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 <số>

<PROJECT / ĐỒ ÁN / GR XX>

Đề tài: <Tên đề tài>

<Thông tin sinh viên>

*Hà Nội, ngày … tháng … năm …*

**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

<Đưa ra mục đích của tài liệu và đối tượng đọc tài liệu>

## Phạm vi

<Đưa ra mô tả sơ bộ về giải pháp thiết kế của bài toán>

## 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).

Kiến trúc **MVC (Model – View – Controller)** là một trong những kiến trúc phần mềm phổ biến và hiệu quả, đặc biệt phù hợp với các ứng dụng web có giao diện người dùng phức tạp, nhiều vai trò và chức năng rõ ràng. MVC chia hệ thống thành ba thành phần chính:

* **Model (M):** Quản lý dữ liệu và logic nghiệp vụ của ứng dụng. Đây là nơi xử lý các thao tác liên quan đến cơ sở dữ liệu, nghiệp vụ như kiểm tra trạng thái phí, cập nhật thông tin hộ khẩu,...
* **View (V):** Là lớp giao diện người dùng – hiển thị thông tin và nhận tương tác từ người dùng.
* **Controller (C):** Là cầu nối giữa View và Model. Controller nhận dữ liệu từ View (thông qua hành động người dùng), xử lý hoặc gọi Model để xử lý, sau đó trả kết quả về View để hiển thị.

Ưu điểm của kiến trúc MVC là:

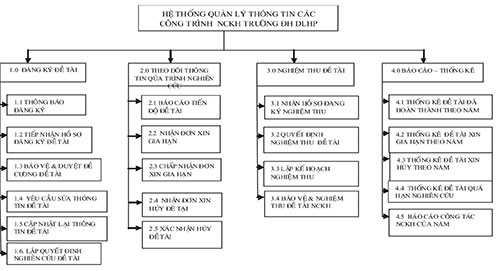
* Tăng tính **phân tách trách nhiệm** trong hệ thống.
* Dễ **bảo trì, nâng cấp và mở rộng** hệ thống.
* Cho phép nhiều **vai trò người dùng** với giao diện khác nhau nhưng dùng chung logic nghiệp vụ.

Do đó, nhóm chọn kiến trúc MVC cho hệ thống phần mềm BlueMoon để đảm bảo hệ thống dễ mở rộng, bảo trì và phù hợp với nghiệp vụ phức tạp, phân quyền đa vai trò.

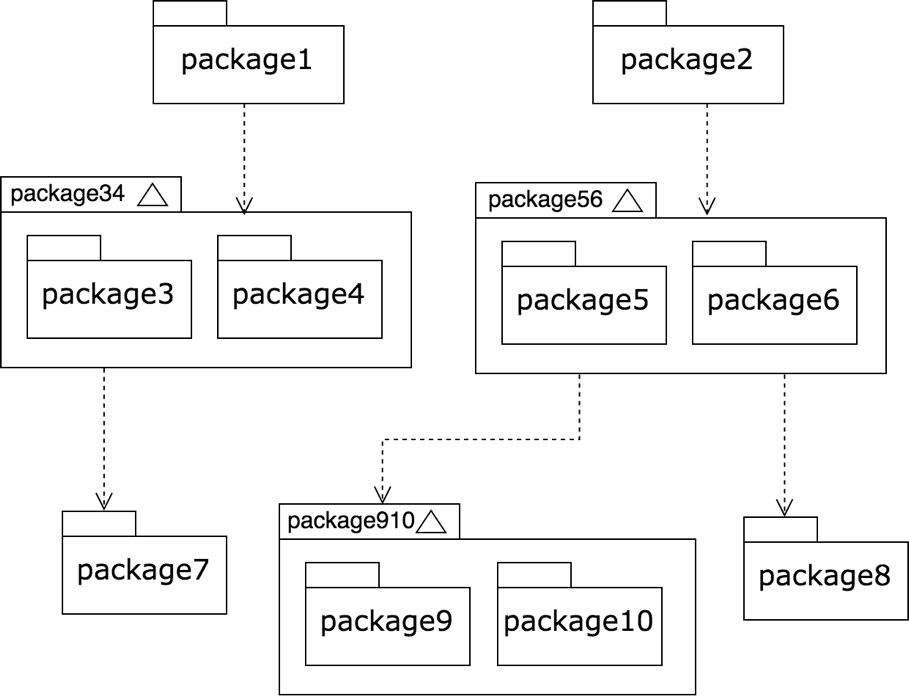
Sử dụng kiến trúc phần mềm đã chọn ở trên, sinh viên mô tả kiến trúc cụ thể cho ứng dụng của mình. Gợi ý: sinh viên áp dụng lý thuyết chung vào hệ thống/sản phẩm của mình như thế nào, có thay đổi, bổ sung hoặc cải tiến gì không. Ví dụ, thành phần M trong kiến trúc lý thuyết MVC sẽ là *những thành phần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

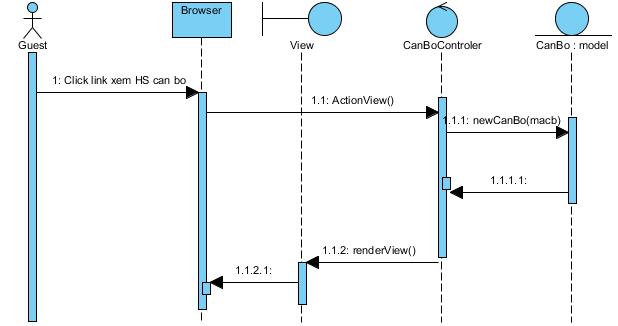
Ví dụ: Xác định các thực thể dữ liệu, liên kết và thuộc tính, xây dựng sơ đồ thực thể liên kết 🡪 thiết kế các bảng dữ liệu theo mô hình quan hệ, xây dựng sơ đồ các bảng quan hệ và đặc tả chi tiết cho từng bảng

| **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:

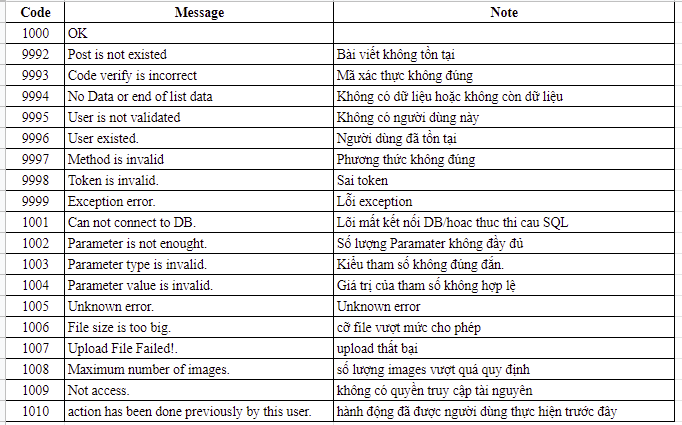


### Thiết kế giao diện

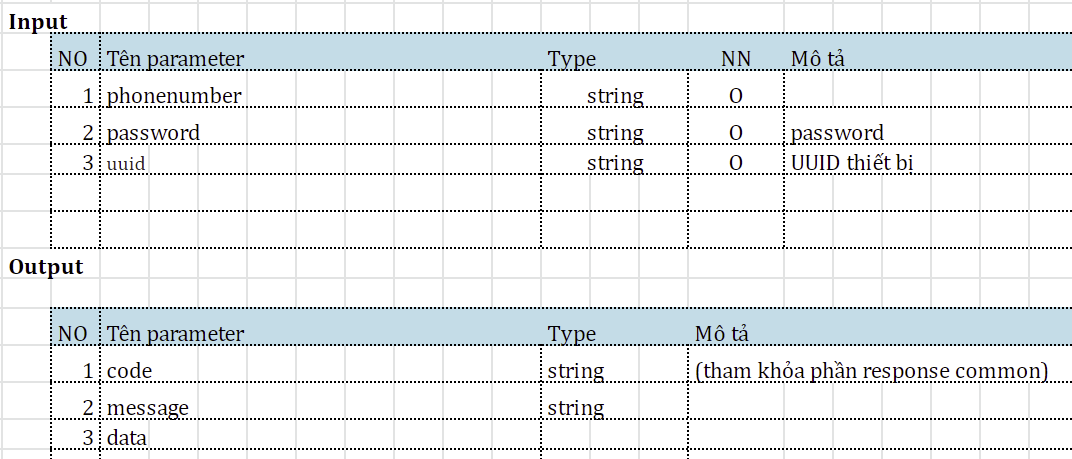
Phần này gồm 2 nội dung:

**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)



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

**1. Đặc tả thông tin về màn hình mục tiêu:**

* **Độ phân giải màn hình:** 1920x1080 (Full HD)
* **Kích thước màn hình:** từ 13 đến 24 inch
* **Số lượng màu hỗ trợ:** 16 triệu màu (True Color – 24-bit)
* **Tỷ lệ khung hình:** 16:9
* **Thiết bị hỗ trợ bổ sung:**
  + Mobile: 375x812px (iPhone X), 360x640px (Android chuẩn)
  + Tablet: 768x1024px (iPad chuẩn)

**2. Chuẩn hóa khi thiết kế giao diện:**

**a. Thiết kế nút (Button)**

* Hình dạng: bo tròn nhẹ (border-radius: 8px)
* Kích thước chuẩn:
  + Nút chính: 140px x 45px
  + Nút phụ: 110px x 35px
* Màu sắc:
  + Nút chính: nền xanh lá cây (#28a745), chữ trắng (#ffffff)
  + Nút phụ: nền xám nhạt (#6c757d), chữ trắng (#ffffff)
* Hiệu ứng hover: sáng hơn 10% màu nền ban đầu.

**b. Điều khiển (Controls)**

* Input: bo viền nhẹ (border-radius: 6px), padding đều 8px, placeholder rõ ràng
* Dropdown/Select: hiển thị rõ mũi tên, hiển thị không quá 5 item/lần
* Checkbox/Radio: dễ bấm, kích thước tối thiểu 20px, viền bo nhẹ (border-radius: 4px)

**c. Thông điệp phản hồi (Feedback messages)**

* Thông báo lỗi: chữ đỏ (#dc3545), icon dấu chấm than (!) bên cạnh, hiển thị bên dưới trường nhập liệu
* Thông báo thành công: chữ xanh lá cây (#28a745), icon tích (✓) bên cạnh, xuất hiện phía trên cùng màn hình trong 2-3 giây
* Vị trí hiển thị thông báo chính: phía trên form hoặc chính giữa màn hình

**d. Phối màu (Color scheme)**

* Màu chủ đạo: xanh nước biển (# 007BFF)
* Màu nền chính: trắng (#ffffff)
* Màu nền phụ: xám nhạt (#f8f9fa)
* Màu phụ trợ: xám đậm (#6c757d)
* Màu cảnh báo: đỏ (#dc3545), vàng cam (#ffc107)

3. Thiết kế mockup giao diện

| Screen Layout | System name | Bluemoon | | Date | 2025/6/3 |
| --- | --- | --- | --- | --- | --- |
| Screen ID/name | BA00100/ Màn hình đăng nhập | | | | |
|  | | | | | |

| Screen Layout | System name | Bluemoon | | Date | 2025/6/3 |
| --- | --- | --- | --- | --- | --- |
| Screen ID/name | BA00101/ Màn hình Dashboard dân cư | | | | |
|  | | | | | |

| Screen Layout | System name | Bluemoon | | Date | 2025/6/3 |
| --- | --- | --- | --- | --- | --- |
| Screen ID/name | BA00102/ Màn hình Dashboard quản lý | | | | |
|  | | | | | |

| Screen Layout | System name | Bluemoon | | Date | 2025/6/3 |
| --- | --- | --- | --- | --- | --- |
| Screen ID/name | BA00103/ Màn hình Quản lý dân cư | | | | |
|  | | | | | |

| Screen Layout | System name | Bluemoon | | Date | 2025/6/3 |
| --- | --- | --- | --- | --- | --- |
| Screen ID/name | BA00104/ Màn hình Quản lý phí | | | | |
|  | | | | | |

| Screen Layout | System name | Bluemoon | | Date | 2025/6/3 |
| --- | --- | --- | --- | --- | --- |
| Screen ID/name | BA00105/ Màn hình Danh sách dân cư | | | | |
|  | | | | | |

| Screen Layout | System name | Bluemoon | | Date | 2025/6/3 |
| --- | --- | --- | --- | --- | --- |
| Screen ID/name | BA00106/ Màn hình Quản lý hóa đơn | | | | |
|  | | | | | |