ĐẠI HỌC BÁCH KHOA HÀ NỘI

Trường Công nghệ thông tin và Truyền thông

Tài liệu thiết kế phần mềm

(Software Design Document – SDD)

Phiên bản 1

<PROJECT / ĐỒ ÁN / GR 15>

Đề tài: <Phần mềm quản lý thu phí chung cư>

Lê Mạnh Tùng - 20225680

Nguyễn Đình Phúc Thành - 20225671

Lý Công Tiến - 20225934

Đoàn Thanh Tùng - 20225946

Nguyễn Thành Vinh - 20225779

Lê Quang Vinh - 20225955

*Hà Nội, ngày 10 tháng 5 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

#### Mục đích của tài liệu

* **Mô tả tổng thể hệ thống:** Tài liệu trình bày chi tiết các yêu cầu chức năng và phi chức năng, kiến trúc tổng thể, các thành phần chủ chốt, quy trình vận hành, cùng với các ràng buộc kỹ thuật cần tuân thủ.
* **Cơ sở cho quá trình phát triển:** Cung cấp nội dung chi tiết hỗ trợ cho nhóm phát triển phần mềm, kiểm thử và triển khai có thể làm việc một cách chính xác và hiệu quả.
* **Đảm bảo tính nhất quán:** Giúp các bên liên quan như chủ đầu tư, quản lý dự án và nhóm phát triển có cái nhìn chung, thống nhất về yêu cầu và cấu trúc hệ thống.
* **Hỗ trợ bảo trì và mở rộng:** Đóng vai trò là tài liệu tham khảo chính thức phục vụ cho các hoạt động bảo trì, nâng cấp và mở rộng hệ thống trong tương lai.

#### Đối tượng sử dụng tài liệu

Tài liệu này được xây dựng dành cho các đối tượng sau:

* **Chủ đầu tư và ban quản lý chung cư:** Giúp nắm rõ cách hệ thống vận hành cũng như lợi ích mà hệ thống mang lại trong công tác quản lý.
* **Nhóm phát triển phần mềm:** Cung cấp thông tin đầy đủ về yêu cầu và kiến trúc hệ thống để triển khai đúng mục tiêu đề ra.
* **Nhóm kiểm thử (QA):** Là cơ sở để xây dựng và thực hiện các kịch bản kiểm thử phù hợp với chức năng hệ thống.
* **Nhóm triển khai và hỗ trợ kỹ thuật:** Hỗ trợ quá trình cài đặt, cấu hình hệ thống và hỗ trợ người dùng cuối một cách hiệu quả.
* **Người dùng cuối (cư dân và ban quản lý):** Có thể tham khảo để hiểu về các tính năng chính của hệ thống và cách nó hỗ trợ trong hoạt động quản lý tòa nhà.

## Phạm vi

Phần mềm quản lý thu phí tại chung cư BlueMoon được xây dựng nhằm hỗ trợ Ban quản trị trong việc quản lý hiệu quả các khoản thu từ cư dân, bao gồm phí dịch vụ, phí quản lý và các khoản đóng góp tự nguyện. Hệ thống cho phép tạo và quản lý các đợt thu phí, theo dõi tiến độ thanh toán và tổng quan tình hình thu chi của chung cư.

Ngoài chức năng thu phí, phần mềm còn quản lý thông tin hộ khẩu, nhân khẩu, bao gồm thành viên hộ gia đình, tình trạng tạm trú, tạm vắng và các thay đổi liên quan, phục vụ công tác báo cáo với cơ quan chức năng.

Hệ thống cũng hỗ trợ phân quyền người dùng, kiểm soát tài khoản của nhân viên và tổ trưởng, đảm bảo an toàn và bảo mật. Các tính năng tra cứu, tìm kiếm và thống kê giúp Ban quản trị dễ dàng theo dõi hoạt động thu chi và tình hình cư dân, từ đó nâng cao hiệu quả quản lý.

## Từ điển thuật ngữ

<Đưa ra các thuật ngữ và giải thích cho thuật ngữ đó trong nghiệp vụ của phần mềm, và được sử dụng trong tài liệu này. Không được tự ý mặc định về kinh nghiệm hoặc kiến thức của người đọc>

## Tài liệu tham khảo

<Liệt kê danh sách tài liệu tham khảo được sử dụng trong tài liệu này, bao gồm các tài liệu liên quan đến dự án>

# Thiết kế kiến trúc

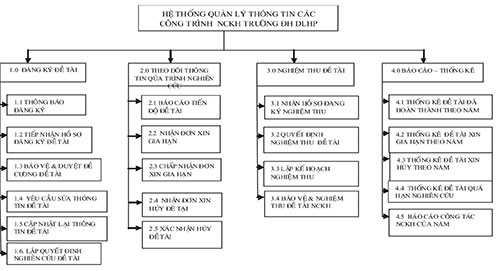
### Lựa chọn kiến trúc phần mềm

Mục này có độ dài từ một đến ba trang. Sinh viên cần lựa chọn kiến trúc phần mềm cho ứng dụng của mình như: kiến trúc ba lớp MVC, MVP, SOA, Microservice, v.v. rồi giải thích sơ bộ về kiến trúc đó (không giải thích chi tiết/dài dòng).

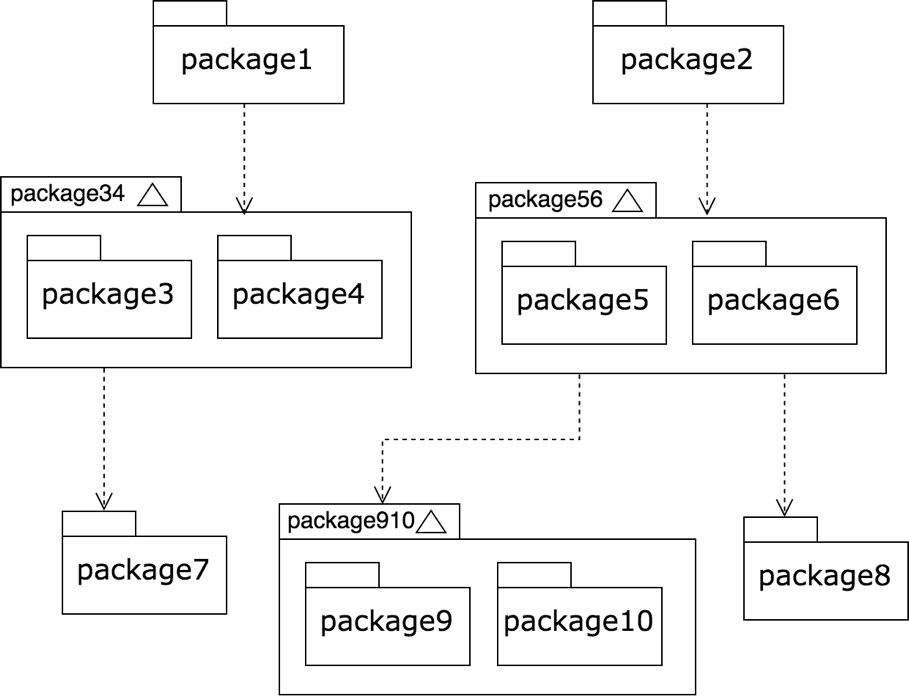
Sử dụng kiến trúc phần mềm đã chọn ở trên, sinh viên mô tả kiến trúc cụ thể cho ứng dụng của mình. Gợi ý: sinh viên áp dụng lý thuyết chung vào hệ thống/sản phẩm của mình như thế nào, có thay đổi, bổ sung hoặc cải tiến gì không. Ví dụ, thành phần M trong kiến trúc lý thuyết MVC sẽ là *những thành phần cụ thể nào* (ví dụ: là interface I + class C1 + class C2, v.v.) trong kiến trúc phần mềm của sinh viên.

### Thiết kế tổng quan

Sinh viên xây dựng sơ đồ kiến trúc phân cấp các chức năng và mô tả chi tiết cho các chức năng, quan hệ giữa các thành phần. Ví dụ:



Sinh viên vẽ biểu đồ gói UML (UML package diagram), nêu rõ sự phụ thuộc giữa các gói (package). SV cần vẽ các gói sao cho chúng được phân theo các tầng rõ ràng, không được sắp đặt package lộn xộn trong hình vẽ. Sinh viên chú ý các quy tắc thiết kế (Các gói không phụ thuộc lẫn nhau, gói tầng dưới không phụ thuộc gói tầng trên, không phụ thuộc bỏ qua tầng, v.v.) và cần giải thích sơ lược về mục đích/nhiệm vụ của từng package. SV tham khảo ví dụ minh họa trong Hình 1.



**Hình 1 Ví dụ biểu đồ phụ thuộc gói**

### Thiết kế chi tiết gói

Sinh viên thiết kế và lần lượt vẽ biểu đồ thiết kế cho từng package, hoặc một nhóm các package liên quan để giải quyết một vấn đề gì đó. Khi vẽ thiết kế gói, sinh viên chỉ cần đưa tên lớp, không cần chỉ ra các thành viên phương thức và thuộc tính. SV tham khảo ví dụ minh họa trong Hình 2.

Sinh viên cần vẽ rõ ràng quan hệ giữa các lớp trong biểu đồ. Các quan hệ bao gồm: phụ thuộc (dependency), kết hợp (association), kết tập (aggregation), hợp thành (composition), kế thừa (inheritance), và thực thi (implementation). Các quan hệ này đều đã được minh họa trong Hình 2.

Sau khi vẽ hình minh họa, sinh viên cần giải thích ngắn gọn về thiết kế của mình.



**Hình 2 Ví dụ thiết kế gói**

## Thiết kế chi tiết

### Thiết kế cơ sở dữ liệu

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.)

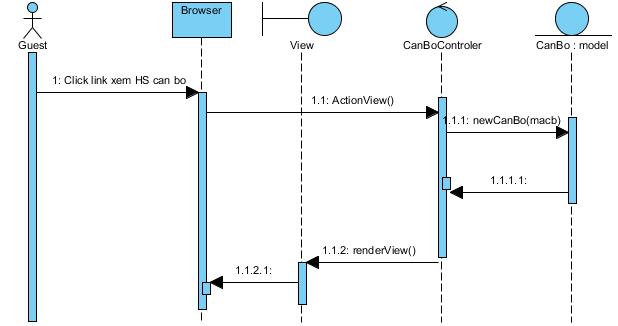
Ví dụ: Xác định các thực thể dữ liệu, liên kết và thuộc tính, xây dựng sơ đồ thực thể liên kết 🡪 thiết kế các bảng dữ liệu theo mô hình quan hệ, xây dựng sơ đồ các bảng quan hệ và đặc tả chi tiết cho từng bảng

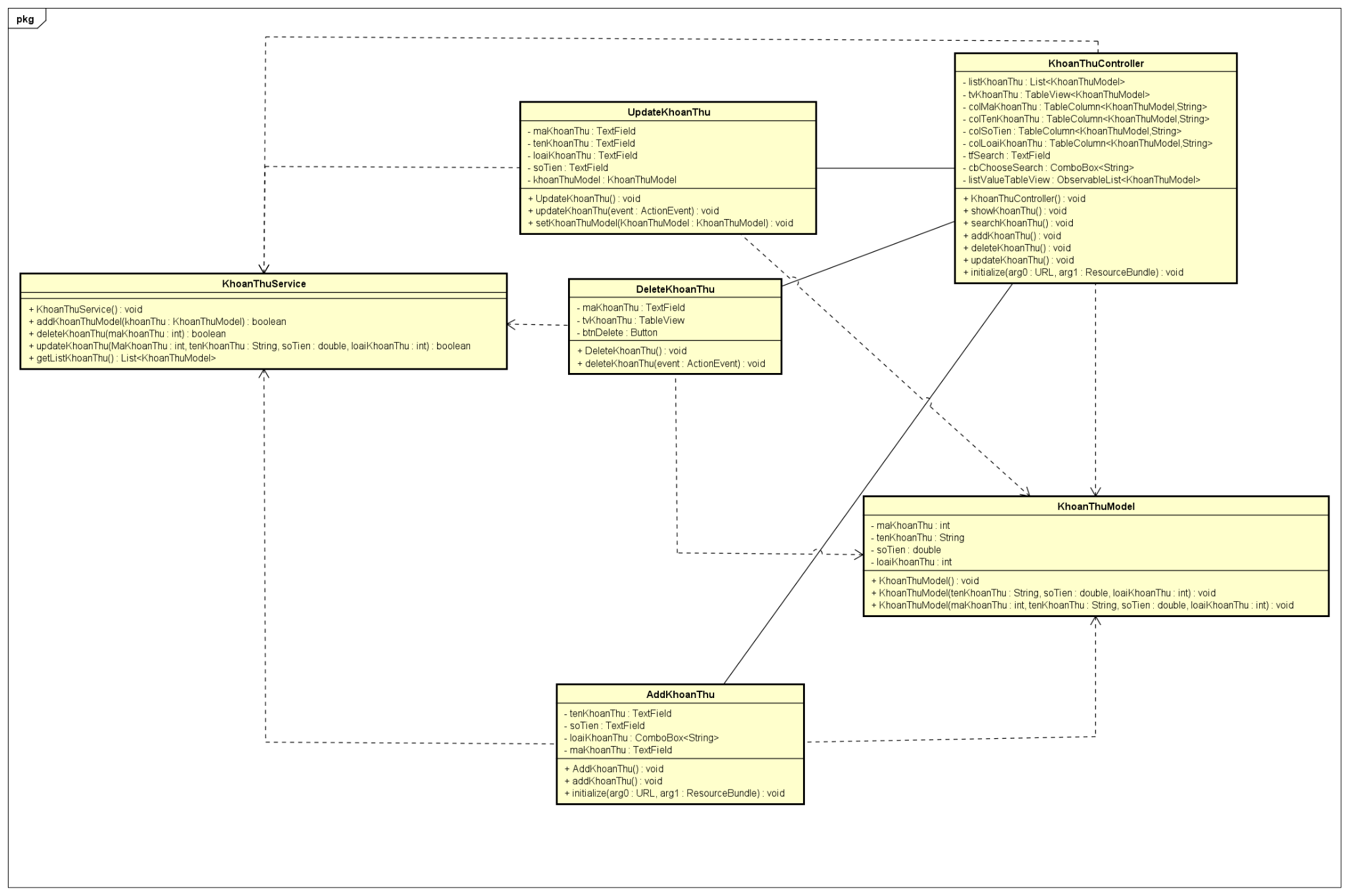
| **dc: Bảng Địa chỉ** | | | | |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Ghi chú** | **Ràng buộc** |
| 1 | Ma | Int | Khóa định danh | Khoá chính |
| 2 | Huyen | String | Quận/Huyện |  |
| 3 | Tinh | String | Thành phố/Tỉnh |  |

### Thiết kế lớp

Sinh viên trình bày thiết kế chi tiết các thuộc tính và phương thức cho các lớp của ứng dụng và sơ đồ lớp.

Để minh họa thiết kế lớp, sinh viên thiết kế luồng truyền thông điệp giữa các đối tượng tham gia bằng biểu đồ trình tự (hoặc biểu đồ giao tiếp). Ví dụ:



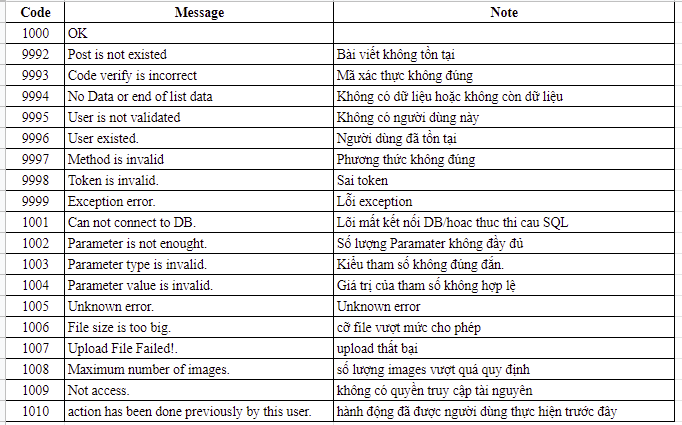
****

### Thiết kế giao diện

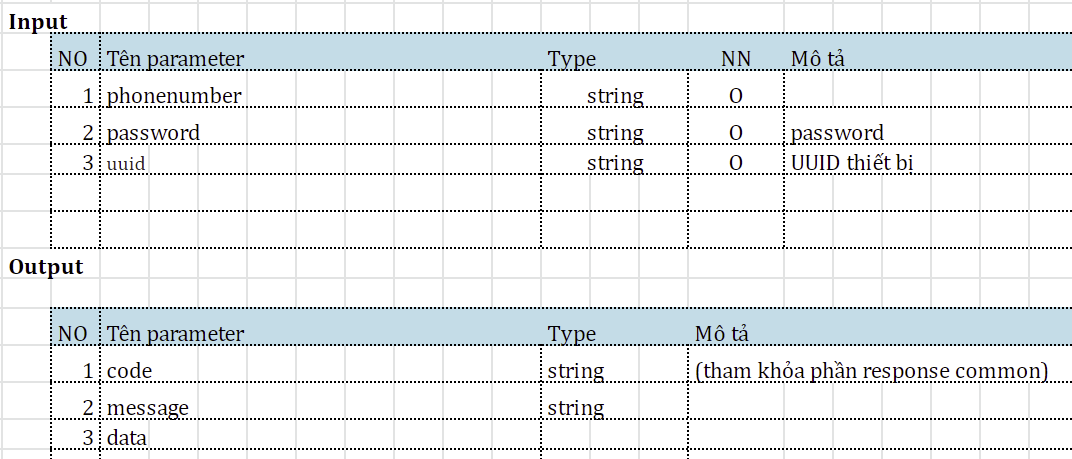
Phần này gồm 2 nội dung:

**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:
  + Ví dụ: Đường dẫn cơ bản: [**https://ABC.def/it4788/**](https://abc.def/it4788/)
  + Tên miền ABC.def  sinh viên tự thiết lập, cho phép local
  + Với việc sử dụng API Login thì truy cập [**https://ABC.def/it4788/login**](https://abc.def/it4788/login&)**?**...  với các tham số đi kèm
* 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)



**2.2.3.2 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.