



## Nhóm 7-Đào Công Đoàn - sssss

Nhập môn công nghệ phần mềm (Trường Đại học Công nghiệp Hà Nội)

**BỘ CÔNG THƯƠNG**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  
**KHOA CÔNG NGHỆ THÔNG TIN**

=====\*\*\*=====



**BÁO CÁO BÀI TẬP LỚN HỌC PHẦN**  
**NHẬP MÔN CÔNG NGHỆ PHẦN MỀM**

**ĐỀ TÀI: Tìm hiểu về mô hình quy trình phát triển phần mềm xoắn ốc và  
áp dụng trong dự án phát triển phần mềm quản lý thu học phí cho  
trường đại học Công Nghiệp Hà Nội**

**Giảng viên hướng dẫn: Vũ Đình Minh**  
**Lớp: 2021.2IT608200.2**

**Nhóm: Nhóm 7**

**THÀNH VIÊN:**

1. Trịnh Đắc Quang Anh - 2020606697
2. Đỗ Duy Bảo - 2020607609
3. Phạm Đức Công - 2020606251
4. Trần Tiến Cường - 2020607490
5. Vũ Phát Đạt - 2020606789
6. Đào Công Đoàn - 2020606491
7. Bùi Như Thành – 2020606935

Hà Nội, 2022

## Mục lục

Mục lục.....	2
Lời nói đầu.....	2
Danh mục hình vẽ.....	3
Mở đầu.....	4
1. Lý do chọn chủ đề/đề tài:.....	4
2. Mục tiêu nghiên cứu:.....	4
3. Kết quả mong muốn đạt được của đề tài:.....	4
4. Nội dung và phạm vi nghiên cứu.....	4
5. Bố cục của báo cáo.....	4
<b>Chương 1: Giới thiệu về dự án phần mềm.....</b>	<b>6</b>
1.1. Giới thiệu chung.....	6
1.2. Mô tả bài toán.....	6
1.2.1. Quản lý thủ quỹ.....	6
1.2.2. Quản lý sinh viên.....	6
1.2.3. Quản lý việc đóng tiền.....	6
1.3. Giới thiệu về mô hình quy trình xoắn ốc và công nghệ áp dụng trong hệ thống thu tiền học của trường đại học công nghiệp Hà Nội.....	7
1.3.1. Mô hình quy trình xoắn ốc.....	7

1.3.2. Phân tích mô hình.....	7
1.3.3. Giới thiệu công cụ Rational Rose.....	9
1.3.4 Phương pháp phân tích và thiết kế phần mềm.....	10
1.4. Kết luận chương 1.....	10
<b>Chương 2: Tài liệu đặc tả phần mềm quản lý thu học phí Đại học Công Nghiệp Hà Nội.....</b>	<b>11</b>
2.1. Liệt kê các tác nhân và ca sử dụng.....	11
2.1.1. Tác nhân.....	11
2.1.2. Use case.....	11
2.2. Biểu đồ các UC và biểu đồ khung cảnh hệ thống.....	12
2.2.1. Biểu đồ khung cảnh hệ thống.....	12
2.2.2. Biểu đồ use case chính.....	13
2.2.3. Biểu đồ use case Sinh Viên.....	15
2.2.4. Biểu đồ use case thủ quỹ.....	16
2.2.5. Biểu đồ use case ban quản lý.....	17
2.3. Đặc tả ca sử dụng.....	18
2.3.1. Đăng nhập.....	18
2.3.2. Nạp tiền.....	19
2.3.3. Thanh toán.....	20
2.3.4. Kiểm Tra Tiền.....	21
2.3.5 Kiểm tra lịch sử giao dịch.....	22
2.3.6. Trợ giúp.....	23
2.3.7. Hiện thị tín chỉ.....	24
2.3.8. Cập nhật tín chỉ.....	25

2.3.9. Cập nhật tiền.....	25
2.3.10. In hóa đơn.....	26
2.3.11. Thông báo lịch sử giao dịch.....	27
2.3.12. Thống kê.....	28
2.4. Biểu đồ lĩnh vực lớp.....	29
<b>Chương 3: Đặc tả và thiết kế phần mềm.....</b>	<b>30</b>
3.1. Kiến trúc phần mềm.....	30
3.2. Thiết kế cấu trúc.....	30
3.3. Thiết kế giao diện.....	31
3.4. Thiết kế cơ sở dữ liệu.....	31
<b>KẾT LUẬN.....</b>	<b>32</b>

### Lời nói đầu

Trong môn Nhập môn công nghệ phần mềm này, nhóm sinh viên chúng em đã tìm hiểu về đề tài: tìm hiểu về mô hình quy trình phát triển phần mềm xoắn ốc và tài liệu phần mềm quản lý thu học phí. Qua một thời gian tìm hiểu đề tài, chúng em đã làm bài báo cáo Thực tập cơ sở này để báo cáo lại những kiến thức, những kinh nghiệm mà chúng em đã tìm ra được trong đề tài.

Chúng em xin cảm ơn thầy giáo viên hướng dẫn đã tận tình giúp đỡ và hướng dẫn chúng em trong quá trình chúng em tìm hiểu và làm báo cáo về nội dung. Sau khi làm bản báo cáo này đã có thể giúp chúng em có thể hiểu hơn

về những kiến thức cơ bản về quy trình phát triển phần mềm ở các doanh nghiệp, từ đó chúng em có thể có nhiều định hướng về chuyên ngành hơn.

Nhóm sinh viên thực hiện!

#### Danh mục hình vẽ

Hình 1.1	Hình ảnh mô hình thác nước
Hình 1.2	Biểu đồ mức cảnh
Hình 1.3	Biểu đồ use case chính sinh viên
Hình 1.4	Hình ảnh use case chính thủ quỹ
Hình 1.5	Biểu đồ use case chính ban quản lý
Hình 2.1	: Biểu đồ UC Sinh Viên
Hình 2.2	Biểu đồ UC thủ quỹ
Hình 2.3	Biểu đồ UC ban quản lý
Hình 3.1	Biểu đồ lĩnh vực lớp
Hình 3.2	Mô hình client server



## Mở đầu

### 1. Lý do chọn chủ đề/đề tài:

Nhóm em mong muốn được học hỏi về cách sử dụng các phần mềm như rational rose hay mockups, sử dụng được các pha trong các mô hình quy trình vào việc phát triển phần mềm, sử dụng ý tưởng này cho phần mềm đóng học phí

### 2. Mục tiêu nghiên cứu:

- Giải thích được các hoạt động và trình tự thực hiện của các hoạt động trong mô hình quy trình phát triển phần mềm xoắn ốc

- Sử dụng được phương pháp mô hình hóa trong phân tích và thiết kế phần mềm

Sử dụng được công cụ rational rose trong tài liệu đặc tả phân tích và thiết kế phần mềm

### 3. Kết quả mong muốn đạt được của đề tài:

- Sử dụng được các công cụ như rational rose trong tài liệu phân tích và đặc tả thiết kế phần mềm

- Giải thích được các hoạt động và trình tự thực hiện của các hoạt động trong mô hình quy trình phát triển phần mềm xoắn ốc

- Sử dụng được phương pháp mô hình hóa trong phân tích và thiết kế phần mềm

### 4. Nội dung và phạm vi nghiên cứu

- mô hình quy trình phát triển phần mềm xoắn ốc
- công cụ rational rose, mockups trong đặc tả và phân tích thiết kế phần mềm

### 5. Bố cục của báo cáo



Ngoài phần mở đầu, kết luận, phần nội dung của báo cáo gồm 3 chương

- Chương 1: trình bày về mô hình quy trình xoắn ốc và ứng dụng trong phát triển phần mềm thu tiền học của trường Đại học Công Nghiệp Hà Nội.
- Chương 2: Trình bày về bài đặc tả yêu cầu phần mềm thu tiền học của trường Đại học Công Nghiệp Hà Nội.
- Chương 3: Trình bày về tài liệu thiết kế phần mềm thu tiền học của trường Đại học Công Nghiệp Hà Nội.

## **Chương 1: Giới thiệu về dự án phần mềm**

### **1.1. Giới thiệu chung**

Ngày nay, xã hội càng phát triển thì việc học càng đóng một vai trò quan trọng với việc làm của mỗi người.

Để ứng dụng thương mại điện tử và phát triển việc giao thương qua mạng của nhà trường, phần mềm thu học phí trực tuyến trên nền tảng web. Không những vậy, hệ thống còn cho phép quản lý nhân lực và quá trình đóng tiền học thông qua web. Điều này giúp cho nhà trường tiết kiệm được nhân công và sinh viên tiết kiệm được thời gian

### **1.2. Mô tả bài toán**

#### **1.2.1. Quản lý thủ quỹ**

Mỗi thủ quỹ đều có một tài khoản gồm account và password để đăng nhập hệ thống. Nhân viên được lưu trữ thông tin thủ quỹ bao gồm mã thủ quỹ, tên thủ quỹ, địa chỉ, số điện thoại, vị trí, lương. Các thủ quỹ được quản lý bởi ban quản lý

#### **1.2.2. Quản lý sinh viên**

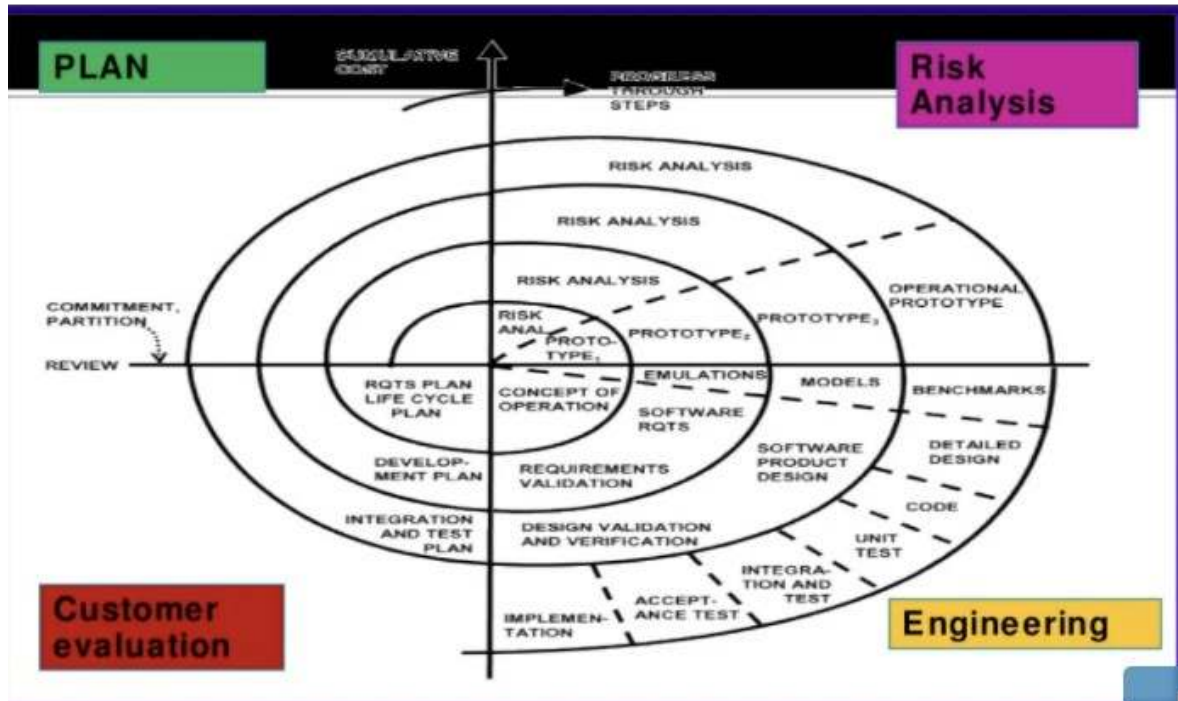
Phần sinh viên: Sau khi sinh viên, nhập học sẽ được cấp một tài khoản sinh viên có thể thay đổi mật khẩu. Sinh viên có thể dùng tài khoản này để tham gia học và sử dụng các dịch vụ của nhà trường, hơn hết là đóng tiền học

#### **1.2.3. Quản lý việc đóng tiền**

Mỗi sinh viên có một quá trình học khác nhau, dẫn đến việc đóng tiền học cũng khác nhau. Vì vậy cần phải quản lý quá trình sinh viên nộp tiền và gửi tin nhắn xác nhận

### 1.3. Giới thiệu về mô hình quy trình xoắn ốc và công nghệ áp dụng trong hệ thống thu tiền học của trường đại học công nghiệp Hà Nội

#### 1.3.1. Mô hình quy trình xoắn ốc



Hình ảnh 1.1: Hình ảnh mô hình thác nước

- Là mô hình kết hợp giữa các tính năng của mô hình prototyping và mô hình thác nước.
- Mô hình xoắn ốc được ưa chuộng cho các dự án lớn, đắt tiền và phức tạp.
- Mô hình này sử dụng những giai đoạn tương tự như mô hình thác nước, về thứ tự, plan, đánh giá rủi ro, ...

#### 1.3.2. Phân tích mô hình

Các pha trong quy trình phát triển xoắn ốc bao gồm:

- Objective identification- Thiết lập mục tiêu: xác định mục tiêu, đối tượng cho từng pha của dự án.

- Alternate evaluation- Đánh giá và giảm thiểu rủi ro: đánh giá rủi ro và thực hiện các hành động để giảm thiểu rủi ro.
- Product development- Phát triển sản phẩm: Lựa chọn mô hình phù hợp để phát triển hệ thống.
- Next phase planning- Lập kế hoạch: Thu thập, phân tích yêu cầu từ của dự án từ phía khách hàng. Bao gồm các công việc: Ước lượng chi phí (*estimating cost*), lên lịch trình thực hiện dự án (*shedule-master*), xác định số lượng nhân lực, môi trường làm việc (*identifying necessary resources and work environment*), tìm hiểu yêu cầu hệ thống (*requirements*) từ đó đưa ra các tài liệu đặc tả (*Bussiness Requirement Specifications* và *System Requirement specifications*) để phục vụ cho việc trao đổi giữa khách hàng và phân tích hệ thống sau này.

Ứng dụng: Mô hình này thường được sử dụng cho các ứng dụng lớn và các hệ thống được xây dựng theo các giai đoạn nhỏ hoặc theo các phân đoạn. Đối với các dự án có độ rủi ro từ trung bình đến cao. Những người sử dụng không chắc chắn về các nhu cầu của họ.

#### 1.3.2.1. Ưu điểm

- Lượng phân tích rủi ro cao. Do đó việc tránh rủi ro được tăng cường.
- Ước lượng chi phí dễ dàng như việc hoàn thành một prototype trong một fragment nhỏ.
- Ứng dụng tốt đối với các dự án lớn và quan trọng.
- Kiểm soát tài liệu và phê duyệt chặt chẽ.
- Chức năng bổ sung hoặc thay đổi có thể được thêm vào những giai đoạn sau.
- Phần mềm sẽ được sản xuất sớm trong vòng đời của phần mềm.
- Ứng dụng được phát triển nhanh và các tính năng được thêm vào một cách có hệ thống.

- Luôn có thời gian cho khách hàng để phản hồi về sản phẩm.

#### 1.3.2.2. Nhược điểm

- Đối với rủi ro, ở giai đoạn phân tích cần một chuyên gia có chuyên môn cao để thực hiện việc phân tích.
- Không hữu ích với dự án có quy mô nhỏ.
- Thời gian và chi phí cho dự án có thể là vô hạn vì đặc tính xoắn ốc của mô hình.
- Tài liệu cho dự án có thể rất dài vì có các giai đoạn trung gian.
- Rủi ro có thể không đáp ứng được tiến độ hoặc ngân sách.
- Sự thành công của dự án phụ thuộc rất nhiều vào giai đoạn phân tích rủi ro.

#### 1.3.3. Giới thiệu công cụ Rational Rose

Rational Rose là một bộ công cụ được sử dụng cho phát triển các hệ phần mềm hướng đối tượng theo ngôn ngữ mô hình hóa UML. Với chức năng của một bộ công cụ trực quan, Rational Rose cho phép chúng ta tạo, quan sát, sửa đổi, quản lý các biểu đồ. Tập kí hiệu mà Rational Rose cung cấp thống nhất với các kí hiệu trong UML. Ngoài ra, Rational Rose còn cung cấp chức năng hỗ trợ quản lý dự án phát triển phần mềm, cung cấp các thư viện hỗ trợ sinh khung mã cho hệ theo một ngôn ngữ lập trình nào đó.

Ngoài ra Rational Rose còn cung cấp:

- Use case: được sử dụng để miêu tả những yêu cầu mang tính chức năng của một hệ thống.
- Actor: giữ vai trò người dùng hệ thống,, phần cứng hoặc các hệ thống khác bên ngoài.

- Relationship: để kết nối các đối tượng với nhau tạo nên bản vẽ use case. Có các kiểu quan hệ cơ bản: Association, Generalization, Include, Extend.

#### **1.3.4 Phương pháp phân tích và thiết kế phần mềm**

Phương pháp thiết kế hướng đối tượng là sự kết hợp của phương pháp hướng dữ liệu và phương pháp hướng chức năng nó chú trọng đến cả thành phần dữ liệu và chức năng của hệ thống.

Theo phương pháp hướng đối tượng thì một hệ thống phần mềm là một tập hợp các đối tượng có khả năng tương tác với nhau. Các đối tượng chính là các sự vật và hiện tượng vật lý cũng như trừu tượng mà chúng ta có trong thế giới thực.

Mỗi đối tượng có dữ liệu riêng được che dấu với thế giới bên ngoài và các thao tác mà đối tượng có thể thực hiện trên các thành phần dữ liệu của đối tượng. Các đối tượng liên lạc, trao đổi thông tin với nhau bằng cách gửi các thông điệp cho nhau.

### **1.4. Kết luận chương 1**

Chương 1 đã giới thiệu chung về dự án phần mềm xoắn ốc và áp dụng trong phát triển phần mềm đóng tiền học phí của sinh viên trường đại học Công nghiệp Hà Nội. mô hình quy trình bản mẫu và giới thiệu về công cụ Rational Rose.

## **Chương 2: Tài liệu đặc tả phần mềm quản lý thu học phí Đại học Công Nghiệp Hà Nội**

### **2.1. Liệt kê các tác nhân và ca sử dụng**

#### **2.1.1. Tác nhân**

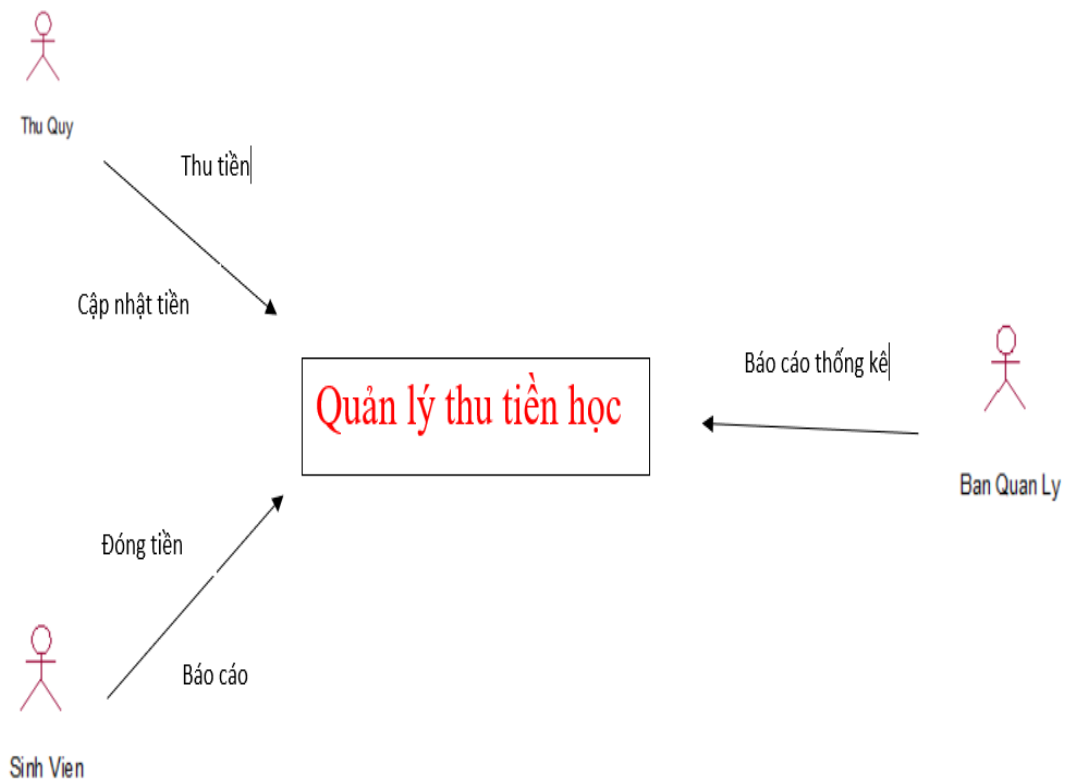
- Sinh viên: là sinh viên của trường
- Thủ quỹ: là người trực tiếp làm việc với sinh viên( thu tiền)
- Ban quản lí: là người điều hành, kiểm soát tiền, tổng hợp thống kê từng kì

#### **2.1.2. Use case**

- Tác nhân sinh viên :
  - Nạp tiền
  - Thanh toán
  - Kiểm tra tiền
  - Kiểm tra giao dịch
  - Trợ giúp
- Tác nhân thủ quỹ:
  - Hiện thị tín chỉ
  - Cập nhật tín chỉ
  - Cập nhật tiền
- Tác nhân ban quản lí:
  - Quản lí sinh viên
  - Quản lí thủ quỹ
  - In hóa đơn
  - Thông báo lịch sử giao dịch
  - Thống kê
  - Quản lí tiền
  - Quản lí tài khoản

## 2.2. Biểu đồ các UC và biểu đồ khung cảnh hệ thống

### 2.2.1. Biểu đồ khung cảnh hệ thống

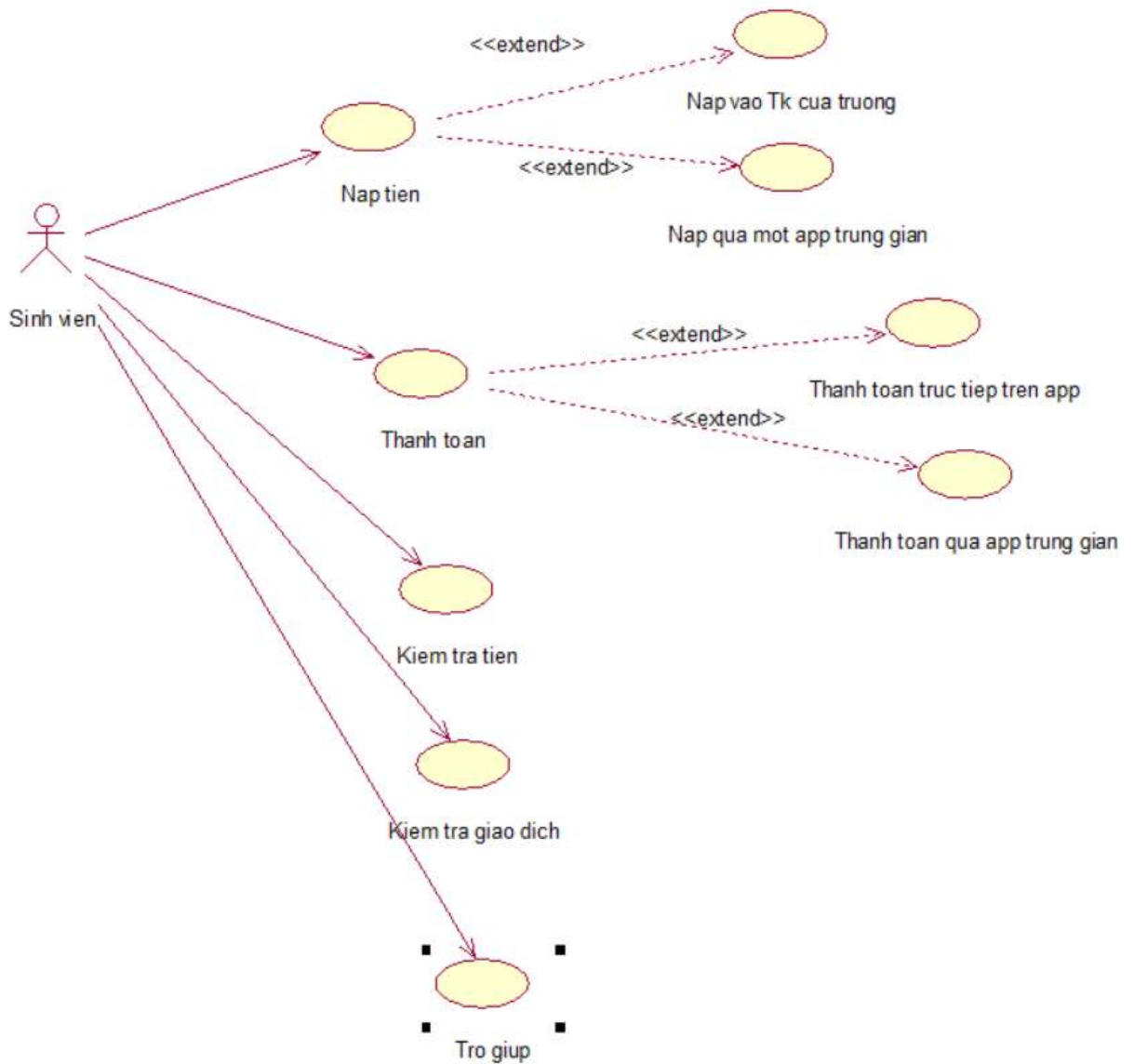


Hình 1.2 : Biểu đồ mức cảnh

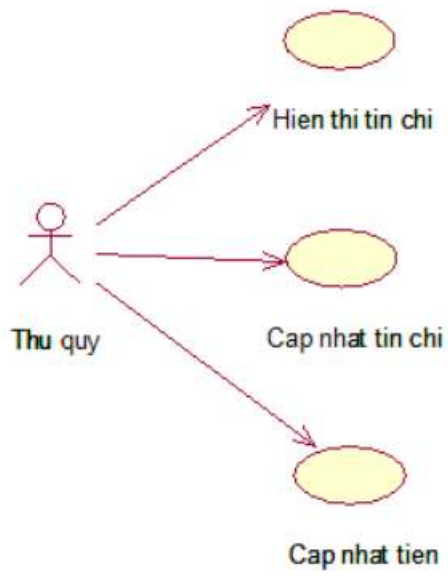


### 2.2.2. Biểu đồ use case chính

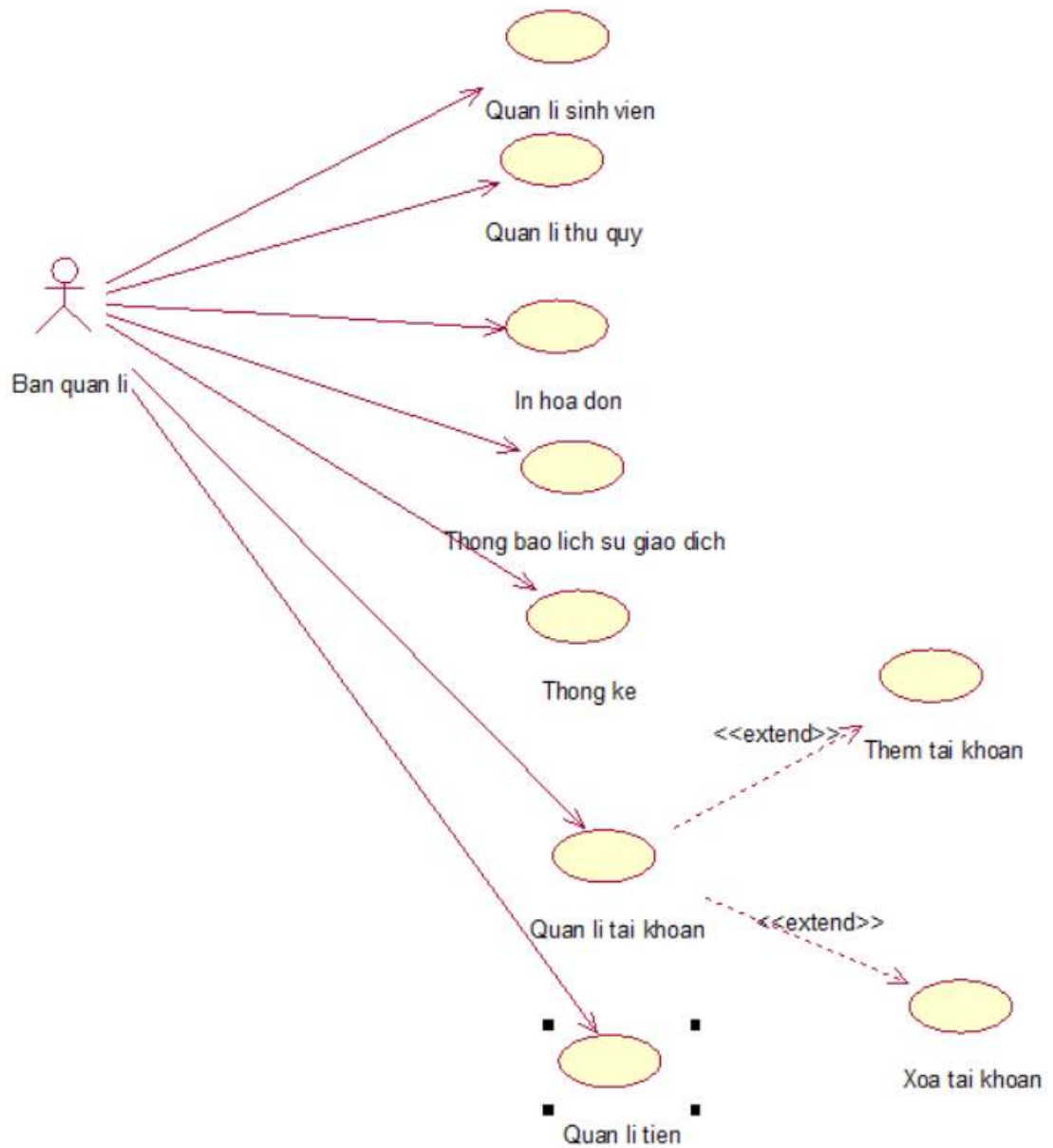
Sơ đồ use case tổng quan dưới đây mô tả sự tương tác giữa người dùng và hệ thống.



Hình 1.3: Biểu đồ use case chính sinh viên



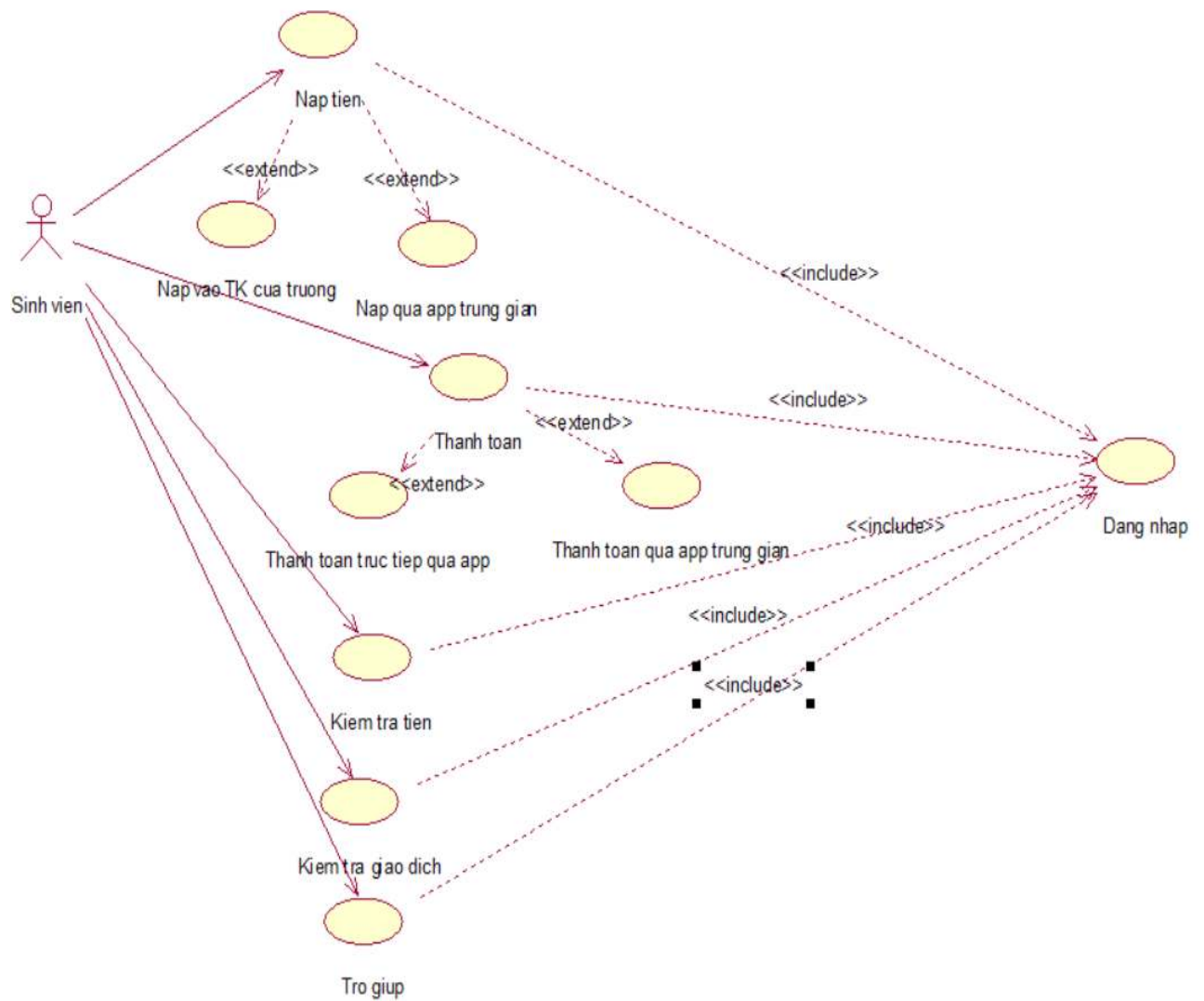
Hình 1.4: Hình ảnh use case chính thủ quỹ



Hình 1.5 Biểu đồ use case chính ban quản lý

### 2.2.3. Biểu đồ use case Sinh Viên

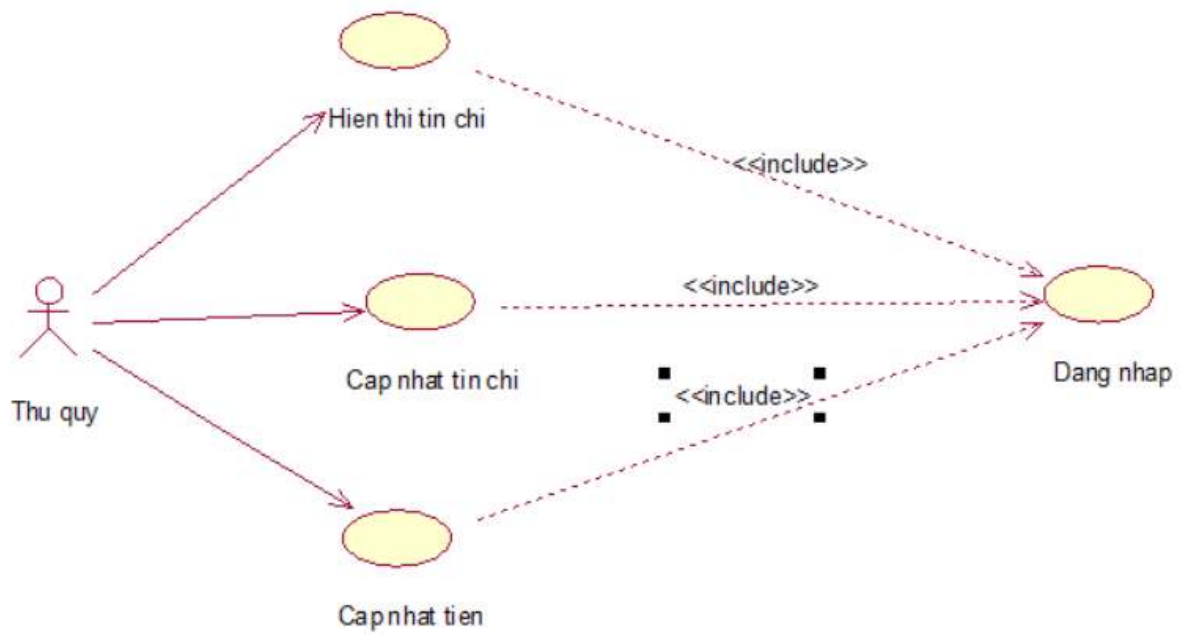
Sơ đồ use case sinh viên dưới đây để hình dung rõ ràng các cách mà một sinh viên có thể tương tác với phần mềm.



Hình 2.1: Biểu đồ UC Sinh Viên

#### 2.2.4. Biểu đồ use case thủ quỹ

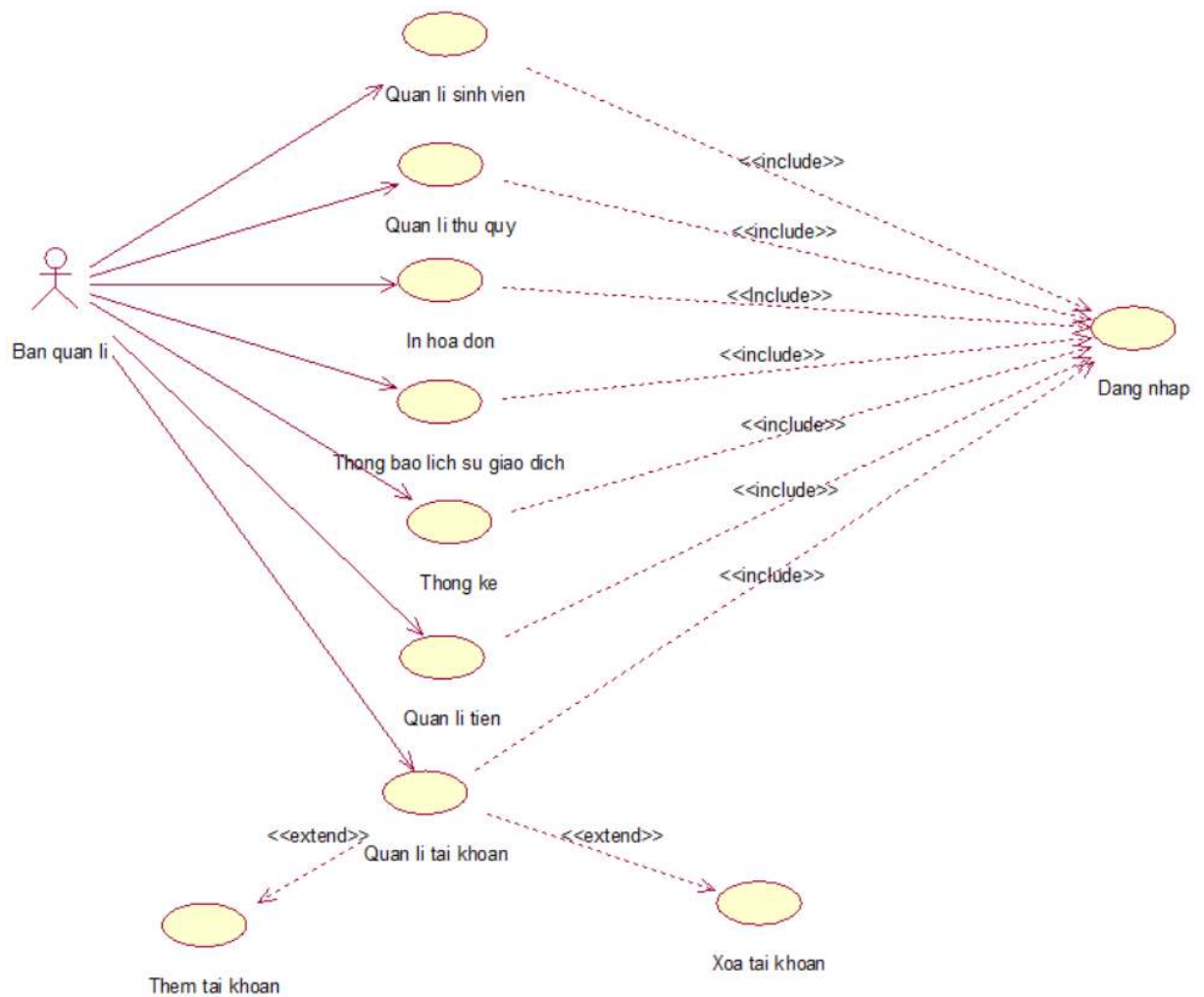
Sơ đồ use case Thủ quỹ dưới đây để hình dung rõ ràng các cách mà một thủ quỹ có thể tương tác với phần mềm.



Hình 2.2: Biểu đồ UC thủ quỹ

### 2.2.5. Biểu đồ use case ban quản lý

Sơ đồ use case Ban quản lý dưới đây để hình dung rõ ràng các cách mà một ban quản lý có thể tương tác với phần mềm.



Hình 2.3 Biểu đồ UC ban quản lý

## 2.3. Đặc tả ca sử dụng

### 2.3.1. Đăng nhập

1. **Tên use case:** Đăng nhập
2. **Tác nhân:** Sinh viên
3. **Mô tả tóm tắt:** Sinh viên muốn thực hiện thanh toán học phí thì phải đăng nhập vào hệ thống theo tài khoản và mật khẩu nhà trường đã cung cấp khi nhập học.

### 4. Luồng sự kiện

#### 4.1) Luồng cơ bản:

1. UC bắt đầu khi sinh viên đăng nhập theo đúng thông tin đã được cung cấp.

2. Hệ thống sẽ kiểm tra thông tin đăng nhập của sinh viên nếu đúng sẽ cho sinh viên truy cập vào hệ thống và bắt đầu thực hiện thanh toán các khoản học phí theo yêu cầu của sinh viên.

#### 4.2) **Luồng sự kiện:**

1. Tại bước 1 của luồng cơ bản nếu thông tin đăng nhập sai hệ thống từ chối cho sinh viên đăng nhập và yêu cầu sinh viên nhập lại, ca sử dụng kết thúc.
2. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ thông báo lỗi use case kết thúc.

**5) Các yêu cầu đặc biệt:** Không có.

**6) Tiền điều kiện:** Không có.

**7) Hậu điều kiện:** Nếu use case thành công sinh viên sẽ đăng nhập được vào hệ thống. Nếu không trạng thái của hệ thống không thay đổi.

**8) Các điểm mở rộng:** Không có.

### 2.3.2. **Nạp tiền**

- 1) **Tên use case:** Nạp tiền
- 2) **Tác nhân:** Sinh viên
- 3) **Mô tả tóm tắt:** UC bắt đầu khi sinh viên tiến hành nạp tiền vào tài khoản theo nguồn tiền quy định trong hệ thống.

#### 4) **Luồng sự kiện**

##### 4.1) **Luồng cơ bản:**

1. UC bắt đầu khi sinh viên chọn nạp tiền và chọn nguồn tiền từ thẻ ngân hàng hoặc từ ứng dụng bên thứ ba được liên kết với hệ thống.
2. Hệ thống sẽ kiểm tra số tiền nạp vào tài khoản nếu số tiền trong tài khoản trùng khớp hoặc lớn hơn so với số tiền sinh viên yêu

cầu nạp vào hệ thống tiến hành nạp tiền vào thông báo lại với sinh viên.

#### 4.2) **Luồng rẽ nhánh:**

1. Tại bước 1 của luồng cơ bản nếu số tiền trong tài khoản ngân hàng hoặc bên thứ ba ít hơn số tiền sinh viên nạp vào hệ thống thì yêu cầu sẽ không được chấp nhận và ca sử dụng kết thúc.
2. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ thông báo lỗi use case kết thúc.

5) **Các yêu cầu đặc biệt:** Không có.

6) **Tiền điều kiện:** Không có.

7) **Hậu điều kiện:** Nếu use case thành công sinh viên sẽ nạp tiền được vào hệ thống và số dư trong tài khoản sẽ thay đổi. Nếu không trạng thái của hệ thống không thay đổi.

8) **Các điểm mở rộng:** Không có.

### 2.3.3. Thanh toán

1) **Tên use case:** Thanh toán.

2) **Tác nhân:** Sinh viên

3) **Tóm tắt:** UC bắt đầu khi sinh viên tiến hành thanh toán số tiền học phí.

#### 4) **Luồng sự kiện**

##### 4.1) **Luồng cơ bản:**

1. Khi tiến hành thanh toán hệ thống sẽ liệt kê ra các khoản học phí của sinh viên, sinh viên có thể lựa chọn thanh toán toàn bộ số tiền học phí hoặc thanh toán từng môn học tùy theo số tiền còn dư trên hệ thống của sinh viên.
2. Sau khi chọn sinh viên chọn thanh toán hệ thống sẽ trừ tiền trong tài khoản của sinh viên và thông báo cho sinh viên biết những môn học đã thanh toán thành công.

##### 4.2) **Luồng sự kiện:**



1. Tại bước 2 của luồng cơ bản nếu số tiền còn dư trong tài khoản sinh viên không đủ hệ thống sẽ không thanh toán và đưa ra thông báo cho sinh viên biết số dư không đủ cần nạp thêm tiền, ca sử dụng kết thúc.
  2. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ thông báo lỗi use case kết thúc.
- 5) **Các yêu cầu đặc biệt:** Không có.
  - 6) **Tiền điều kiện:** Không có.
  - 7) **Hậu điều kiện:** Nếu use case thành công sinh viên sẽ thanh toán được các khoản học phí đã chọn và số dư trong tài khoản thay đổi. Nếu không trạng thái của hệ thống không thay đổi.
  - 8) **Các điểm mở rộng:** Không có.

#### 2.3.4. Kiểm Tra Tiền

1. **Tên use case:** Kiểm Tra Tiền
2. **Tác nhân:** Sinh Viên
3. **Mô tả tóm tắt:** Sinh viên kiểm tra học phí thì phải đăng nhập vào hệ thống theo tài khoản và mật khẩu nhà trường đã cung cấp để kiểm tra học phí của mình đã đóng và được nạp vào chưa
4. **Luồng sự kiện**

##### 4.1.Luồng cơ bản:

a.UC bắt đầu khi sinh viên đăng nhập theo đúng thông tin đã được cung cấp.

b.Hệ thống sẽ kiểm tra thông tin đăng nhập của sinh viên nếu đúng sẽ cho sinh viên truy cập vào hệ thống và bắt đầu thực hiện kiểm tra học phí xem sinh viên đó đã đóng đủ học phí theo quy định của nhà trường chưa

##### 4.2.Luồng sự kiện:

a. Tại bước 1 của luồng cơ bản nếu thông tin kiểm tra học phí cho sinh viên, nếu học phí của sinh viên chưa đủ sinh viên đó phải nạp thêm tiền vào tài khoản theo đúng quy định của nhà trường

b. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ thông báo lỗi use case kết thúc.

**5. Các yêu cầu đặc biệt:** Không có.

**6. Tiền điều kiện:** Không có.

**7. Hậu điều kiện:** : Nếu use case thành công sinh viên sẽ kiểm tra được học phí của mình đã được đóng đủ hay chưa. Nếu không trạng thái của hệ thống không thay đổi.

**8. Các điểm mở rộng:** Không có.

### 2.3.5 Kiểm tra lịch sử giao dịch

1) **Tên use case:** Kiểm tra lịch sử giao dịch

2) **Tác nhân:** Sinh viên

3) **Mô tả tóm tắt:** UC bắt đầu khi sinh viên tiến hành kiểm tra lịch sử giao dịch trong hệ thống khi sinh viên đó đã nạp tiền vào hệ thống.

4) **Luồng sự kiện**

**4.1) Luồng cơ bản:**

a) UC bắt đầu khi sinh viên kích vào phần lịch sử giao dịch để check xem số tiền trong tài khoản trên hệ thống đã được trừ hay chưa.

b) Hệ thống sẽ kiểm tra lại số tiền trong tài khoản mà sinh viên đó đã đóng, rồi sẽ in lịch sử giao dịch đó về các khoản tiền mà sinh viên đã đóng để sinh viên đó biết được lịch sử giao dịch đóng tiền của mình

**4.2) Luồng sự kiện**

- a) Tại bước 1 của luồng cơ bản nếu sinh viên đó đã đóng đầy đủ tiền của kì học đó nhưng lịch sử giao dịch vẫn còn trừ thiếu thì sinh viên đó phải gọi cho giáo viên chủ nhiệm để xem lại .
- b) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ thông báo lỗi use case kết thúc.

**5) Các yêu cầu đặc biệt:** Không có.

**6) Tiền điều kiện:** Không có.

**7) Hậu điều kiện:** Nếu use case thành công sinh viên sẽ xem được lịch sử giao dịch của mình đã thành công. Nếu không trạng thái của hệ thống không thay đổi.

**8) Các điểm mở rộng:** Không có

### 2.3.6. Trợ giúp

**1) Tên use case:** Trợ giúp

**2) Tác nhân:** Sinh viên

**3) Tóm tắt:** UC bắt đầu khi sinh viên tiến hành click vào phần trợ giúp trên web để được sự hỗ trợ của admin

**4) Luồng sự kiện**

**4.1) Luồng cơ bản:**

- a) Khi tiến hành chọn từ “Trợ giúp” trên web trường thì ở đó sẽ có nhiều mục để sinh viên chọn và được tư vấn về đội hỗ trợ của nhà trường
- b) Sau khi sinh viên chọn trợ giúp thì hệ thống sẽ gửi thông báo về đội hỗ trợ để đội hỗ trợ có thể giúp sinh viên giải quyết được vấn đề

**4.2) Luồng sự kiện:**

- a) Tại bước 2 của luồng cơ bản nếu sự trợ giúp của đội hỗ trợ chưa được phản hồi thì sinh viên có thể gọi trực tiếp lên trên nhà trường để được giải quyết những vấn đề còn khúc mắc
- b) Ti bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ thông báo lỗi use case kết thúc.

**5. Các yêu cầu đặc biệt:** Không có.

**6. Tiền điều kiện:** Không có.

**7. Hậu điều kiện:** Nếu use case thành công sinh viên sẽ được hỗ trợ giải quyết những vấn đề một cách nhanh chóng từ đội hỗ trợ. Nếu không trạng thái của hệ thống không thay đổi.

**8. Các điểm mở rộng:** Không có.

### **2.3.7. Hiện thị tín chỉ**

1) Tên use case: Hiện thị tín chỉ

2) Tác nhân: Thủ quỹ

3) Tóm tắt: UC bắt đầu thủ quỹ hiện thị số tín chỉ phải đăng kí để cho sinh viên biết

4) Luồng sự kiện

4.1) Luồng cơ bản:

- a) Sau khi sinh viên đăng kí số tín chỉ thì thủ quỹ sẽ hiện thị số tín chỉ sinh viên phải đăng kí cho họ biết trong kì này
- b) Số tín chỉ này sẽ được hiện lên trên trang web hoặc app của nhà trường

4.2) Luồng sự kiện:

Sinh Viên kiểm tra mà số tín chỉ trong kì ko đạt điều kiện thì sẽ thông báo cho nhà trường để thủ quỹ kiểm tra lại

5) Các yêu cầu đặc biệt: Không có.

6) Tiền điều kiện: Không có.

7) Hậu điều kiện: Không có

8) Các điểm mở rộng: Không có.

### **2.3.8. Cập nhật tín chỉ**

1) Tên use case: Cập nhật tín chỉ

2) Tác nhân: Thủ quỹ

3) Tóm tắt: UC bắt đầu khi sinh viên đã đăng kí tín chỉ và thủ quỹ cập nhật số tín chỉ ấy

4) Luồng sự kiện

4.1) Luồng cơ bản:

a) Sau khi sinh viên đăng kí tín chỉ thì thủ quỹ sẽ cập nhật lại số tín chỉ đã đăng kí và hiển thị số tín chỉ chưa đăng kí của sinh viên

b) Số tín chỉ này sẽ được hiện lên trên trang web hoặc app của nhà trường

4.2) Luồng sự kiện:

a) Sinh viên kiểm tra số tín chỉ mình đã đăng kí khác so với số tín chỉ của nhà trường hiển thị, cần thông báo lại ngay để sửa đổi

b) Nếu số tiền bị trừ không phù hợp với số tín chỉ, cần báo với nhà trường để chỉnh sửa

5) Các yêu cầu đặc biệt: Không có.

6) Tiền điều kiện: Không có.

7) Hậu điều kiện: Không có

8) Các điểm mở rộng: Không có.

### **2.3.9. Cập nhật tiền**

1) Tên use case: Cập nhật tiền

2) Tác nhân: Thủ quỹ

3) Tóm tắt: UC bắt đầu khi thủ quỹ cập nhật lại số tiền

4) Luồng sự kiện

4.1) Luồng cơ bản:

Sau khi sinh viên đóng tiền số tín chỉ thì thủ quỹ sẽ cập nhật lại số tiền mà sinh viên còn lại trong tài khoản cũng như công nợ

4.2) Luồng sự kiện:

- a) Khi số tiền trong tài khoản của sinh viên không đủ để thanh toán, sẽ đưa ra thông báo cho sinh viên
- b) Khi có yêu cầu xem xét lại số tiền đã đóng, thủ quỹ cần thực hiện việc cập nhật tiền thêm 1 lần nữa

5) Các yêu cầu đặc biệt: Không có.

6) Tiền điều kiện: Không có.

7) Hậu điều kiện: Không có

8) Các điểm mở rộng: Không có.

### **2.3.10. In hóa đơn**

1) Tên use case: In hóa đơn

2) Tác nhân: Nhân viên

3) Mô tả tóm tắt: Use case trên cho nhân viên thông báo cho sinh viên thanh toán trực tiếp biết số tiền đã thanh toán

4) Luồng sự kiện

4.1) Luồng cơ bản:

- a) UC bắt đầu khi nhân viên nhấn vào nút in hóa đơn
- b) Hệ thống sẽ kiểm tra số tiền mà sinh viên đã thanh toán sau đó sẽ in ra thành hóa đơn để thông báo cho sinh viên

4.2) Luồng rẽ nhánh:

- a) Tại bước 2 ,hệ thống thông báo chưa thanh toán và từ chối in hóa đơn use case kết thúc
- b) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ thông báo lỗi use case kết thúc.

**5) Các yêu cầu đặc biệt:** Không có.

**6) Tiền điều kiện:** Sinh viên đã thanh toán.

**7) Hậu điều kiện:** Không có

**8) Các điểm mở rộng:** Không có.

### **2.3.11. Thông báo lịch sử giao dịch**

**1) Tên use case:** Thông báo lịch sử giao dịch

**2) Tác nhân:** Nhân viên thu tiền

**3) Mô tả tóm tắt:** Use case giúp nhân viên thông báo cho sinh viên thanh toán tiền trực tuyến biết được số tiền đã thanh toán

**4) Luồng sự kiện**

**4.3) Luồng cơ bản:**

1. UC bắt đầu khi nhân viên nhấn nút thu tiền
2. Hệ thống sẽ kiểm tra số tiền mà sinh viên đã thanh toán sau đó hệ thống sẽ gửi thông báo tới tài khoản vừa thanh toán số tiền vừa chuyển.

**4.4) Luồng sự kiện:**

1. Tại bước 2, hệ thống không thể kiểm tra số tiền vừa thanh toán và thông báo tài khoản chưa thanh toán thì nhân viên nhấn cập nhật, use case sử dụng kết thúc.

**5) Các yêu cầu đặc biệt:** Không có.

**6) Tiền điều kiện:** Sinh viên đã thanh toán.

**7) Hậu điều kiện:** Không có

**8) Các điểm mở rộng:** Không có.

### 2.3.12. Thống kê

1) **Tên use case:** Thống kê

2) **Tác nhân:** Nhân viên

3) **Mô tả tóm tắt:** Use case sử dụng cho phép nhân viên lấy thông tin, báo cáo nhỏ từ hệ thống, tổng hợp thành báo cáo hàng tháng hoặc báo cáo khi có yêu cầu về doanh thu

4) **Luồng sự kiện**

4.1) **Luồng cơ bản:**

- a) Nhân viên đăng nhập vào hệ thống bằng tài khoản của mình, lựa chọn chức năng cần thống kê, lấy bản thống kê từ hệ thống.
- b) Lựa chọn cách tổng hợp : có thể theo tháng, tuần, hoặc theo thời gian...sau đó chọn “ lưu “ và có thể in báo cáo

4.2) **Luồng sự kiện:**

Tại bước 1 của luồng dữ liệu hệ thống không thể in ra bảng thống kê, use case kết thúc.

5) **Các yêu cầu đặc biệt:** Không có.

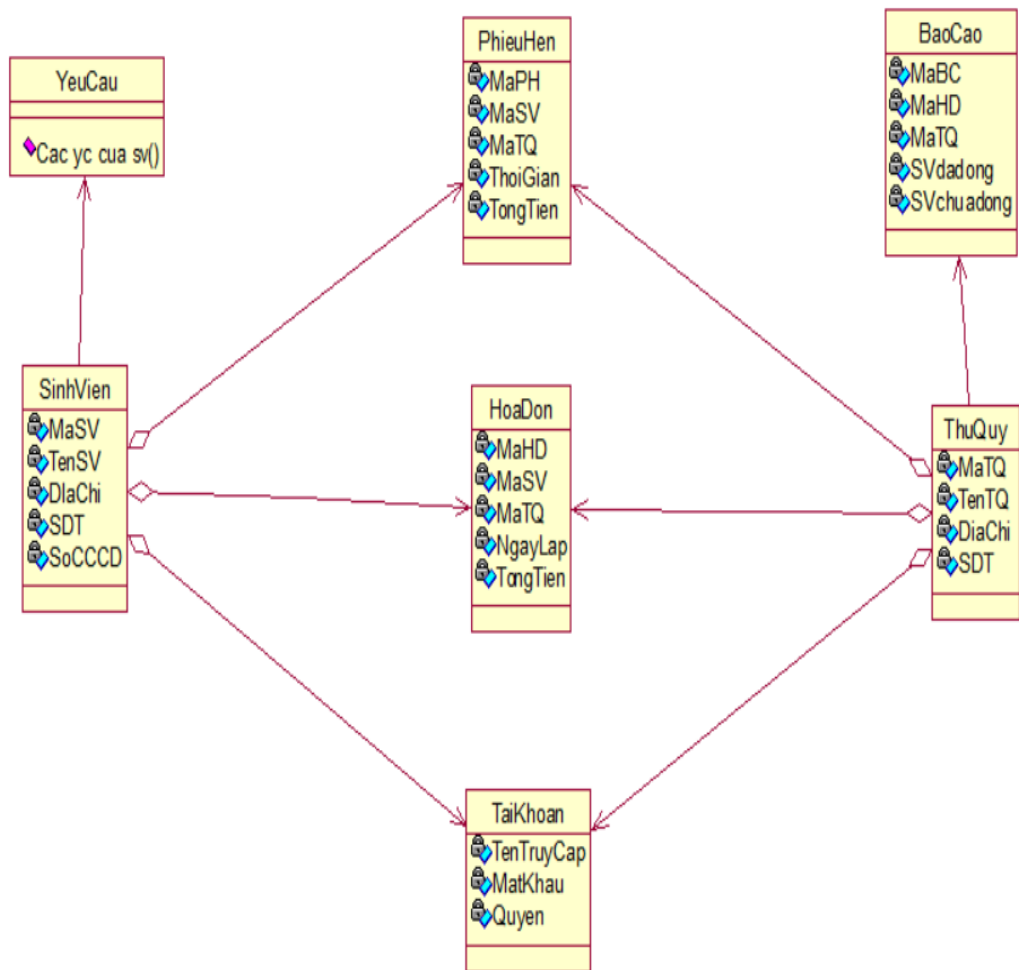
6) **Tiền điều kiện:** Không có.

7) **Hậu điều kiện:** Không có

8) **Các điểm mở rộng:** Không có.



## 2.4. Biểu đồ lĩnh vực lớp



Hình 3.1: Biểu đồ lĩnh vực lớp

### Chương 3: Đặc tả và thiết kế phần mềm

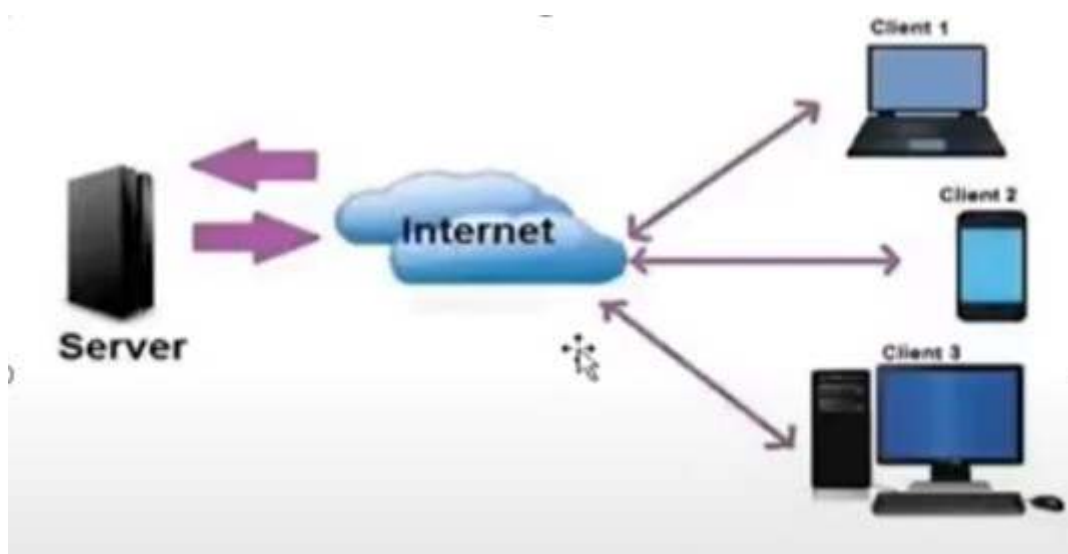
Nội dung Chương 3 trình bày về thiết kế cấu trúc; thiết kế kiến trúc (các thành phần phân hệ); thiết kế giao diện; thiết kế CSDL.

#### 3.1. Kiến trúc phần mềm

Sử dụng mô hình client server. (Tại sao bạn lại dùng kiến trúc Client/Server?)

Mô hình mạng client server đây là mô hình mạng máy tính trong đó các máy tính client được đóng vai trò như một máy khách (client), chúng sẽ gửi yêu cầu (request), đến các máy chủ (server). Để máy chủ xử lý những yêu cầu đó và trả kết quả về cho máy khách (client).

Ở đây server sẽ là máy chủ được đặt tại cửa hàng, còn client sẽ là các thiết bị mà khách hàng dùng để truy cập website của cửa hàng



Hình 3.2. Mô hình client server

#### 3.2. Thiết kế cấu trúc

Dựa trên đặc tả các chức năng chương 2 (trình tự các chức năng) Trình tự (hàm, thủ tục, phương thức, lớp thực thể), hoạt động, lớp tham gia ca sử dụng.

Chúng ta xác định được các hàm/thủ tục/lớp thực thể sau:

-

-

-

Từ lớp thực thể: sinh ra cơ sở dữ liệu

### **3.3. Thiết kế giao diện**

### **3.4. Thiết kế cơ sở dữ liệu**

## KẾT LUẬN

Sau 13 tuần nghiên cứu thiết kế và phát triển, được sự hướng dẫn của giảng viên, nhóm chúng em đã hoàn thiện 1 phần mềm thu tiền học của trường đại học công nghiệp Hà Nội đáp ứng được các tính năng nghiệp vụ chính, bao gồm đăng kí, đăng nhập,...ngoài ra nhóm bổ sung thêm 1 số tính năng như quản lí sinh viên, quản lý thủ quỹ ,... Nhóm đã tuân thủ theo mô hình xoắn ốc với các sơ đồ và hình vẽ như đã được trình bày trong phần nội dung báo cáo. Thông qua học và làm đề tài, chúng em có thể có thêm hành trang về việc phân tích một đối tượng, không chỉ riêng trong sách vở mà còn có áp dụng vào thực tế đời sống khi phân tích một vấn đề cụ thể. Thời gian tìm hiểu và thực hiện đề tài cũng là dịp để chúng em hiểu biết thêm về cách thức, bước đi để thiết kế, xây dựng một hệ thống Website là như thế nào. Từ đó chúng em có thêm nhiều kiến thức quý báu để tự tin bước vào đội ngũ những người làm công nghệ thông tin góp phần phát triển đất nước trong 1 tương lai gần.