**TRƯỜNG ĐẠI HỌC PHENIKAA**

**KHOA CÔNG NGHỆ THÔNG TIN**

**⸎⸎⸎⸎⸎**

A logo of a university

AI-generated content may be incorrect.

**BÁO CÁO BÀI TẬP LỚN**

**HỌC PHẦN PHÂN TÍCH VÀ THIẾT KẾ PHẦN MỀM**

**Đề tài : Xây dựng App hiến máu nhân đạo (Heart Beat)**

Nhóm thực hành : Nhóm 9

Thành viên : Hà Mạnh Long – 23010390

Bùi Anh Quốc – 23010328

Giáo viên hướng dẫn : Vũ Quang Dũng

Nguyễn Xuân Quế

Lớp tín chỉ : PT&TKPM(N05)

**HÀ NỘI, THÁNG 10/2025**

# Lời mở đầu

Trong bối cảnh xã hội hiện đại, nhu cầu về máu phục vụ cấp cứu và điều trị y tế ngày càng gia tăng, hiến máu nhân đạo đã trở thành một hoạt động mang ý nghĩa nhân văn sâu sắc, góp phần cứu sống hàng triệu sinh mạng mỗi năm. Tuy nhiên, thực tế cho thấy công tác tổ chức, quản lý và kết nối giữa người hiến máu, các cơ sở y tế và ban tổ chức sự kiện vẫn còn gặp nhiều khó khăn, thiếu tính đồng bộ và hiệu quả.

Nhận thấy nhu cầu thực tế này, nhóm chúng em đã quyết định thực hiện đề tài “Xây dựng ứng dụng hiến máu nhân đạo”. Mục tiêu của dự án là phát triển một nền tảng số giúp kết nối cộng đồng người hiến máu với các bệnh viện, trung tâm y tế và ban tổ chức sự kiện hiến máu. Thông qua ứng dụng, người hiến máu có thể dễ dàng đăng ký tham gia, theo dõi lịch sử hiến máu của bản thân, nhận thông báo nhắc nhở và quản lý lịch hẹn. Đồng thời, nhân viên y tế và quản trị viên có công cụ để quản lý sự kiện, xác nhận đăng ký, cũng như thống kê và lưu trữ dữ liệu một cách khoa học.

Trong khuôn khổ môn học Phân tích & Thiết kế Phần mềm, dự án này là cơ hội để nhóm áp dụng những kiến thức đã học vào một quy trình phát triển hệ thống hoàn chỉnh – từ phân tích yêu cầu, thiết kế mô hình, đến triển khai và kiểm thử. Báo cáo sẽ trình bày chi tiết toàn bộ quá trình thực hiện, những thách thức nhóm đã gặp phải, giải pháp được đề xuất, và định hướng mở rộng trong tương lai.

Chúng em kỳ vọng sản phẩm không chỉ mang lại hiệu quả thiết thực trong công tác tổ chức hiến máu mà còn góp phần lan tỏa tinh thần nhân đạo, xây dựng một cộng đồng gắn kết và bền vững vì sức khỏe cộng đồng.

**Mục lục**

[Lời mở đầu 2](#_Toc209731927)

[**I.** **Tuần 1 – Phân tích yêu cầu:** 4](#_Toc209731928)

[**1.** **Actors (Vai trò):** 4](#_Toc209731929)

[**2.** **Tổng quan hệ thống:** 4](#_Toc209731930)

[**3.** **Yêu cầu chức năng:** 5](#_Toc209731931)

[**4.** **Yêu cầu phi chức năng:** 5](#_Toc209731932)

[**II.** **Tuần 2 – Mô hình hóa UseCase & Kịch bản:** 6](#_Toc209731933)

[**1.** **Biểu đồ UseCase:** 6](#_Toc209731934)

[**2.** **Biểu đồ hoạt động (Activity Diagram):** 7](#_Toc209731935)

[**3.** **Kịch bản UseCase chi tiết:** 10](#_Toc209731936)

[**III.** **Tuần 3 – Thiết kế sơ đồ lớp & Tạo cơ sở code:** 12](#_Toc209731937)

[**1.** **Thiết kế sơ đồ lớp (Class Diagram):** 12](#_Toc209731938)

[**2.** **Cơ sở Code:** 16](#_Toc209731939)

[**IV.** **Tuần 4 – Thiết kế tương tác:** 16](#_Toc209731940)

[**1.** **Biểu đồ trình tự (Sequence Diagram):** 16](#_Toc209731941)

[**2.** **Thiết kế giao diện (Figma):** 22](#_Toc209731942)

[**V.** **Tuần 5 – Thiết kế hành vi và trạng thái:** 23](#_Toc209731943)

[**1.** **Biểu đồ trạng thái (State Machine Diagram):** 23](#_Toc209731944)

[**2.** **Lập trình:** 26](#_Toc209731945)

[**VI.** **Tuần 6 – Thiết kế kiến trúc hệ thống:** 26](#_Toc209731946)

[**1.** **Biểu đồ Gói (Package Diagram):** 26](#_Toc209731947)

[**2.** **Lập trình:** 27](#_Toc209731948)

[**VII.** **Tuần 7 – Áp dụng mẫu thiết kế (Design Patterns):** 28](#_Toc209731949)

[**1.** **Áp dụng Pattern:** 28](#_Toc209731950)

[**2.** **Lập trình:** 28](#_Toc209731951)

[**VIII.** **Tuần 8 – Lập trình chức năng lõi:** 28](#_Toc209731952)

[**1.** **Lập trình chức năng “”:** 28](#_Toc209731953)

[**IX.** **Tuần 9 – Lập trình giao diện và tích hợp:** 28](#_Toc209731954)

[**1.** **Lập trình giao diện:** 28](#_Toc209731955)

[**2.** **Tích hợp:** 28](#_Toc209731956)

[**X.** **Tuần 10 – Kiểm thử và báo cáo:** 28](#_Toc209731957)

[**1.** **Kiểm thử đơn vị (Unit Test):** 28](#_Toc209731958)

[**2.** **Báo cáo:** 28](#_Toc209731959)

# **Tuần 1 – Phân tích yêu cầu:**

## **Actors (Vai trò):**

* **Người hiến máu (Donor):**

+ Đăng nhập/ Đăng ký (Login/Register).

+ Đăng ký/ Hủy đăng ký hiến máu (Register/Cancel Blood Donation).

+ Xem lịch hẹn hiến máu (View Appointment).

+ Xem lịch sử hiến máu (View Donation History).

* **Nhân viên y tế (Staff):**

+ Đăng nhập/ Đăng ký (có mã code riêng do QTV cấp).

+ Quản lý lịch hiến máu (Manage Donation Schedule).

+ Xác nhận đăng ký hiến máu(Confirm Donation Registration).

+ Ghi nhận kết quả hiến máu (Record Results).

* **Quản trị viên (Admin):**

+ Đăng nhập (Login).

+ Quản lý nhân sự (cấp mã code riêng cho NVYT).

+ Quản lý người hiến máu (Manage Donors).

+ Quản lý sự kiện hiến máu (Manage Donation Events).

+ Quản lý địa điểm hiến máu (Manage Locations).

* **Hệ thống (System):**

+ Kiểm tra điều hiện hiến máu (độ tuổi, sức khỏe, khoảng cách ngày hiến máu trước đó,...).

+ Gửi thông báo nhắc nhở lịch hẹn/ địa điểm hiến máu.

+ Lưu trữ và backup dữ liệu

## **Tổng quan hệ thống:**

* **Mục tiêu hệ thống:**

+ Kết nối người hiến máu, nhân viên y tế và ban tổ chức.

+ Hỗ trợ quy trình đăng ký, quản lý, nhắc nhở và lưu trữ lịch sử.

* **Phạm vi hệ thống:**

+ Đối tượng phục vụ: người hiến máu, nhân viên y tế, quản trị viên.

+ Nền tảng: Android/ Web/ Mobile-first.

* **Các ràng buộc:**

+ Cần Internet.

+ Phải tuân thủ quy định y tế về bảo mật thông tin cá nhân.

* **Giả định và phụ thuộc:**

+ Người dùng có thiết bị kết nối Internet.

+ Người hiến máu cung cấp thông tin chính xác khi đăng ký.

## **Yêu cầu chức năng:**

- Hệ thống cho phép người hiến máu tạo tài khoản và đăng ký hiến máu trong các sự kiện còn chỗ.

- Người hiến máu có thể hủy đăng ký trong thời gian cho phép.

- Nhân viên y tế có thể quản lý danh sách đăng ký, xác nhận và cập nhật trạng thái hiến máu.

- Quản trị viên có thể quản lý thông tin sự kiện, địa điểm và người hiến máu, nhân viên y tế.

- Hệ thống gửi thông báo nhắc nhở trước lịch hiến máu ít nhất 1 ngày.

- Hệ thống cho phép người hiến máu tra cứu lịch sử hiến máu của bản thân.

## **Yêu cầu phi chức năng:**

- **Bảo mật**: Bảo vệ thông tin cá nhân và y tế, xác thực đăng nhập an toàn.

- **Hiệu năng**: Thời gian phản hồi tra cứu sự kiện ≤ 2 giây.

- **Sẵn sàng**: Hệ thống hoạt động ổn định, có thể phục vụ ≥ 500 người dùng cùng lúc.

- **Khả năng mở rộng**: Có thể mở rộng số lượng sự kiện và địa điểm dễ dàng.

- **Dễ dùng**: Giao diện rõ ràng, hỗ trợ mobile-first.

- **Bảo toàn dữ liệu**: Sao lưu định kỳ và khôi phục nhanh khi có sự cố.

**- Khả năng truy cập**: Hỗ trợ đa ngôn ngữ (Việt – Anh).

# **Tuần 2 – Mô hình hóa UseCase & Kịch bản:**

## **Biểu đồ UseCase:**

A diagram of a system

AI-generated content may be incorrect.

## **Biểu đồ hoạt động:**

* **Admin:**

A diagram of a server

AI-generated content may be incorrect.

* **Staff:**

A diagram of a flowchart

AI-generated content may be incorrect.

* **Donor:**

A diagram of a flowchart

AI-generated content may be incorrect.

## **Kịch bản UseCase chi tiết:**

* 1. **UseCase: Đăng ký hiến máu:**

- **Actor chính:** Người hiến máu.

**- Mục tiêu:** Người hiến máu đăng ký tham gia một đợt hiến máu thành công.

**- Tiền điều kiện:** Người hiến máu đã có tài khoản hoặc thực hiện đăng nhập.

**- Luồng chính/ Thành công:**

1. Người hiến máu mở ứng dụng và chọn chức năng ***Đăng ký hiến máu***.

2. Ứng dụng yêu cầu nhập thông tin cá nhân (họ tên, ngày sinh, số CMND/CCCD, nhóm máu, chọn địa điểm, thời gian,...).

3. Hệ thống sẽ ***kiểm tra điều kiện hiến máu*** (tuổi, tình trạng sức khỏe, khoảng cách lần hiến máu trước,...).

4. Nếu hợp lệ, hệ thống lưu thông tin đăng ký.

5. Hệ thống hiện thị thông báo: “Đăng ký hiến máu thành công”.

6. Hệ thống ***gửi thông báo nhắc nhở*** trước lịch hẹn.

**- Trường hợp đặc biệt/ Đăng ký không thành công:**

**+** ***Người hiến máu không đủ điều kiện:***

1. Tại bước 3, hệ thống phát hiện điều kiện không đạt (Ví dụ: chưa đủ tuổi, khoảng cách hiến máu lần trước quá gần,...).
2. Hệ thống hiện thị thông báo: “Bạn chưa đủ điều kiện hiến máu”.
3. Hệ thống quay lại màn hình chính.

***+ Địa điểm/Thời gian hiến máu đã đầy chỗ:***

1. Tại bước 2, sau khi chọn địa điểm hệ thống thông báo: “Lịch hẹn tại điểm này đã đầy.”.

2. Người hiến máu được yêu cầu chọn địa điểm hoặc thời gian khác.

* 1. **UseCase: Xác nhận đăng ký hiến máu:**

**- Actor chính:** Nhân viên y tế.

**- Mục tiêu:** Xác nhận đăng ký của người hiến máu hợp lệ.

**- Tiền điều kiện:** Người hiến máu đã gửi yêu cầu đăng ký thành công.

**- Luồng chính/ Thành công:**

1. Nhân viên y tế đăng nhập vào hệ thống.

2. Nhân viên y tế chọn ***Danh sách đăng ký hiến máu.***

3. Nhân viên y tế mở hồ sơ một người hiến máu.

4. Hệ thống sẽ hiện thị thông tin + lịch sử hiến máu.

5. Nhân viên y tế ***xác nhận nếu đủ điều kiện***.

6. Hệ thống cập nhật trạng thái: “Đã xác nhận”.

7. Người hiến máu nhận được ***thông báo xác nhận***.

**- Trường hợp đặc biệt/ Người hiến máu không đủ điều kiện:**

***+ Người hiến máu không đủ điều kiện về sức khỏe:***

1. Tại bước 4, nhân viên y tế thấy người hiến máu không đạt điều kiện (ví dụ: huyết áp không phù hợp,...).
2. Nhân viên chọn “Từ chối vì không đủ yêu cầu sức khỏe”.
3. Hệ thống lưu trạng thái và gửi thông báo từ chối đến người hiến máu.
   1. **UseCase: Ghi nhận kết quả hiến máu:**

**- Actor chính:** Nhân viên y tế.

**- Mục tiêu:** Lưu kết quả hiến máu sau khi người hiến máu hoàn tất.

**- Luồng chính/Thành công:**

1. Sau khi hiến máu, nhân viên y tế mở ứng dụng, chọn ***Ghi nhận kết quả hiến máu***.

2. Nhập thông tin: số lương máu lấy được, nhóm máu, tình trạng sau khi hiến vào hồ sơ người hiến máu.

3. Hệ thống lưu kết quả vào hồ sơ.

4. Hệ thống ***cập nhật vào lịch sử*** hiến máu.

5. Hệ thống tự động lưu trữ & backup dữ liệu.

**- Trường hợp đặc biệt/Lỗi lưu kết quả:**

1. Tại bước 3, hệ thống gặp sự cố lỗi kết nối.

2. Nếu có sự cố hệ thống sẽ thông báo: “Không thể lưu kết quả, vui lòng thử lại.”.

3. Nhân viên y tế có thể lưu lại qua sổ khám và khi hệ thống ổn định sẽ lưu lại sau.

# **Tuần 3 – Thiết kế sơ đồ lớp & Tạo cơ sở code:**

## **Thiết kế sơ đồ lớp:**

1. **Các lớp chính:**

**- User (lớp cha trừu tượng):**

**+ Attribute (thuộc tính):** id, name, role

**+ Method (phương thức):** login(), signUp()

**+ Validate (xác thực):** id không được rỗng, name 2-50 ký tự, role thuộc {admin, staff, user}.

**- Admin ( kế thừa User):**

**+ Method (phương thức):** manageLocation(List<Hospital>), manageStaff(List<Staff>), viewDonationHistoryByEvent(eventId)

**+ Validate (xác thực):** Chỉ role = “admin” mới được truy cập.

**- Donor ( kế thừa User):**

**+ Attribute (thuộc tính):** dateOfBirth, age, bloodGroup, city, gender, phoneNumber, willingToDonate, about, avatar, donationHistiory

**+ Method (phương thức):** registerDonateBlood(), cancelRegistrationDonate(), viewHistory(), viewAppointment()

**+ Validate (xác thực):**

**- Staff ( kế thừa User):**

**+ Attribute (thuộc tính):** staffId, code

**+ Method (phương thức):** confirmRegistration(), manageEvent(Event), manageDonor(List<Donor>)

**+ Validate (xác thực):** mỗi 1 staff có 1 mã code riêng, chỉ có role = “staff” mới có thể truy cập.

**- Hospital:**

**+ Attribute (thuộc tính):** hospitalId, hospitalName, imgUrl, address, phone, province, district, createAt, updateAt, eventList.

**+ Validate (xác thực):**

**- Event:**

**+ Attribute (thuộc tính):** id, name, description, location, date, time, deadline, donorList, capacity, donorCount.

**+ Method (phương thức):** updateDonorCount()

**+ Validate (xác thực):**

**- DonateBlood:**

**+ Attribute (thuộc tính):** userId, donor, nationalIdCard(CCCD), status.

**+ Method (phương thức):** addToEvent(), deleteFromEvent().

**+ Validate (xác thực):** nationalIdCard đúng định dạng 12 số,

**- UserDonationHistory:**

**+ Attribute (thuộc tính):** id, donorId, donorFullName, quantityInMI, status, event.

**+ Method (phương thức):** getDonorById(donorId)

**+ Validate (xác thực):**

**- SystemService (Helper/ Service)**

**+ Attribute (thuộc tính):** eventList: List<Event>.

**+ Method (phương thức):** Validation(), Notification(), backUpData().

1. **Mối quan hệ giữa các lớp:**

**- Kế thừa:**

+ Admin, Donor, Staff kế thừa từ User.

**- Admin – Staff:**

+ 1 Admin có thể quản lý nhiều Staff (1-n).

**- Admin – Hospital:**

+ 1 Admin có thể quản lý nhiều Hospital (1-n).

**- Hospital – Event:**

+ 1 Hospital có thể quản lý nhiều Event (1-n).

+ 1 Event chỉ thuộc về 1 Hospital.

**- Event – DonateBlood – Donor:**

+ Mỗi quan hệ n – m giữa Donor và Event thông qua DonateBlood.

+ 1 Donor có thể tham gia nhiều Event.

+ 1 Event có thể có nhiều Donor.

**- Staff – Event:**

+ 1 Staff có thể quản lý nhiều Event (1-n).

**- Staff – Donateblood:**

+ 1 Staff có thể xác nhận nhiều đơn đăng ký hiến máu (1-n).

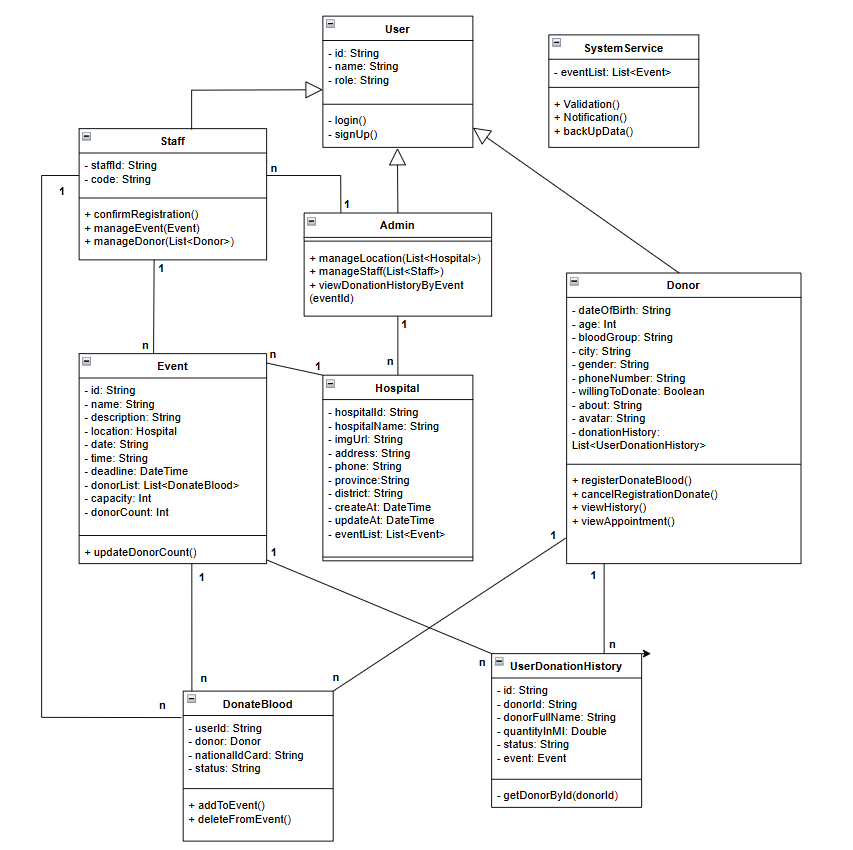
**- Donor – UserDonationHistory:**

+ 1 Donor có thể có nhiều lịch sử hiến máu (1-n).

**- Event – UserDonationHistory:**

+ 1 Event có thể có nhiều bản ghi lịch sử hiến máu (1-n).

1. **Biểu đồ Lớp (Class Diagram):**



## **Cơ sở Code:**

# **Tuần 4 – Thiết kế tương tác:**

## **Biểu đồ trình tự:**

1. **Login & SignUp:**

A screenshot of a computer screen

AI-generated content may be incorrect.

A diagram of a sign up

AI-generated content may be incorrect.

1. **Register/Cancel Donation Blood:**

A screenshot of a diagram

AI-generated content may be incorrect.

A diagram with text and arrows

AI-generated content may be incorrect.

1. **Staff confirm registration:**

A diagram of a software flowchart

AI-generated content may be incorrect.

1. **Staff manage event & donor:**

A diagram of a service

AI-generated content may be incorrect.

A diagram with arrows and text

AI-generated content may be incorrect.

1. **Donor views history & appointment:**

A diagram of a service

AI-generated content may be incorrect.

1. **System service:**

A diagram of a software project

AI-generated content may be incorrect.

## **Thiết kế giao diện (Figma):**

* **Link thiết kế giao diện Figma:** <https://www.figma.com/design/nlb29AHZX4q6bgxKT03d6j/Heart-Beat-Figma?node-id=0-1&m=dev&t=zltqMDcrCJK8UzzN-1>.
* **Có thể xem tại phần Design trên Github:**

<https://github.com/AnhQuocs/PTTKPM25->26\_ClassN05\_Nhom\_9

# **Tuần 5 – Thiết kế hành vi và trạng thái:**

## **Biểu đồ trạng thái (State Machine Diagram):**

* **Donate Blood Stage Machine(mô tả lifecycle của 1 lượt đăng**

**ký của donor cho 1 event):**

+ NotRegistered: Donor chưa đăng ký hiến máu.

+ Pending: Donor vừa đăng ký, chờ Staff xác nhận.

+ Confirmed: Staff duyệt đăng ký thành công

+ Cancelled: Donor tự hủy hoặc bị từ chối.

+ Completed: Donor tham gia hiến máu thành công.

+ Archived: backUp dữ liệu, lưu trữ vào lịch sử.

A diagram of a system

AI-generated content may be incorrect.

* **User State Machine (Admin/Staff/Donor – mô tả trạng thái tài khoản):**

+ NewAccount: tạo tài khoản mới (chưa verify hoặc active).

+ Active: tài khoản hoạt động (có thể thực hiện các chức năng tùy thuộc vào role = {admin, staff, donor}).

+ Suspended: tài khoản bị tạm khóa(admin suspend).

+ Deactivated: tài khoản bị xóa/ vô hiệu hóa(user/admin delete)

A diagram of a computer flowchart

AI-generated content may be incorrect.

* **Event State Machine(quản lý trạng thái của 1 event):**

+ Scheduled: event được tạo bởi admin (chưa mở đăng ký).

+ Open: đang mở cho phép đăng ký hiến máu.

+ Full: đã đủ chỗ/ đủ số lượng không nhận thêm.

+ Closed: đã đóng đăng ký hoặc sự kiện đã kết thúc.

+ Archived: backUp/lưu trữ.

A diagram of a flowchart

AI-generated content may be incorrect.

## **Lập trình:**

# **Tuần 6 – Thiết kế kiến trúc hệ thống:**

## **Biểu đồ Gói (Package Diagram):**

1. **Mối quan hệ giữa các tầng:**

* **UI → UseCase:**

**-** UI nhận thao tác từ người dùng (click, nhập dữ liệu,..).

**-** Sau đó gọi đến UseCase để xử lý logic.

* **UseCase → Repository:**

- UseCase không truy cập dữ liệu trực tiếp.

- Gọi Repository để lấy dữ liệu từ API, Firebase hoặc cache...(calls).

* **Repository → Domain Entities:**

**-** Repository chịu trách nhiệm ánh xạ (mapping) dữ liệu từ nguồn ngoài thành Domain Entity.

**-** Đồng thời lưu trữ (persist) các entity vào CSDL khi cần.

* **Repository** ..|> **UseCase:**

**-** Repository trả kết quả về cho UseCase(return).

**-** Đảm bảo dữ liệu mà UseCase nhận được luôn ở dạng entity chuẩn.

* **UseCase ..|> Domain Entities:**

- UseCase thao tác với entity (đọc, validate,..).

- Nhưng không thay đổi định nghĩa hay cấu trúc của entity(read-only/ DTO).

* **Quan hệ cụ thể (Ví dụ):**

- AuthUseCase → AuthRepository để xử lý đăng nhập/đăng ký.

- EventUseCase → EventRepository để lấy danh sách sự kiện.

- AdminUseCase → AdminRepository để quản lý người dùng.

1. **Biểu đồ gói:**

A diagram of a computer system

AI-generated content may be incorrect.

## **Lập trình:**

# **Tuần 7 – Áp dụng mẫu thiết kế (Design Patterns):**

## **Áp dụng Pattern:**

## **Lập trình:**

# **Tuần 8 – Lập trình chức năng lõi:**

### **Lập trình chức năng “”:**

# **Tuần 9 – Lập trình giao diện và tích hợp:**

## **Lập trình giao diện:**

## **Tích hợp:**

# **Tuần 10 – Kiểm thử và báo cáo:**

## **Kiểm thử đơn vị (Unit Test):**

## **Báo cáo:**