 Thiết kế tổng quan 3

2.1.3 Thiết kế chi tiết gói 4

2.2 Thiết kế chi tiết 5

2.2.1 Thiết kế cơ sở dữ liệu 5

2.2.2 Thiết kế lớp 5

2.2.3 Thiết kế giao diện 6

# Giới thiệu

## Mục đích

Tài liệu này mô tả chi tiết thiết kế hệ thống quản lý và thu phí chung cư BlueMoon được phát triển bằng công nghệ web với Node.js và PostgreSQL. Tài liệu được tạo ra để cung cấp hướng dẫn chi tiết cho đội phát triển trong quá trình cài đặt phần mềm, làm cơ sở cho việc kiểm thử và bảo trì hệ thống, đảm bảo tính nhất quán trong thiết kế và phát triển.

Đối tượng đọc tài liệu:

* Đội phát triển phần mềm
* Kiến trúc sư phần mềm
* Tester và đội QA
* Người quản lý dự án

## Phạm vi

Tài liệu này mô tả thiết kế cho hệ thống quản lý chung cư BlueMoon phiên bản 1.0, bao gồm các module chính:

* Module xác thực và phân quyền: Đăng nhập, quản lý session, phân quyền theo vai trò
* Module Admin (Kế toán): Quản lý khoản thu, thu phí, thống kê tài chính
* Module Tổ trưởng/ Tổ phó: Quản lý hộ khẩu, nhân hẩu, tạm trú tạm vắng, báo cáo
* Module Cư dân: Thông tin cá nhân, thanh toán trực tuyến, phản hồi

Hệ thống được thiết kế như một ứng dụng web sử dụng:

Backend: Node.js với Express.js framework

Frontend: Vite, React

Database: PostgreSQL với Sequelize ORM

Session Management: Express-session với MongoDBstore

## Từ điển thuật ngữ

|  |  |
| --- | --- |
| Thuật ngữ | Định nghĩa |
| Route Handler | Hàm xử lý HTTP request trong Express.js |
| Middleware | Hàm trung gian xử lý request trước khi đến route handler |
| Schema | Cấu trúc định nghĩa model trong PostgreSQL |
| Collection | Bảng dữ liệu trong PostgreSQL |
| Session | Phiên làm việc của người dung được lưu trên server |
| EJS Template | Template engine để render HTML động |
| Populate | Kĩ thuật join dữ liệu trong PostgreSQL |
| Hộ khẩu | Đơn vị quản lý gồm một hoặc nhiều nhân khẩu cùng sinh sống |
| Nhân khẩu | Cá nhân cư trú trong chung cư BlueMoon |
| Chủ hộ | Người đại diện hộ khẩu |
| Khoản thu | Các loại phí mà hộ cần nộp |
| Tạm trú/ Tạm vắng |  |

## Tài liệu tham khảo

# Thiết kế kiến trúc

### Lựa chọn kiến trúc phần mềm

Hệ thống được thiết kế theo kiến trúc monolithic web application sử dụng MVC pattern với các đặc điểm sau:

**Node.js + Express.js Framework:**

* **Routes**: Định nghĩa các endpoint API và web routes
* **Controllers**: Logic xử lý nghiệp vụ được tích hợp trong route handlers
* **Models**: PostgreSQL schema định nghĩa cấu trúc dữ liệu
* **Views**: EJS templates để render giao diện người dùng

**Session-based Authentication:**

* Sử dụng express-session để quản lý phiên đăng nhập
* Session data được lưu trữ server-side
* Role-based access control với middleware functions

**PostgreSQL Database:**

* NoSQL document database với flexibility cao
* Sequelize ORM để tương tác với database
* Schema validation và data relationships

**Lý do lựa chọn:**

1. **Rapid Development**: Express.js cung cấp framework minimalist dễ phát triển
2. **Scalability**: PostgreSQL hỗ trợ horizontal scaling tốt
3. **Flexibility**: NoSQL schema cho phép thay đổi cấu trúc dữ liệu dễ dàng
4. **JavaScript Ecosystem**: Sử dụng một ngôn ngữ cho cả frontend và backend
5. **Real-time Capabilities**: Node.js event-driven architecture phù hợp cho real-time features

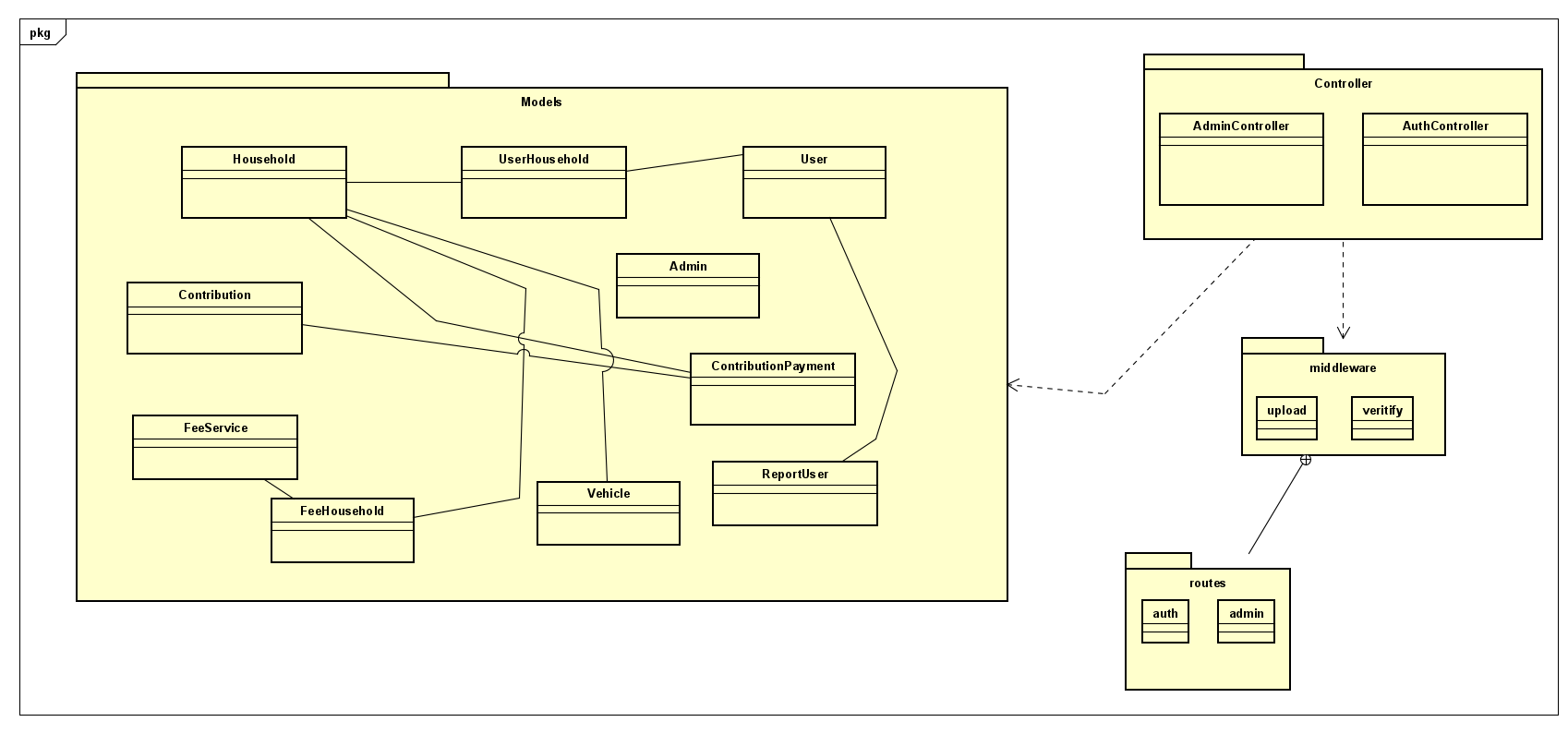
### Thiết kế tổng quan

Sơ đồ kiến trúc phân cấp các chức năng và mô tả chi tiết cho các chức năng, quan hệ giữa các thành phần:

A diagram of a company

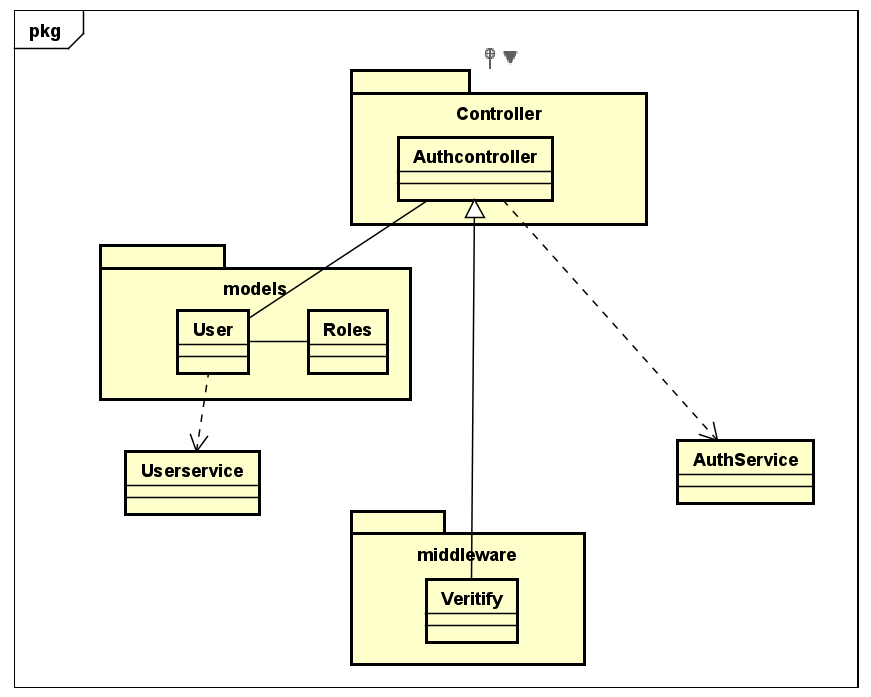
AI-generated content may be incorrect.

Package diagram:



### Thiết kế chi tiết gói

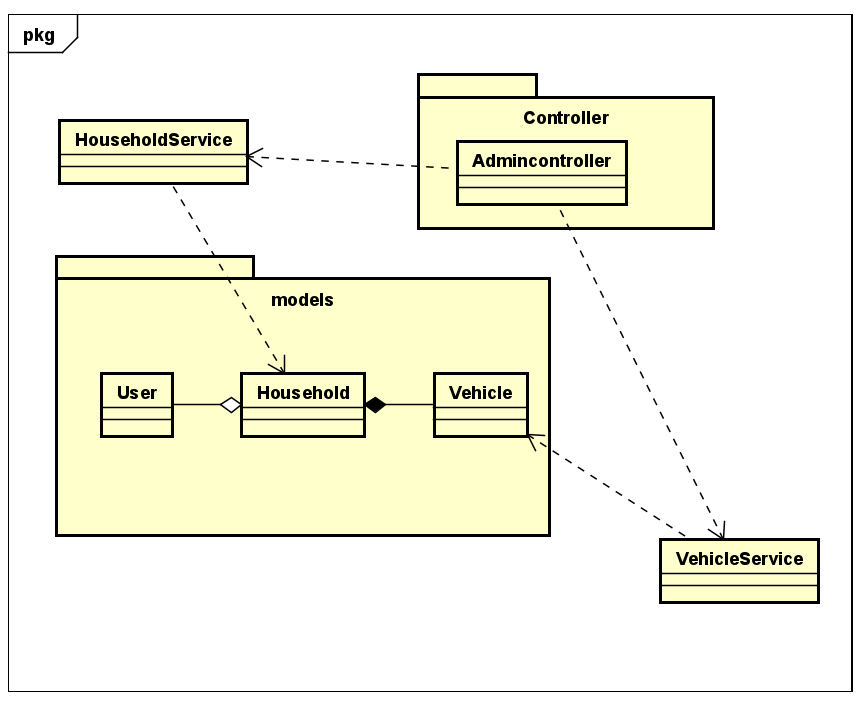
Giải thích ngắn gọn về thiết kế của mình :

**User Management & Authentication:  
  
**

**Mục đích:** Quản lý người dùng, xác thực và phân quyền trong hệ thống.

**Quan hệ chính:**

* **Dependency:** UserService phụ thuộc vào User entity để thao tác dữ liệu
* **Association:** AuthController liên kết với User để xử lý đăng nhập
* **Composition:** AuthController sở hữu và quản lý Middleware

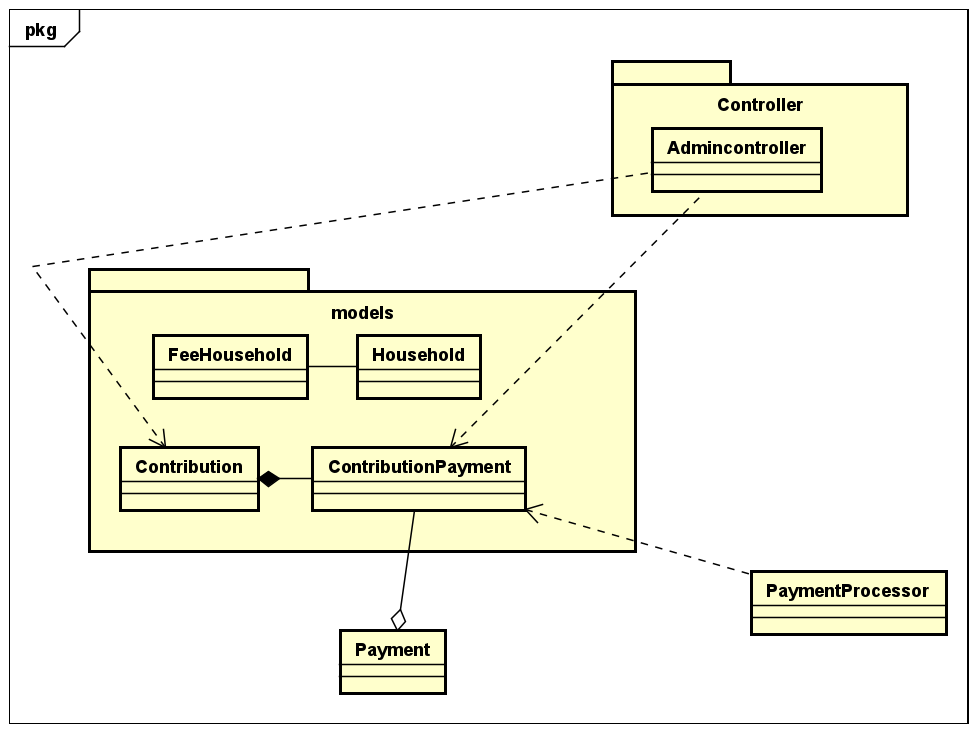
**Household & Vehicle Management:**   
  
****

**Mục đích:** Quản lý thông tin hộ gia đình và phương tiện giao thông.

**Quan hệ chính:**

* **Aggregation:** Household có nhiều User (quan hệ 1-nhiều, User có thể tồn tại độc lập)
* **Composition:** Household sở hữu Vehicle (quan hệ mạnh, Vehicle phụ thuộc vào Household)
* **Dependency:** Các Service phụ thuộc vào Entity để thao tác dữ liệu

**Financial Management:**

****

**Mục đích:**Quản lý các khoản phí, đóng góp và thanh toán của cư dân.

**Quan hệ chính:**

* **Association:**FeeHousehold là bảng trung gian thể hiện quan hệ nhiều-nhiều giữa FeeReference và Household
* **Composition:**Contribution sở hữu nhiều ContributionPayment
* **Inheritance:**ContributionPayment kế thừa từ lớp trừu tượng Payment
* **Dependency:**PaymentProcessor phụ thuộc vào ContributionPayment để xử lý thanh toán

## Thiết kế chi tiết

### Thiết kế cơ sở dữ liệu

Ảnh có chứa biểu đồ, văn bản, hình vẽ, đen và trắng

Nội dung do AI tạo ra có thể không chính xác.

*ERD Quản Lý chung cư*

Phần này có độ dài từ hai đến bốn trang. Sinh viên thiết kế, vẽ và giải thích biểu đồ thực thể liên kết (E-R diagram). Từ đó, sinh viên thiết kế cơ sở dữ liệu tùy theo hệ quản trị cơ sở dữ liệu mà mình sử dụng (SQL, NoSQL, Firebase, v.v.)

Ví dụ: Xác định các thực thể dữ liệu, liên kết và thuộc tính, xây dựng sơ đồ thực thể liên kết 🡪 thiết kế các bảng dữ liệu theo mô hình quan hệ, xây dựng sơ đồ các bảng quan hệ và đặc tả chi tiết cho từng bảng

1. USER

|  |  |  |
| --- | --- | --- |
| Trường | Kiểu dữ liệu | Ràng buộc |
| ID | INT | PRIMARY KEY |
| fullname | VARCHAR | NOT NULL |
| email | VARCHAR | UNIQUE |
| CCCD | VARCHAR | FOREIGN KEY |
| phoneNumber | VARCHAR |  |
| gender | VARCHAR |  |
| dateOfbirth | VARCHAR |  |
| temporaryResident | VARCHAR |  |
| permanentResident | VARCHAR |  |

2. Admin

|  |  |  |
| --- | --- | --- |
| Trường | Kiểu dữ liệu | Ràng buộc |
| ID | INT | PRIMARY KEY |
| fullname | VARCHAR | NOT NULL |
| role | VARCHAR |  |
| phoneNumber | DATE |  |
| phone | VARCHAR |  |
| email | VARCHAR | NOT NULL |
| password | VARCHAR | NOT NULL |
| refreshToken | TEXT |  |

3. Household

|  |  |  |
| --- | --- | --- |
| Trường | Kiểu dữ liệu | Ràng buộc |
| ID | INT | PRIMARY KEY |
| ApartmentNumber | VARCHAR |  |
| area | FLOAT |  |
| isActive | VARCHAR |  |

4. Vehicle

|  |  |  |
| --- | --- | --- |
| Trường | Kiểu dữ liệu | Ràng buộc |
| ID | INT | PRIMARY KEY |
| plateNumber | VARCHAR |  |
| vehicleType | VARCHAR |  |
| pricePerMonth | FLOAT |  |

5. Contribution

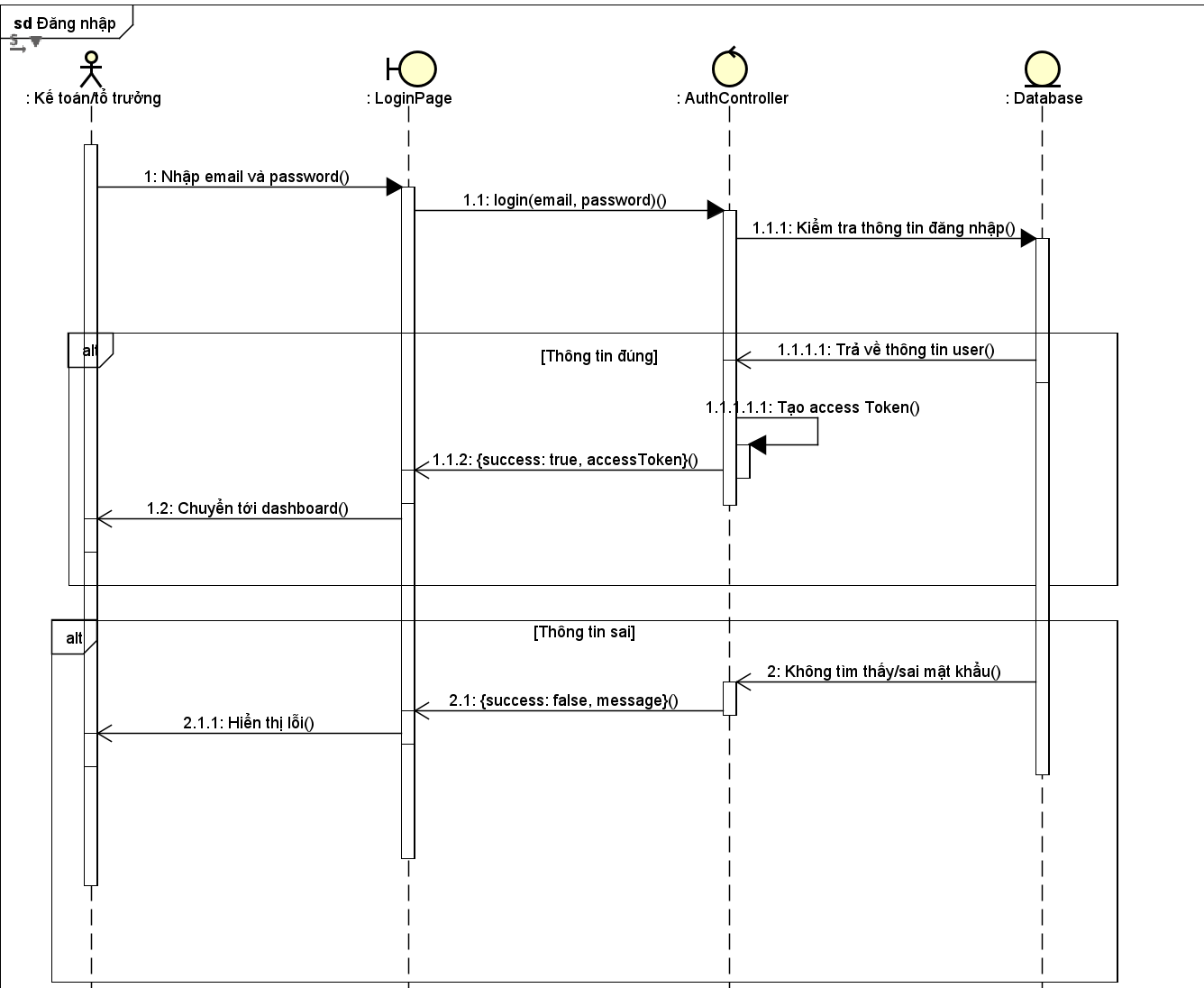
|  |  |  |
| --- | --- | --- |
| Trường | Kiểu dữ liệu | Ràng buộc |
| ID | INT | PRIMARY KEY |
| status | VARCHAR |  |
| name | VARCHAR |  |
| description | VARCHAR |  |
| goal | FLOAT |  |
| donate | FLOAT |  |
| startDate | DATETIME |  |
| endDate | DATETIME |  |

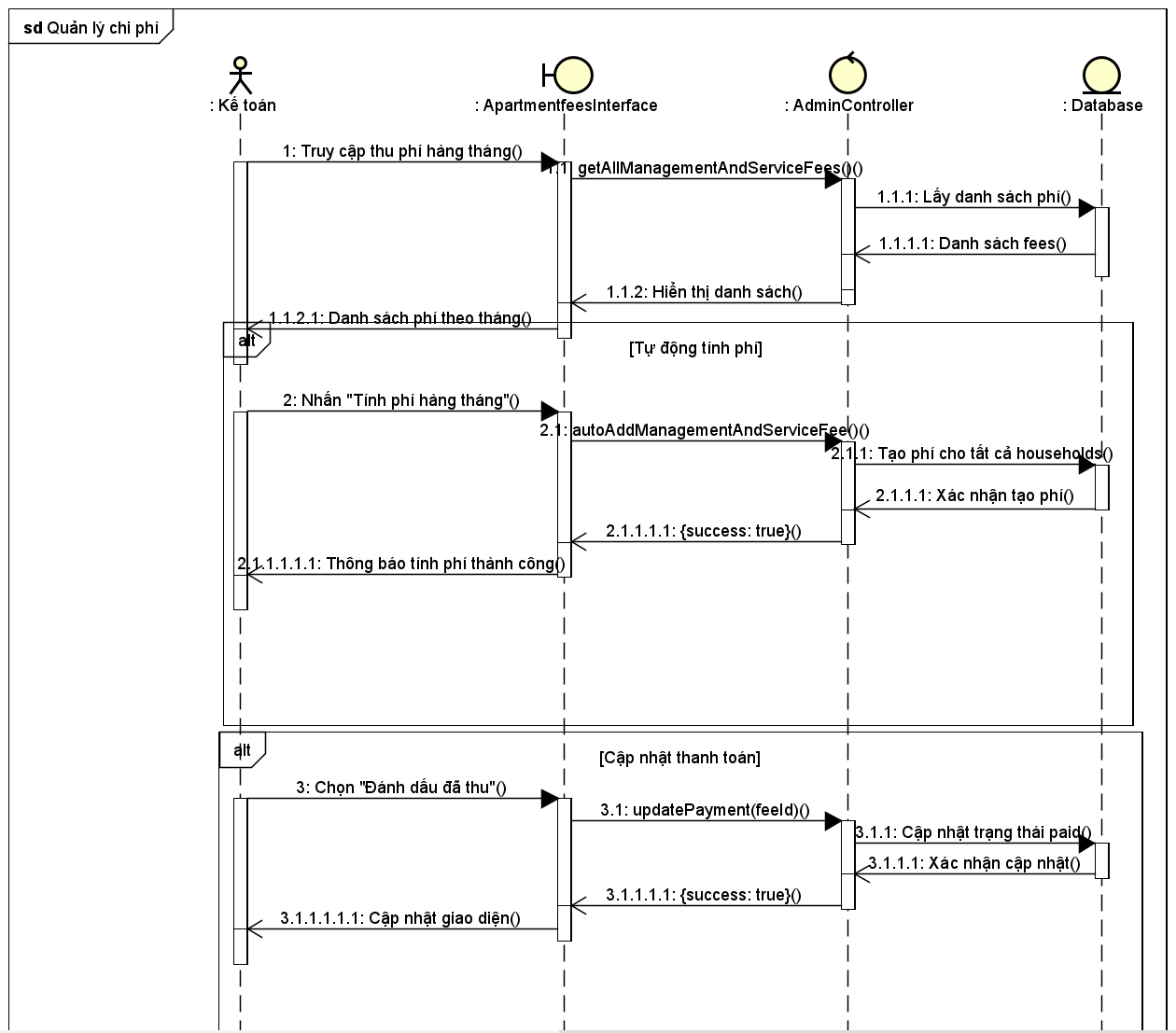
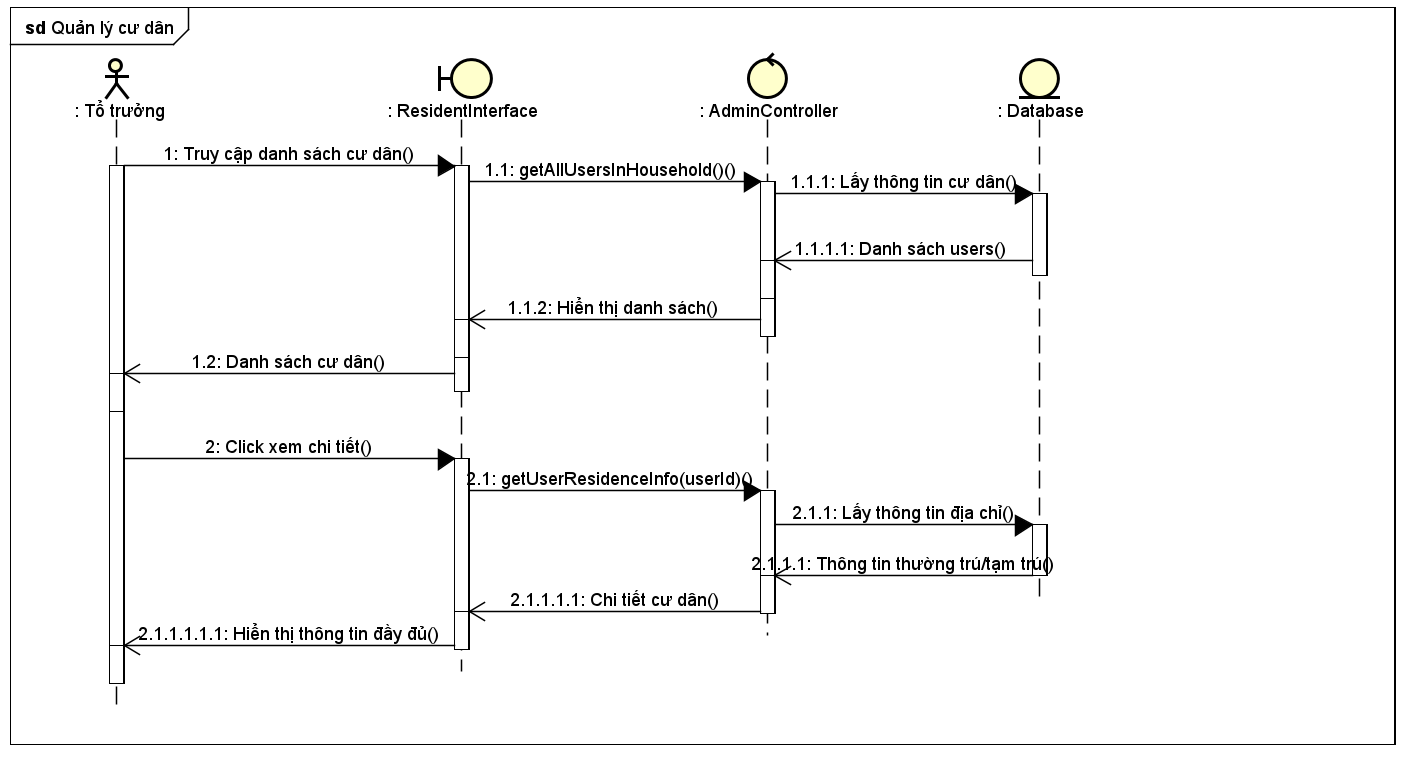
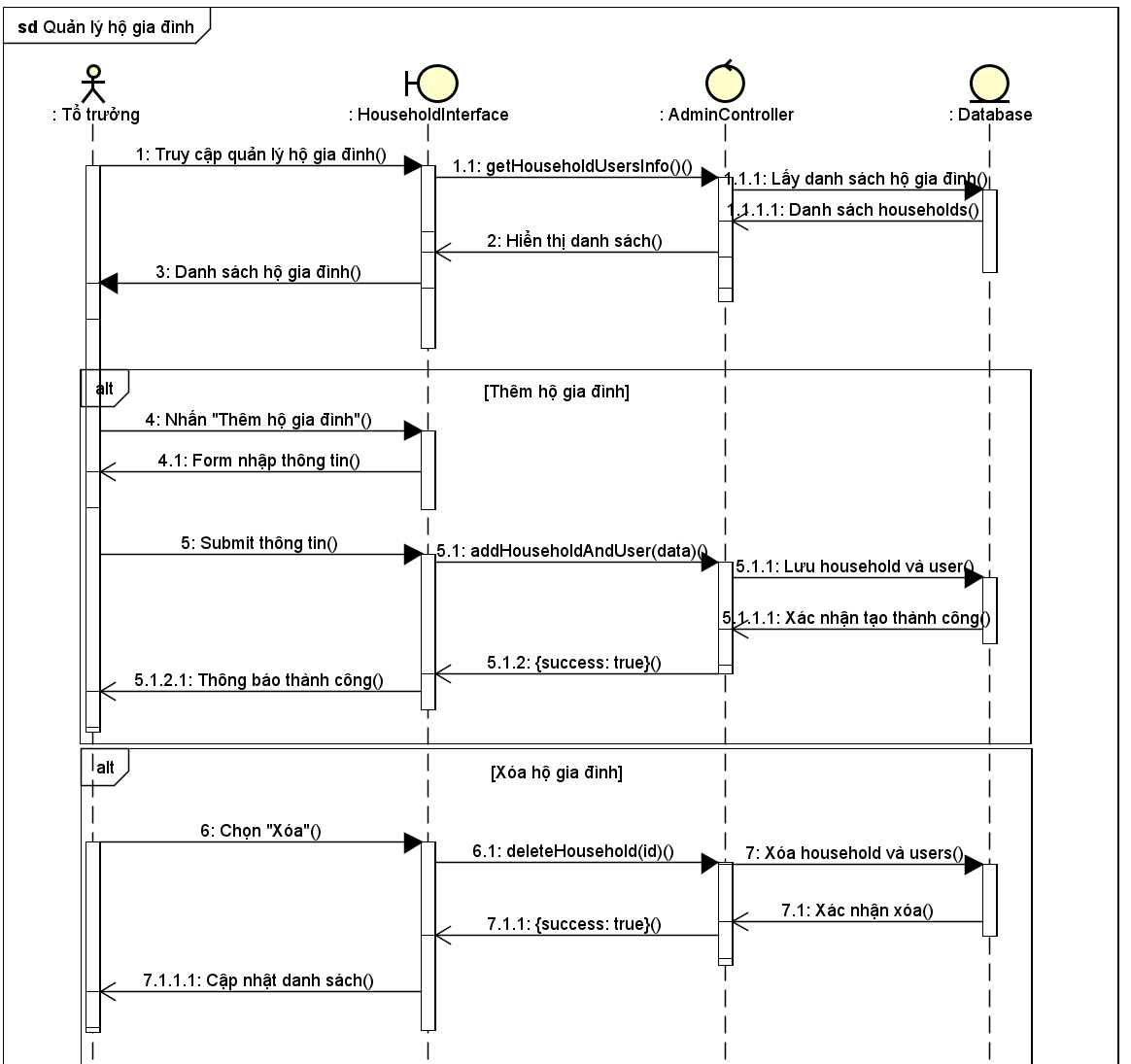
6.FeeService

|  |  |  |
| --- | --- | --- |
| Trường | Kiểu dữ liệu | Ràng buộc |
| ID | INT | PRIMARY KEY |
| unit | VARCHAR |  |
| servicePrice | FLOAT |  |
| type | VARCHAR |  |
| serviceName | VARCHAR |  |
| isRequired | VARCHAR |  |

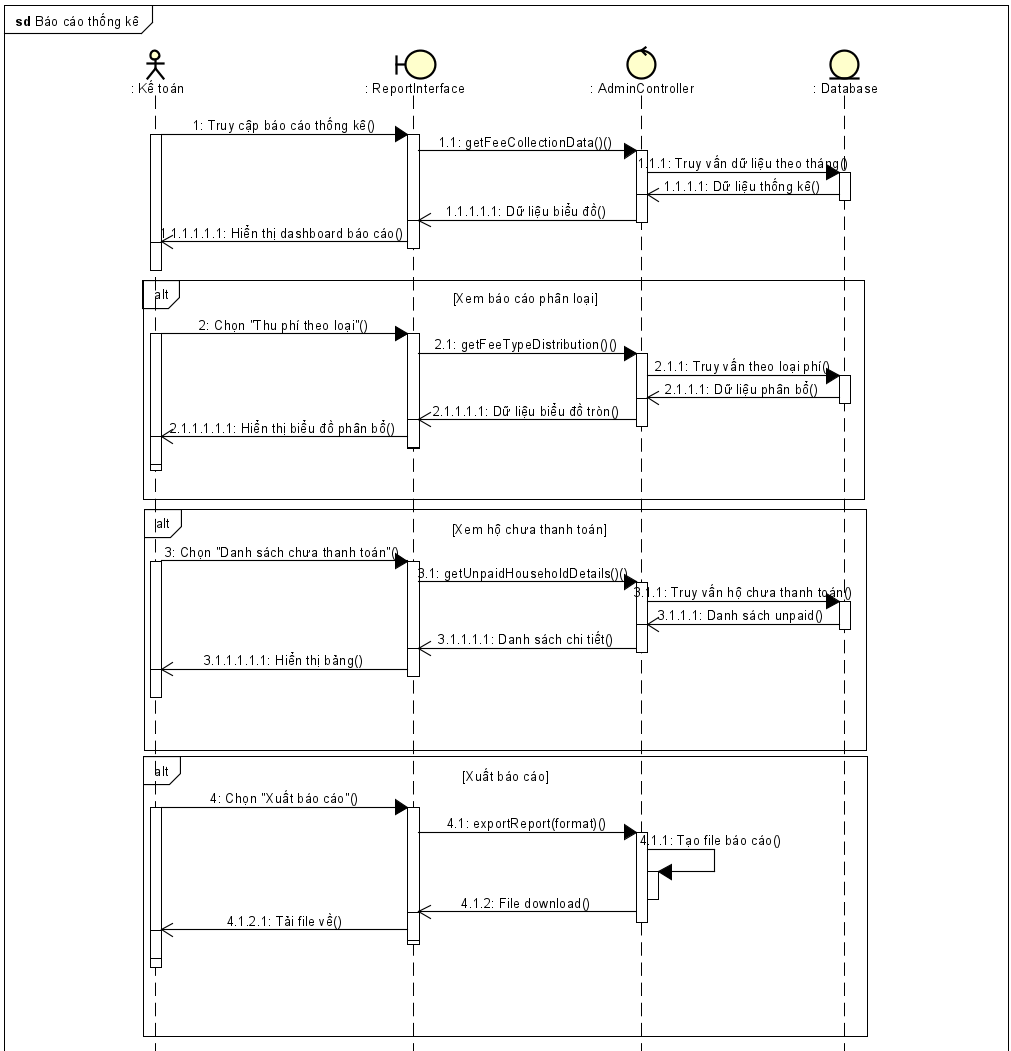
### Thiết kế lớp

* 1. Biểu đồ trình tự đăng nhập



* 1. Biểu đồ trình tự quản lý chi phí  
       
     
  2. Biểu đồ trình tự quản lý cư dân  
       
     
  3. Biểu đồ trình tự quản lý hộ gia đình  
       
     
  4. Biểu đồ trình tự quản lý quyên góp  
       
     A diagram of a project

     AI-generated content may be incorrect.  
       
     6. Biểu đồ trình tự quản lý phương tiện  
       
     A screenshot of a computer

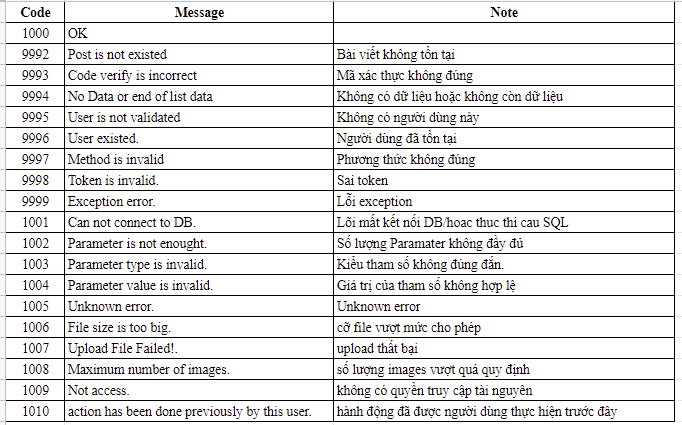
     AI-generated content may be incorrect.  
       
     7. Biểu đồ trình tự báo cáo thống kê  
       
     

### Thiết kế giao diện

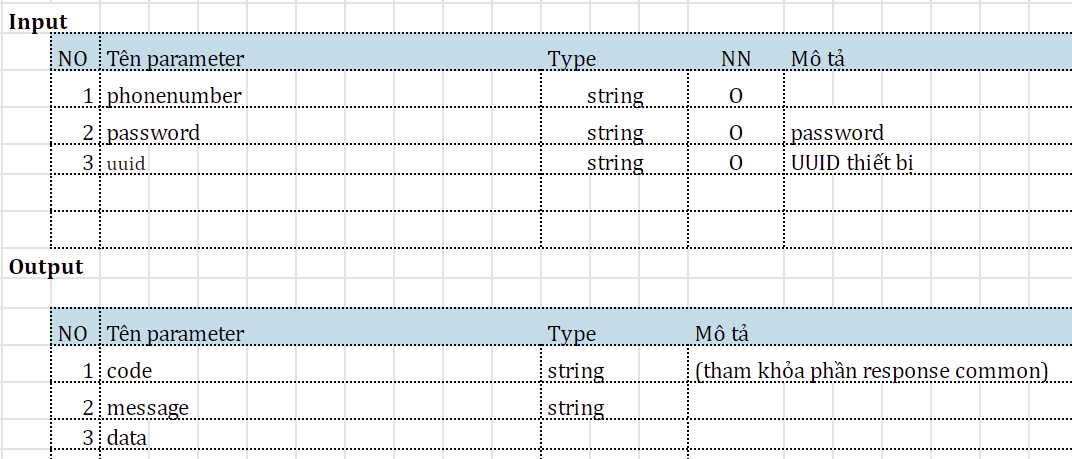
Phần này gồm 2 nội dung:

**Thiết kế giao diện API của ứng dụng**: đặc tả chi tiết các API của phía Backend

* Mô tả tổng quan về danh sách API:
  + Đường dẫn truy cập trên máy local: https://localhost:8080
  + Đường dẫn truy cập trên máy khách: [**https://bluemoon.thaihadtp.id.vn**](https://bluemoon.thaihadtp.id.vn)
* Danh sách mã Response khi xử lý API phía Backend, các mã này được xây dựng theo chức năng / nghiệp vụ của sản phẩm.
  + Ví dụ:



* Đặc tả chi tiết cho từng API, ví dụ:
  + API: /signup
  + Mô tả: API cho phép đăng ký một tài khoản mới của người dùng
    - Request dạng : **POST**
    - Input:
    - **- phonenumber** (số điện thoại của người dùng)
    - **- password** (mật khẩu của người dùng)
  + Kết quả: 1000|OK Nếu đăng ký thành công thì hiển thị giao diện bắt đầu ứng dụng. Nếu lỗi thì gửi mã lỗi kèm thông điệp.
  + Chi tiết input / output (O: bắt buộc, X: không bắt buộc)



**Thiết kế giao diện người dùng GUI**:

* Sinh viên đặc tả thông tin về màn hình mà ứng dụng của mình hướng tới, bao gồm độ phân giải màn hình, kích thước màn hình, số lượng màu sắc hỗ trợ, v.v. Tiếp đến, sinh viên đưa ra các thống nhất/chuẩn hóa của mình khi thiết kế giao diện như thiết kế nút, điều khiển, vị trí hiển thị thông điệp phản hồi, phối màu, v.v.
* Sinh viên xây dựng các thiết kế giao diện mockup cho các màn hình sản phẩm.
* Sau cùng sinh viên đưa ra một số hình ảnh minh họa thiết kế giao diện cho các chức năng. Lưu ý, sinh viên không nhầm lẫn giao diện thiết kế với giao diện của sản phẩm sau cùng.