**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**--------------------------------------------------**

**BÁO CÁO ĐỒ ÁN MÔN HỌC**

**PHÁT TRIỂN PHẦN MỀM HƯỚNG DỊCH VỤ**

**Đề tài: Ứng dụng nhập học cho tân sinh viên PTIT HCM**

**GIẢNG VIÊN HƯỚNG DẪN: ThS. Huỳnh Trung Trụ**

**Sinh viên thực hiện: - Đinh Văn Ty N20DCCN070  
 - Nguyễn Quốc Duy N20DCCN013**

**Lớp: D20CQCNHT01-N**

**Khóa: 2020-2025**

**Hệ: Đại học chính quy**

**TP.THỦ ĐỨC, tháng 01 năm 2024**

**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**--------------------------------------------------**

**BÁO CÁO ĐỒ ÁN MÔN HỌC**

**PHÁT TRIỂN PHẦN MỀM HƯỚNG DỊCH VỤ**

**Đề tài: Ứng dụng nhập học cho tân sinh viên PTIT HCM**

**GIẢNG VIÊN HƯỚNG DẪN: ThS. Huỳnh Trung Trụ**

**Sinh viên thực hiện: - Đinh Văn Ty - N20DCCN070  
 - Nguyễn Quốc Duy - N20DCCN013**

**Lớp: D20CQCNHT01-N**

**Khóa: 2020-2025**

**Hệ: Đại học chính quy**

**TP.THỦ ĐỨC, tháng 01 năm 2024**

Mục lục

[PHẦN 1: ĐẶC TẢ YÊU CẦU VÀ QUẢN LÝ HỆ THỐNG 1](#_Toc155111086)

[1. Tổng quan 1](#_Toc155111087)

[1.1. Giới thiệu 1](#_Toc155111088)

[1.2. Quy mô hoạt động 1](#_Toc155111089)

[1.3. Quá trình nhập học trên thực tế 1](#_Toc155111090)

[1.4. Phân tích 1](#_Toc155111091)

[1.5. Mô tả sơ bộ 2](#_Toc155111092)

[1.6. Đối tượng sử dụng 2](#_Toc155111093)

[PHẦN 2: THIẾT KẾ HỆ THỐNG 2](#_Toc155111094)

[1. Thiết kế hệ thống 2](#_Toc155111095)

[2.Cơ sở dữ liệu 3](#_Toc155111096)

[3.Mô hình Microsevice và Restful API 6](#_Toc155111097)

[PHẦN 3: THIẾT KẾ GIAO DIỆN 9](#_Toc155111098)

[1. Giao diện trên ứng dụng desktop 10](#_Toc155111099)

[2. Giao diện trên website 14](#_Toc155111107)

[Phần 4: Tài liệu hướng dẫn sử dụng API 18](#_Toc155111115)

[1.Giới thiệu: 18](#_Toc155111116)

[2.SinhVien 18](#_Toc155111117)

[3.Học Phí 22](#_Toc155111118)

[4.VnPay (Thanh toán học phí) 23](#_Toc155111119)

[5.Đồng phục 24](#_Toc155111120)

[6.Ký túc xá 25](#_Toc155111121)

[7.Login 26](#_Toc155111122)

[Phụ lục 27](#_Toc155111123)

[1. Danh mục các hồ sơ cần cho quá trình nhập học 27](#_Toc155111124)

PHẦN 1: ĐẶC TẢ YÊU CẦU VÀ QUẢN LÝ HỆ THỐNG

1. Tổng quan

1.1. Giới thiệu

Nhập học là quy trình bắt buộc để sinh viên xác nhận việc theo học tại một cơ sở giáo dục nào đó trên cả nước. Đây là thời điểm mà những học viên được làm quen, giới thiệu về môi trường học tập cũng như cơ sở vật chất của trường.

Tuy nhiên việc này trên thực tế thường mất thời gian hơn dự kiến do toàn bộ quá trình thường được yêu cầu phải hoàn thành trong một hoặc vài ngày, kết hợp thêm tâm lý muốn hoàn thành sớm thủ tục của các bạn học viên mà lượng người dồn về tại cùng một thời điểm là rất lớn, dẫn đến tình trạng quá tải.

Nắm bắt được thực trạng đó, nhóm quyết định xây dựng hệ thống ***“Nhập học tân sinh viên”*** để hiểu rõ hơn về các nghiệp vụ thực tế cũng như đóng góp được một giải pháp giúp phần nào đó cải thiện được vấn đề nêu trên.

1.2. Quy mô hoạt động

Phần mềm được phát triển sử dụng trong phạm vi Học viện Công nghệ Bưu chính Viễn thông theo **từng năm học**.

Hệ thống được xây dựng trên hai nền tảng **website** và **ứng dụng desktop**.

1.3. Quá trình nhập học trên thực tế

Sinh viên được phát phiếu hồ sơ, túi đựng hồ sơ, phiếu đăng ký thẻ sinh viên liên kết ngân hàng. Dựa trên số thứ tự có sẵn, sinh viên chờ đến lượt và nộp các loại hồ sơ theo yêu cầu tại từng bàn khác nhau theo hướng dẫn. Ngoài ra, sinh viên có thể tiến hành nộp các khoản thu liên quan đến học phí và ký túc xá nếu có nhu cầu. Sau khi sinh viên nhận được thẻ sinh viên tạm thời của mình, quá trình nhập học được xem là hoàn tất.

1.4. Phân tích

Từ quá trình nhập học thực tế có thể nhận thấy những thao tác có liên quan đến hồ sơ, giấy tờ hay thủ tục đăng ký ký túc xá, đồng phục cũng như hoàn tất các khoản thu liên quan đều có thể được thực hiện trực tuyến thông qua hệ thống.

Như vậy có thể hình dung sơ bộ về các chứng năng chính sẽ có trong hệ thống:

1. Các thao tác liên quan đến thông tin cá nhân của sinh viên,

2. Tiếp nhận các loại hồ sơ sinh theo quy định (xem *phụ lục*),

3. Hỗ trợ đăng ký thẻ liên kết sinh viên,

4. Đăng ký ký túc xá và đồng phục,

5. Thanh toán học phí và các chi phí phát sinh.

1.5. Mô tả sơ bộ

Sinh viên tiến hành đăng nhập vào hệ thống với tên đăng nhập và mật khẩu mặc định (số CCCD và ngày tháng năm sinh dạng ddmmyyyy).

Tại giao diện chính, sinh viên sẽ dựa theo các đề mục để thực hiện các thao tác cần thiết. Khi hoàn thành xong từng mục sẽ được trả về giao diện này để tiếp tục thực hiện thao tác.

Sinh viên được cập nhật các thông tin cá nhân và thông tin của người thân nhiều lần trong thời gian hệ thống hoạt động.

Sinh viên cần chuẩn bị các loại hồ sơ cần thiết và tải lên cloud (Google Drive, OneDrive, Dropbox, Mega,… ) trước khi nộp về học viện thông qua hệ thống.

Sinh viên dựa trên nhu cầu của bản thân để đăng ký dịch vụ liên quan đến thẻ liên kết ngân hàng của mình.

Việc đăng ký mua đồng phục là bắt buộc trước khi tiến hành thanh toán, sinh viên nếu có nguyện vọng ở tại ký túc xá của học viện cũng có thể đăng ký trên form.

Sau khi hoàn thành bước trên sẽ truy cập được vào giao diện thanh toán, sinh viên có hai tùy chọn “trực tuyến” hoặc “trực tiếp”. Nếu lựa chọn hình thức ban đầu “trực tiếp” thì sinh viên vẫn có thể chuyển sang “trực tuyến” để hoàn thành các khoản thu trong thời gian hệ thống hoạt động.

*Lưu ý: khi thanh toán hoàn tất sẽ* ***không thể*** *truy cập vào hai chức năng “đăng ký ký túc xá, đồng phục” và “thanh toán” được nữa do hóa đơn đã được lưu vào cơ sở dữ liệu.*

1.6. Đối tượng sử dụng

Hiện tại chỉ phục vụ một đối tượng duy nhất đó là **sinh viên**.

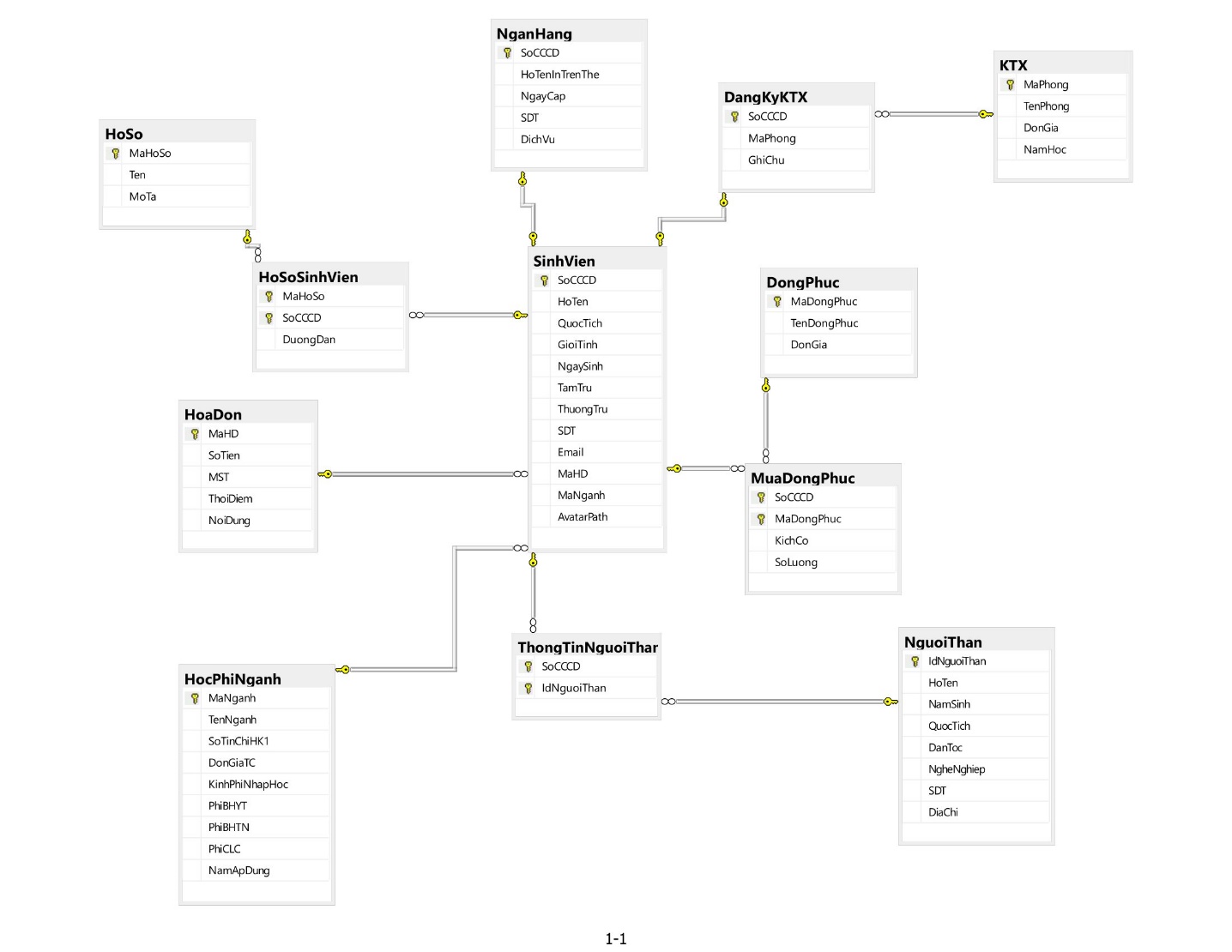
PHẦN 2: THIẾT KẾ HỆ THỐNG

1. Thiết kế hệ thống

Kiến trúc hệ thống: microservice

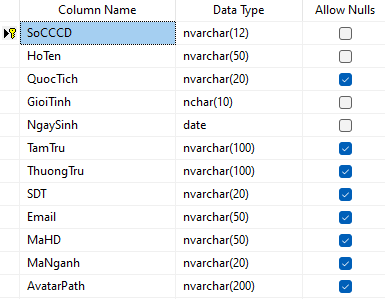
Cơ sở dữ liệu: hệ quản trị cơ sở dữ liệu Microsoft SQL Server

1. Cơ sở dữ liệu

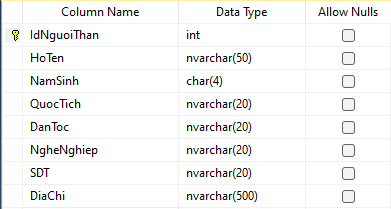
****Hình 2.1 Sơ đồ Diagram

Chi tiết bảng:

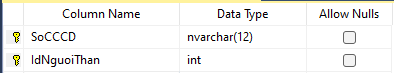
* Bảng SinhVien: lưu trữ thông tin cơ bản của sinh viên



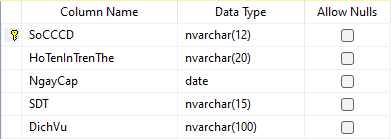
* Bảng NguoiThan: lưu trữ thông tin của người thân



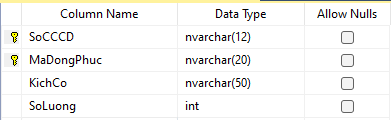
* Bảng ThongTinNguoiThan: lưu liên kết giữa người thân và sinh viên



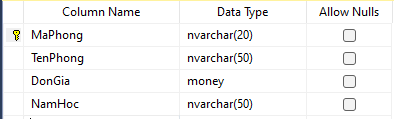
* Bảng NganHang: lưu trữ thông tin đăng ký ngân hàng của sinh viên



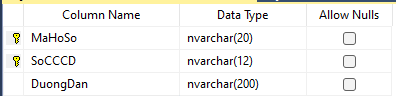
* Bảng MuaDongPhuc: lưu việc đăng ký đồng phục của sinh viên



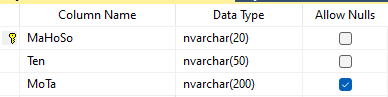
* Bảng KTX: lưu danh sách ký túc xá hiện có



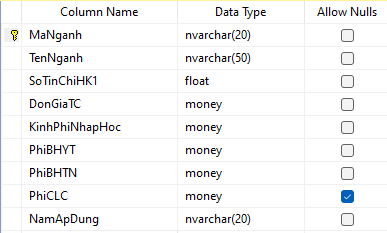
* Bảng HoSoSinhVien: lưu trữ đường dẫn của các loại hồ sơ



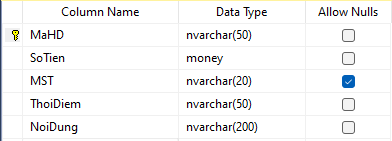
* Bảng HoSo: lưu danh sách hồ sơ



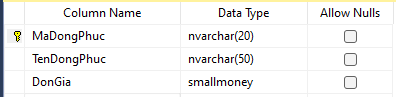
* Bảng HocPhiNganh: chứa thông tin về chi tiết học phí theo từng ngành đào tạo



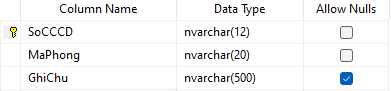
* Bảng HoaDon: lưu thông tin hóa đơn sau khi thanh toán



* Bảng DongPhuc: lưu danh sách đồng phục học viện cung cấp

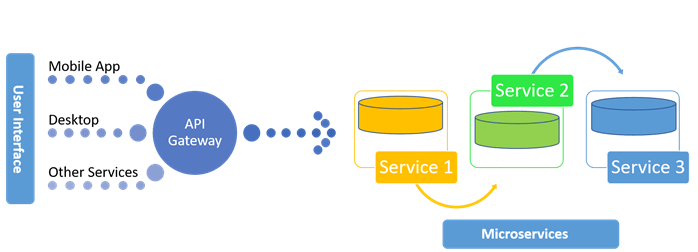


* Bảng DangKyKTX: lưu thông tin đăng ký ký túc xá của sinh viên



1. Mô hình Microsevice và Restful API
   1. **Microservice**

Microservices là một kiến trúc phần mềm trong đó một ứng dụng được chia thành nhiều dịch vụ nhỏ, độc lập, có thể tương tác với nhau thông qua các API. Mỗi dịch vụ được thiết kế để thực hiện một chức năng cụ thể, chẳng hạn như quản lý người dùng, xử lý thanh toán, hoặc cung cấp nội dung.



Hình 3.1 Mô hình microservice

Kiến trúc microservice có một số ưu điểm so với kiến trúc nguyên khối truyền thống, bao gồm:

* Tính linh hoạt và khả năng mở rộng: Các dịch vụ microservice có thể được phát triển, triển khai và quản lý độc lập với nhau. Điều này giúp cho việc phát triển và triển khai ứng dụng trở nên linh hoạt và dễ dàng hơn. Ngoài ra, các dịch vụ microservice có thể được mở rộng theo nhu cầu, giúp ứng dụng có thể đáp ứng được lượng truy cập ngày càng tăng.
* Tính chịu lỗi: Các dịch vụ microservice có thể hoạt động độc lập với nhau. Điều này giúp cho ứng dụng vẫn có thể hoạt động bình thường ngay cả khi một số dịch vụ bị lỗi.
* Tính bảo mật: Các dịch vụ microservice có thể được bảo vệ riêng biệt với nhau. Điều này giúp cho việc bảo mật ứng dụng trở nên dễ dàng hơn.

Tuy nhiên, kiến trúc microservice cũng có một số nhược điểm, bao gồm:

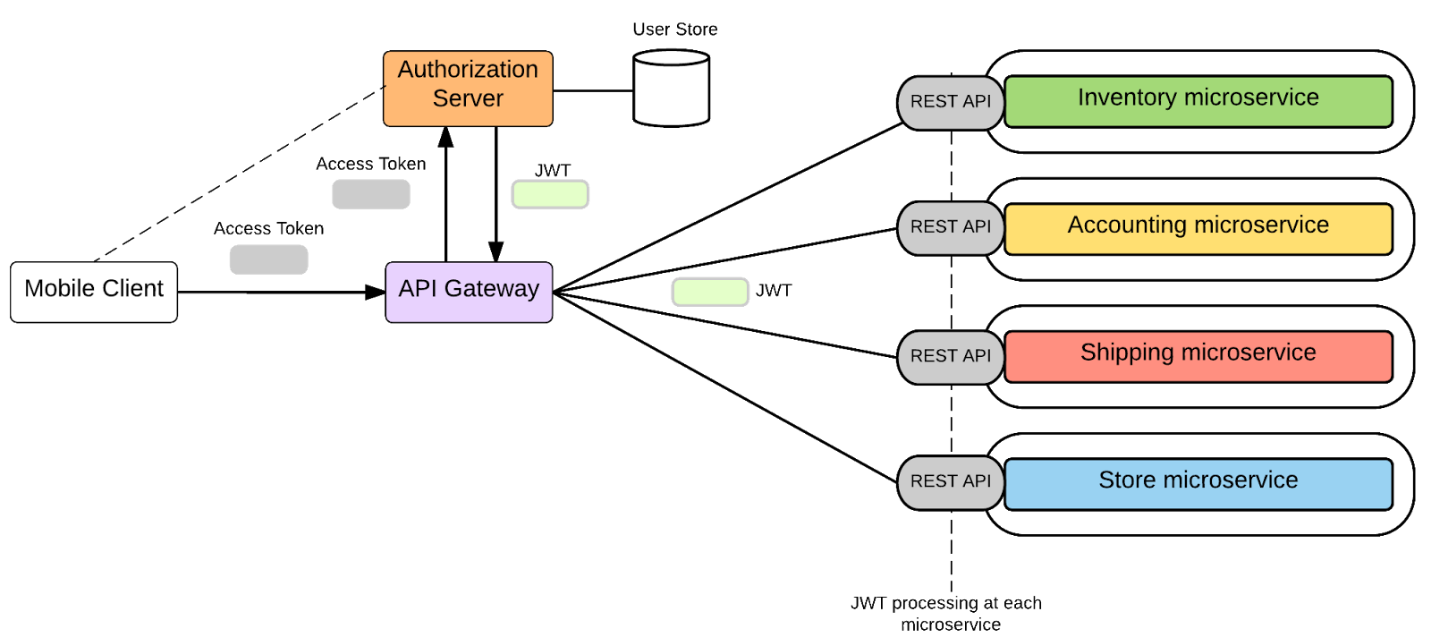
* Tính phức tạp: Kiến trúc microservice có thể trở nên phức tạp nếu không được thiết kế và triển khai đúng cách.
* Chi phí: Việc triển khai kiến trúc microservice có thể tốn kém hơn so với kiến trúc nguyên khối.
  1. **Ocelot Gateway**

Ocelot Gateway là một API Gateway mã nguồn mở được viết bằng .NET. API Gateway là một thành phần phần mềm nằm ở phía trước các dịch vụ microservice, đóng vai trò là một điểm truy cập duy nhất cho các dịch vụ này. Ocelot Gateway cung cấp một số tính năng hữu ích cho các ứng dụng microservice, bao gồm:

* Routing: Ocelot Gateway có thể được sử dụng để định tuyến các yêu cầu HTTP đến các dịch vụ microservice cụ thể.
* Caching: Ocelot Gateway có thể được sử dụng để lưu trữ các kết quả của các yêu cầu HTTP trong bộ nhớ cache, giúp tăng hiệu suất ứng dụng.
* Tính bảo mật: Ocelot Gateway có thể được sử dụng để xác thực và ủy quyền người dùng, giúp bảo vệ ứng dụng khỏi các truy cập trái phép.
* Mở rộng: Ocelot Gateway có thể được mở rộng bằng cách thêm các middleware và filter tùy chỉnh.

Trong đồ án, chúng em đã áp dụng được 4 tính năng của Ocelot Gateway đó là Routing, Caching, LoadBalancer, Authorization. Việc sử dụng Ocelot Gateway đã giúp cho đề tài đạt được những lưu điểm sau:

* Khi dữ liệu đang ở trong cache, API Gateway sẽ không tạo HTTP request đến microservice của chúng ta. Điều này có ý nghĩa tiết kiệm tài nguyên ở microservice. Khi cache hết hạn, API Gateway sẽ tạo request đến microservice 1 lần nữa và lưu trong cache.
* Bảo mật cho API bằng cách ngăn chặn các truy cập trái phép bằng JWT Bearer.



Hình 3.2.1 Mô hình minh họa

* 1. **Restful API**

API là từ viết tắt của cụm từ Application Programming Interface, đây là tập hợp những quy tắc và cơ chế mà theo đó thì: Một ứng dụng hoặc một thành phần nào đó sẽ tương tác với một ứng dụng hoặc một số thành phần khác. API có thể sẽ được trả về dữ liệu mà người dùng cần cho chính ứng dụng của bạn với những kiểu dữ liệu được dùng phổ biến như JSON hoặc XML.

REST (***Representational State Transfer***) là một kiến trúc được sử dụng để thiết kế các API cho các ứng dụng web. Restful API định nghĩa các quy tắc để tạo các API dựa trên giao thức HTTP.

Dưới đây là các nguyên tắc cơ bản để thiết kế các API theo kiến trúc Restful:

|  |  |
| --- | --- |
| **Bảng nguyên tắc khi thiết kế API theo kiến trúc** Restful | |
| **Nguyên tắc** | **Nội dung** |
| Sử dụng phương thức HTTP đúng | Restful API sử dụng các phương thức HTTP như GET, POST, PUT, PATCH, DELETE để thực hiện các hoạt động trên các tài nguyên. GET được sử dụng để truy vấn tài nguyên, POST để tạo mới tài nguyên, PUT và PATCH để cập nhật tài nguyên và DELETE để xóa tài nguyên. |
| Sử dụng URI mô tả tài nguyên | Restful API sử dụng URI (Uniform Resource Identifier) để định danh tài nguyên. URI phải được thiết kế sao cho dễ đọc, dễ hiểu và dễ bảo trì. |
| Sử dụng các trạng thái HTTP đúng | Restful API sử dụng các trạng thái HTTP để trả về kết quả cho yêu cầu của client. Ví dụ: 200 OK để trả về kết quả thành công, 404 Not Found để trả về lỗi tài nguyên không tồn tại. |
| Sử dụng định dạng dữ liệu đúng | Restful API sử dụng các định dạng dữ liệu như **JSON hoặc XML** để trao đổi dữ liệu giữa client và server. Định dạng dữ liệu phải được chọn sao cho phù hợp với nhu cầu sử dụng của ứng dụng. |
| Sử dụng các tham số truy vấn và tham số đường dẫn đúng | Restful API sử dụng các tham số truy vấn và tham số đường dẫn để truy cập vào các tài nguyên. Tham số truy vấn được sử dụng để lọc dữ liệu, còn tham số đường dẫn được sử dụng để định danh tài nguyên. |
| Sử dụng các token bảo mật đúng | Restful API sử dụng các token bảo mật như JWT (JSON Web Token) để xác thực và phân quyền cho các yêu cầu. Token bảo mật được mã hóa và chứa các thông tin về người dùng và quyền truy cập. |

* 1. **Repository Pattern**

Repository pattern là một mẫu thiết kế phần mềm cho phép truy cập và quản lý dữ liệu một cách trừu tượng. Mẫu này tách biệt việc truy cập dữ liệu khỏi các lớp ứng dụng, giúp cho code trở nên linh hoạt và dễ bảo trì hơn.

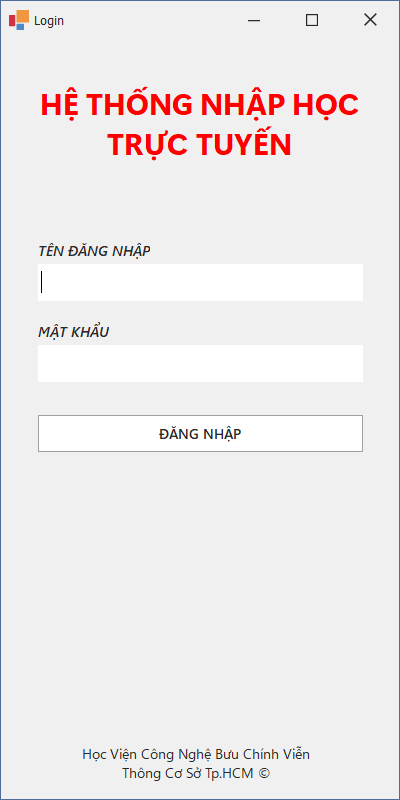
Repository pattern mang lại một số lợi ích sau:

* Tăng tính trừu tượng: Repository pattern tách biệt việc truy cập dữ liệu khỏi các lớp ứng dụng. Điều này giúp cho code của bạn trở nên linh hoạt và dễ bảo trì hơn.
* Dễ dàng thay đổi cơ sở dữ liệu: Nếu bạn muốn thay đổi cơ sở dữ liệu, bạn chỉ cần thay đổi lớp ConcreteRepository mà không cần thay đổi các lớp ứng dụng.
* Dễ dàng kiểm tra: Repository pattern giúp cho việc kiểm tra code trở nên dễ dàng hơn. Bạn có thể tạo các lớp mock cho lớp ConcreteRepository để kiểm tra các lớp ứng dụng một cách độc lập.

PHẦN 3: THIẾT KẾ GIAO DIỆN

1. Giao diện trên ứng dụng desktop

1.1. Giao diện đăng nhập

**

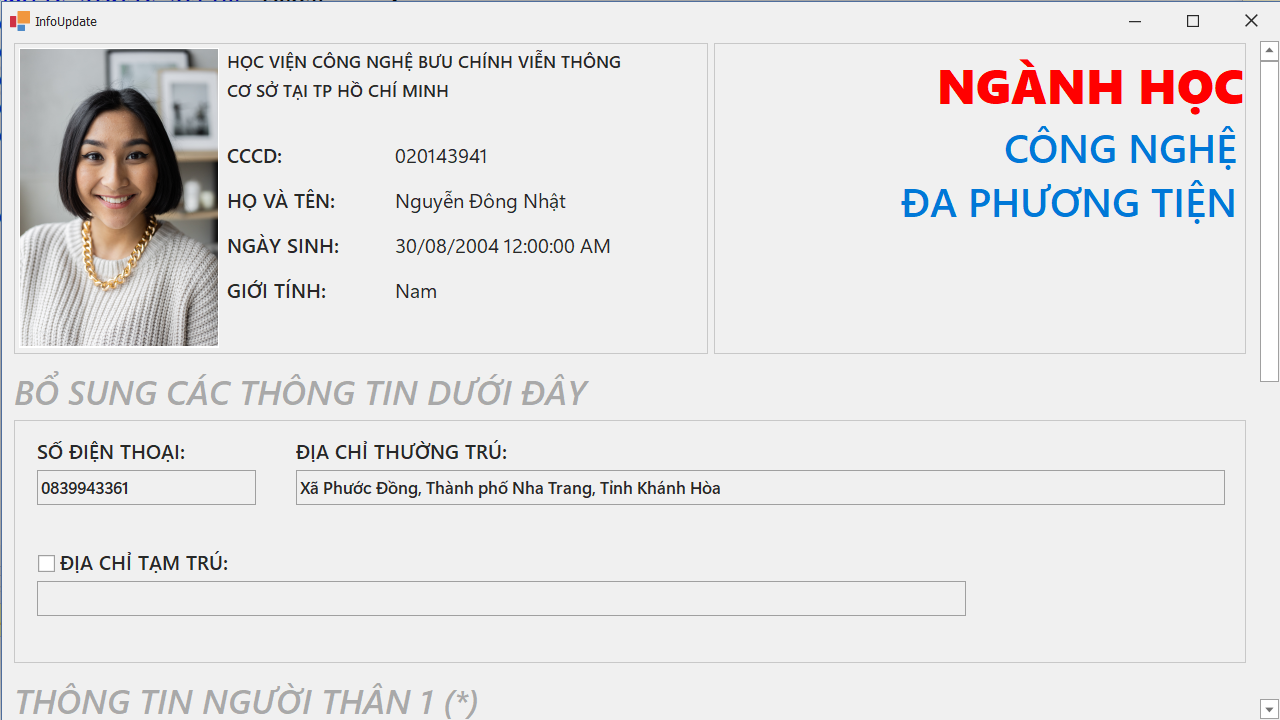
*Hình 3.1*

1.2. Giao diện trang chủ

**

*Hình 3.2*

1.3. Giao diện thông tin cá nhân

**

*Hình 3.3*

1.4. Giao diện upload hồ sơ

**

*Hình 3.4.1*

**

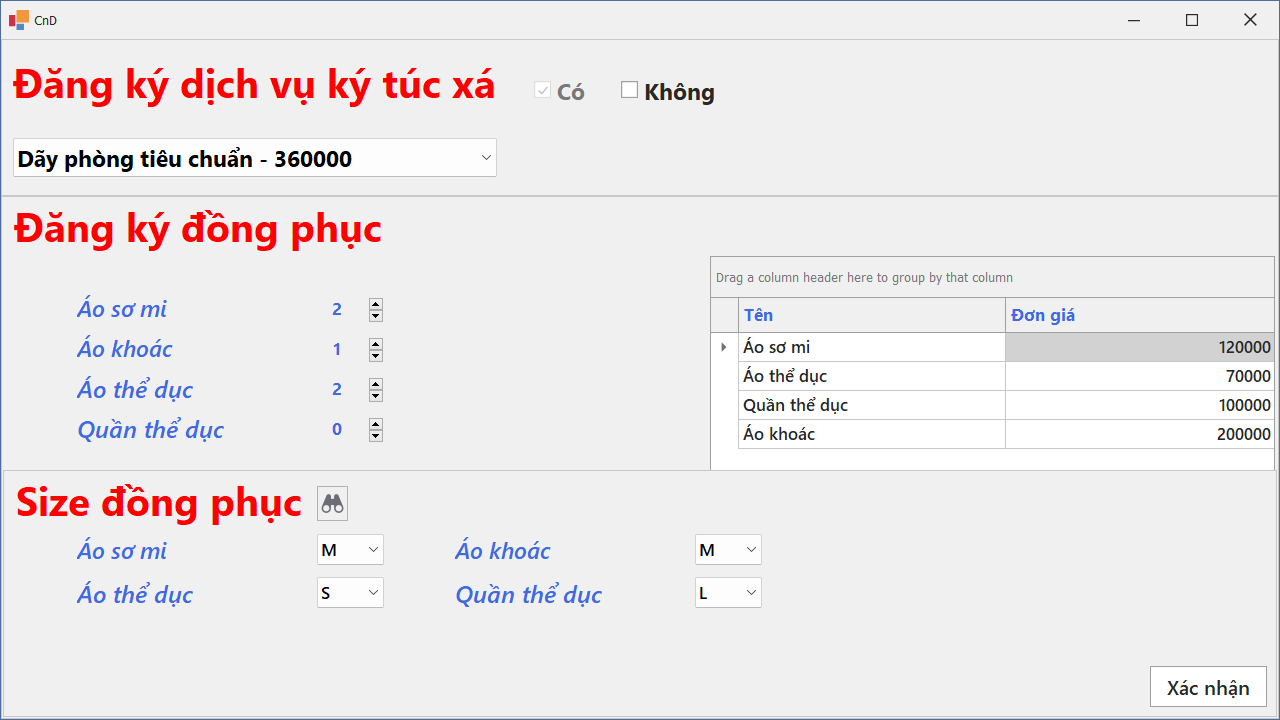
*Hình 3.4.2*

1.5. Giao diện đăng ký thẻ sinh viên liên kết

**

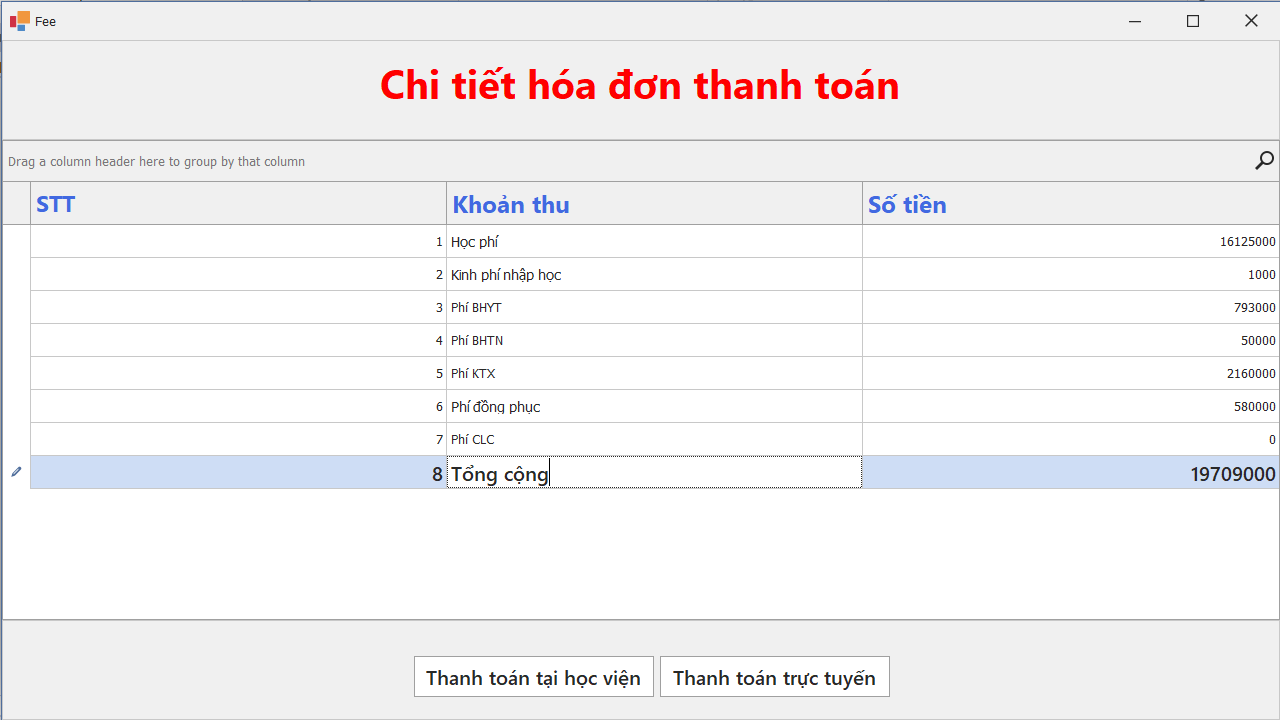
*Hình 3.5*

1.6. Giao diện đăng ký ký túc xá và mua đồng phục

**

*Hình 3.6*

1.7. Giao diện thanh toán

**

*Hình 3.7*

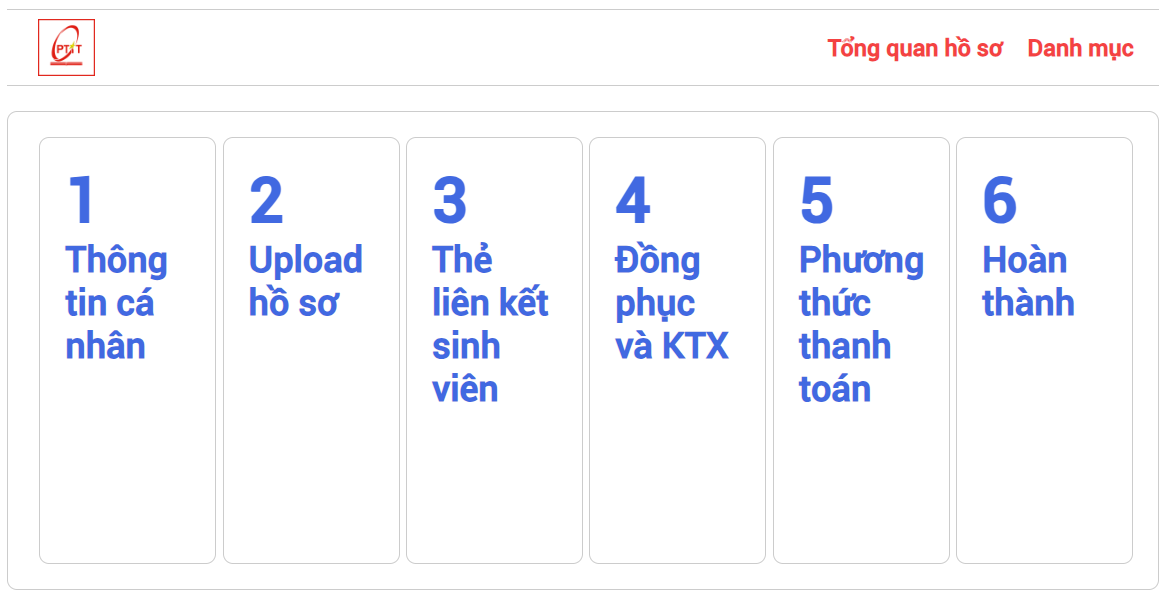
2. Giao diện trên website

2.1. Giao diện đăng nhập

**

*Hình 3.8*

2.2. Giao diện trang chủ

**

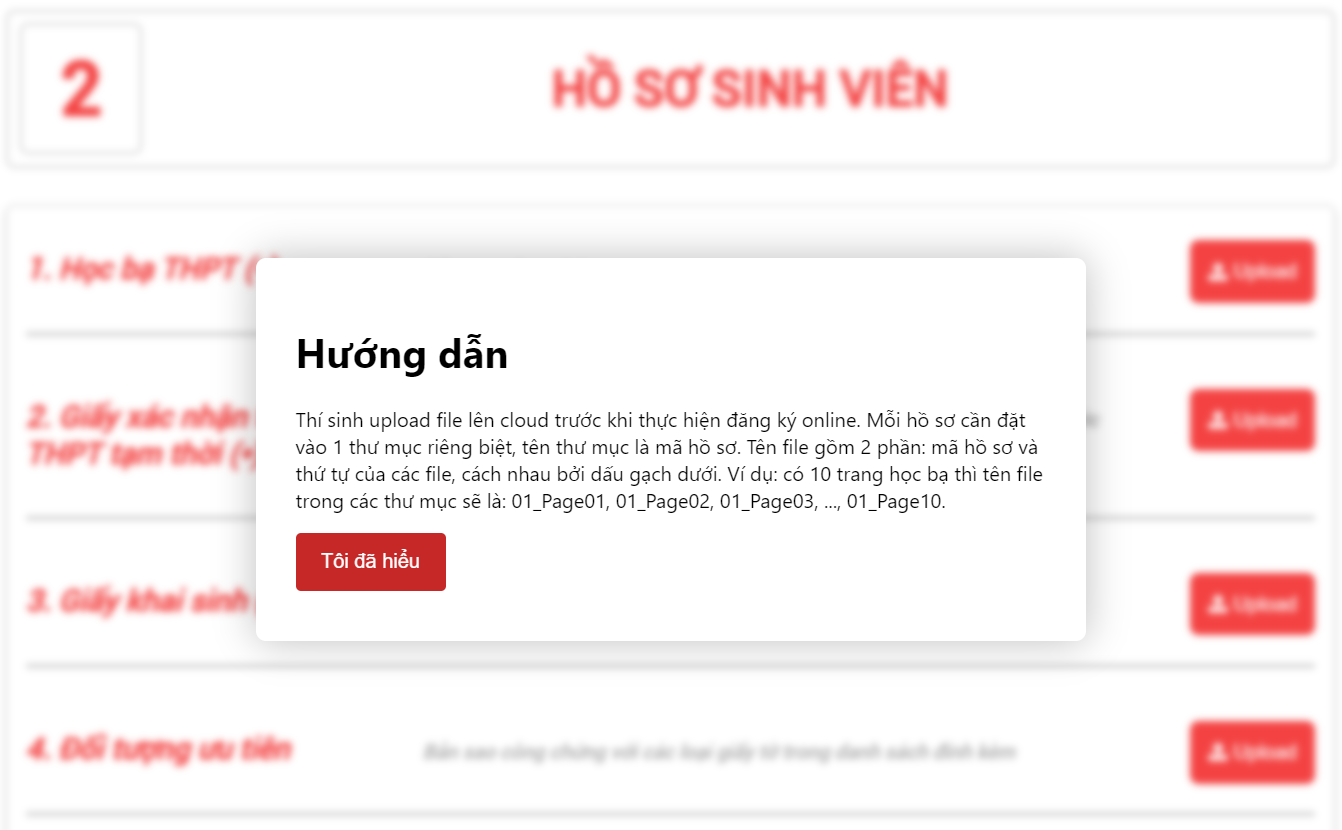
*Hình 3.9*

2.3. Giao diện thông tin cá nhân

**

*Hình 3.10.1*

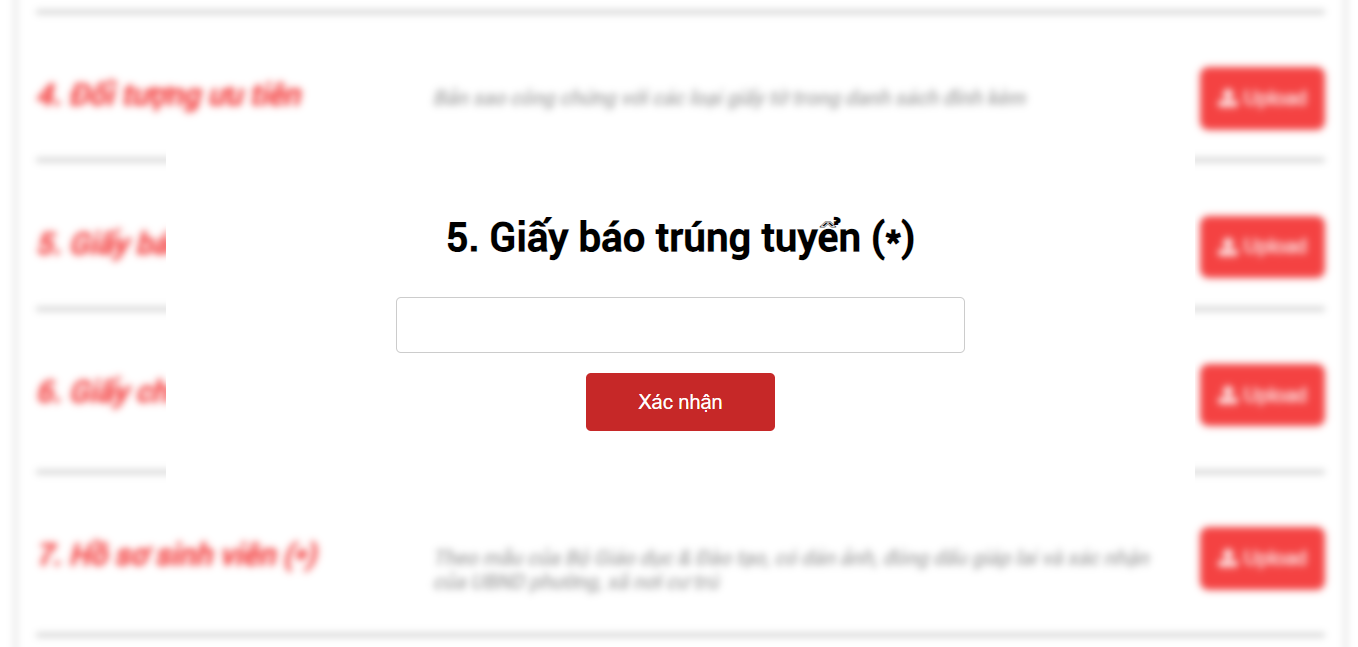
2.4. Giao diện upload hồ sơ

**

*Hình 3.11.1*

**

*Hình 3.11.2*

**

*Hình 3.11.3*

2.5. Giao diện đăng ký thẻ sinh viên liên kết

**

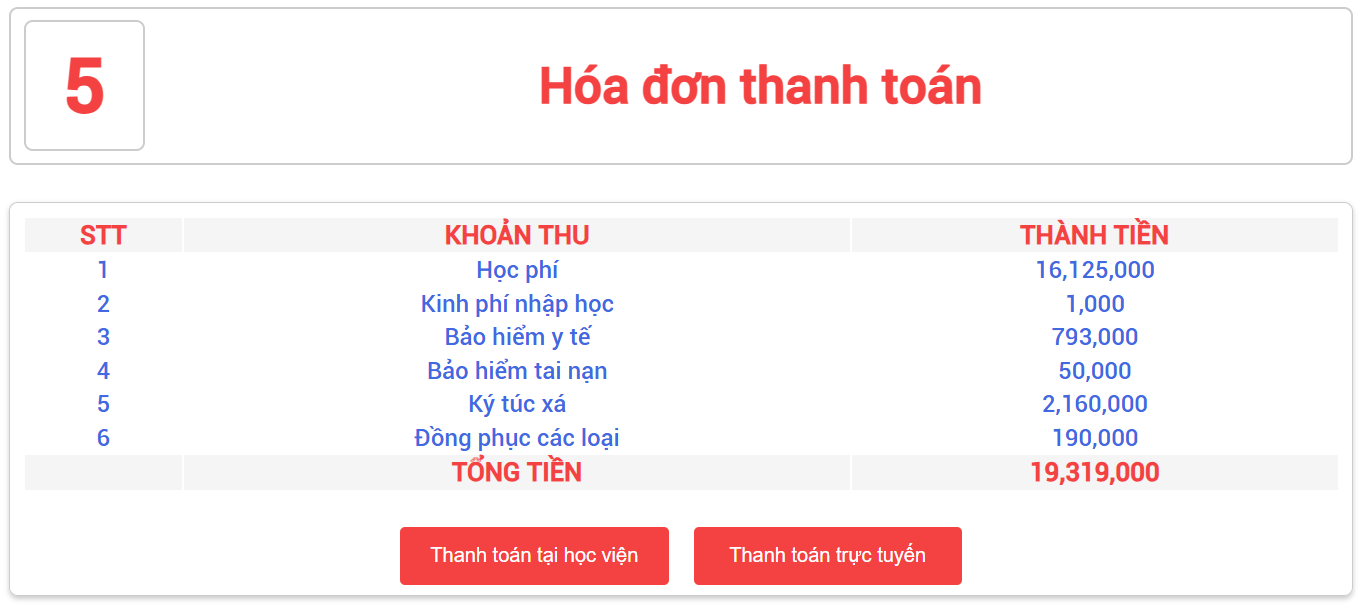
*Hình 3.12*

2.6. Giao diện đăng ký ký túc xá và mua đồng phục

**

*Hình 3.13*

2.7. Giao diện thanh toán

**

*Hình 3.14*

Phần 4: Tài liệu hướng dẫn sử dụng API

1.Giới thiệu:

API có dùng phương thức xác thực JWT Bearer, khi gọi API cần thêm các truyền vào header trong mỗi lần gọi API có yêu cầu Authorize. Các phương thức cần Auth (Authorize) nếu chưa xác thực thì sẽ trả về mã lỗi 401.

* [Key]Authorization:Bearer<dấu cách><token>
* Ví dụ: Bearer<khoảngtrắng>eyJhbGciOiJIUzUxMiIsInR5cCI6IkpXVCJ9.eyJVc2VyTmFtZSI6IjAwMDIxNTUyMDU0MSIsIm5iZiI6MTcwMjM3MjA5OCwiZXhwIjoxNzAyMzczMjk4LCJpYXQiOjE3MDIzNzIwOTh9.qROFeLzQs1mwzZfMMayesat43PudA4BzzgxE2UEEJBWTH7IPjhzwSW5CLu5zOM46AAB67Le0lTEqwngthiDdKw

2.SinhVien

* 1. **Get phiếu thông tin sinh viên [Auth]**
* Phương thức: GET
* Cấu trúc: [http://localhost:5000/apigateway/SinhVien/{cccd}](http://localhost:5000/apigateway/SinhVien/%7bcccd%7d)
* Tham số: {cccd}: số cccd của sinh viên cần lấy thông tin
* Url: <http://localhost:5000/apigateway/SinhVien/000215520541>
* Kết quả trả về:

+ Thành công: 200

**{**

**"soCCCD": "000215520541",**

**"hoTen": "Nguyễn Trần Quốc Tuấn ",**

**"quocTich": "string",**

**"gioiTinh": "string ",**

**"ngaySinh": "2023-11-30T00:00:00",**

**"thuongTru": "string",**

**"sdt": "string",**

**"email": "string",**

**"avatarPath": null,**

**"nguoiThan1": {**

**"hoTen": "string",**

**"namSinh": "2002",**

**"quocTich": "string",**

**"danToc": "string",**

**"ngheNghiep": "string",**

**"sdt": "string",**

**"diaChi": "string"**

**},**

**"nguoiThan2": {**

**"hoTen": "string",**

**"namSinh": "2002",**

**"quocTich": "string",**

**"danToc": "string",**

**"ngheNghiep": "string",**

**"sdt": "string",**

**"diaChi": "string"**

**}**

**}**

+ Lỗi: 404 Not Found, 400 Bad Request

* 1. **Chỉnh sửa phiếu thông tin sinh viên[Auth]**
* Phương thức: PUT
* Cấu trúc: [http://localhost:5000/apigateway/SinhVien/updateSV/{cccd}](http://localhost:5000/apigateway/SinhVien/updateSV/%7bcccd%7d)
* Tham số: {cccd} số cccd của sinh viên cần sửa đổi
* Body:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Required** | **Default** | **Description** |
| soCCCD | string | Có | Số cccd có sẵn trong db | Số CCCD |
| hoTen | string | Có |  |  |
| quocTich | string | Có |  |  |
| gioiTinh | string | Có |  |  |
| ngaySinh | datetime | Có |  |  |
| thuongTru | string | Có |  |  |
| sdt | string | Có |  |  |
| email | string | Không |  |  |
| avartarPath | string | Có |  | Đường dẫn của ảnh thẻ |
| nguoiThan1/nguoiThan2 tương tự nhau, phải đầy đủ 2 người thân | | |  |  |
| hoTen | string | Có |  |  |
| namSinh | string | Có |  |  |
| quocTich | string | Có |  |  |
| danToc | string | Có |  |  |
| ngheNghiep | string | Có |  |  |
| sdt | string | Có |  |  |
| diachi | string | Có |  |  |

* Url:<http://localhost:5000/apigateway/SinhVien/updateSV/000215520541>
* Kết quả trả về:

+ Thành công: HTTP Code: 200

+ Lỗi không tìm thấy sinh viên: 404

+ Lỗi số cccd trong body không dùng với {cccd} trên đường dẫn: 400

+ Lỗi do server không thực hiện được việc cập nhật: 500, “Có lỗi xảy ra khi cập nhật sinh viên”

* 1. **Thêm mới hồ sơ sinh viên [Auth]**
* Phương thức: POST
* Cấu trúc: [http://localhost:5000/apigateway/SinhVien/hoso/{cccd}](http://localhost:5000/apigateway/SinhVien/hoso/%7bcccd%7d)
* Tham số: {cccd}: số cccd của sinh viên cần thêm mới hồ sơ
* Body:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Required** | **Default** | **Description** |
| maHoSo | string | Có |  | Mã hồ sơ |
| soCCCD | string | Có |  | Số cccd của sv |
| duongDan | string | có |  | Đường dẫn chứa file hồ sơ |

* Url: <http://localhost:5000/apigateway/SinhVien/hoso/000215520541>
* Kết quả trả về:

+ Thành công: 201

+ Lỗi không tìm thấy sinh viên: 404

+ Lỗi do server không thực hiện được việc cập nhật: 500, “Có lỗi xảy ra khi cập nhật hồ sơ”

+Lỗi khác: 400 Bad Request

* 1. **Đăng ký ngân hàng [Auth]**
* Phương thức: POST
* Cấu trúc: [http://localhost:5000/apigateway/SinhVien/banking/{cccd}](http://localhost:5000/apigateway/SinhVien/banking/%7bcccd%7d)
* Tham số: {cccd}: số cccd của sinh viên cần thêm mới hồ sơ
* Body:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Required** | **Default** | **Description** |
| hoTenInTrenThe | string | Có |  |  |
| soCCCD | string | Có |  | Số cccd của sv |
| ngayCap | string | Có |  | Ngày cấp của CCCD |
| SDT | string | Có |  |  |
| DichVu | String | Có |  | Các dịch vụ đi kèm như: Mobile banking hoặc Internet banking |

* Url: <http://localhost:5000/apigateway/SinhVien/banking/000215520541>
* Kết quả trả về:

+ Thành công: 201

+ Lỗi không tìm thấy sinh viên: 404

+ Lỗi do server không thực hiện được việc cập nhật: 500, “Có lỗi xảy ra khi cập nhật”

+Lỗi khác: 400 Bad Request

1. Học Phí
   1. **Lấy tổng học phí [Auth]**

* Phương thức: GET
* Cấu trúc: [http://localhost:5000/apigateway/HocPhi/tonghocphi/{cccd}](http://localhost:5000/apigateway/HocPhi/tonghocphi/%7bcccd%7d)
* Tham số: {cccd}: số cccd của sinh viên cần lấy thông tin
* Url:<http://localhost:5000/apigateway/HocPhi/tonghocphi/000215520541>
* Kết quả trả về:

+ Thành công: HTTP Code 200, số tiền tổng học phí (decimal).

+ Lỗi không tìm thấy sinh viên: 404

+ Lỗi khác: 400

* 1. **Lấy chi tiết học phí [Auth]**
* Phương thức: GET
* Cấu trúc: [http://localhost:5000/apigateway/HocPhi/chitiethp/{cccd}](http://localhost:5000/apigateway/HocPhi/chitiethp/%7bcccd%7d)
* Tham số: {cccd}: số cccd của sinh viên cần lấy thông tin
* Url:<http://localhost:5000/apigateway/HocPhi/chitiethp/000215520541>
* Kết quả trả về:

+ Thành công: 200, đơn vị VND  
**{**

**"hocPhi": 23650000,**

**"kinhPhiNhapHoc": 1000000,**

**"phiBHYT": 793000,**

**"phiBHTN": 50000,**

**"phiKTX": 2160,**

**"phiDongPhuc": 310000,**

**"phiCLC": 300000**

**}**

+ Lỗi không tìm thấy sinh viên: 404

+ Lỗi khác: 400

* 1. **Xem thông tin hóa đơn của sinh viên [Auth]**
* Phương thức: GET
* Cấu trúc: [http://localhost:5000/apigateway/HocPhi/hoadon/{cccd}](http://localhost:5000/apigateway/HocPhi/hoadon/%7bcccd%7d)
* Tham số: {cccd}: số cccd của sinh viên cần lấy thông tin
* Url:<http://localhost:5000/apigateway/HocPhi/hoadon/000215520541>
* Kết quả trả về:

+ Thành công: 200

**{**

**"maHD": "14217487",**

**"soTien": 1753016,**

**"mst": null,**

**"thoiDiem": "20231204220950",**

**"noiDung": "000215520541ThanhToanHocPhi, DongPhuc,KTX 17530160"**

**}**

+ Lỗi không tìm thấy sinh viên: 404

+ Sinh viên chưa thanh toán: 404, “Không tìm thấy hóa đơn/Chưa thanh toán”

+ Lỗi khác: 400

1. VnPay (Thanh toán học phí)
   1. **Get link thanh toán[Auth]**

* Phương thức: GET
* Cấu trúc: [http://localhost:5000/apigateway/HocPhi/getlink/{cccd}](http://localhost:5000/apigateway/HocPhi/hoadon/%7bcccd%7d)
* Tham số: {cccd}: số cccd của sinh viên cần lấy link thanh toán
* Url:<http://localhost:5000/apigateway/HocPhi/getlink/000215520541>
* Kết quả trả về:

+ Thành công: link dẫn đến trang thanh toán

**https://sandbox.vnpayment.vn/paymentv2/vpcpay.html?vnp\_Amount=1619300000&vnp\_Command=pay&vnp\_CreateDate=20231212214713&vnp\_CurrCode=VND&vnp\_IpAddr=192.168.1.12&vnp\_Locale=vn&vnp\_OrderInfo=052300012260ThanhToanHocPhi%2C++16193000&vnp\_OrderType=hocphi&vnp\_ReturnUrl=https%3A%2F%2Flocalhost%3A7212%2Fapi%2FHocPhi%2FgetResponseJsonVnPay%2F&vnp\_TmnCode=DNSAKTB7&vnp\_TxnRef=638380144331572315&vnp\_Version=2.1.0&vnp\_SecureHash=a35a633b1cc478890315c56b7de3e8f8f43741c761683b5f08de0d43f69a2e458fb2ea50751e42a14503f15e8386654685c02f45b81370a68b53e50c84d5a5f1**

+ Không tìm thấy sinh viên: 404

+ Sinh viên đã thanh toán: 200, “Sinh viên đã thanh toán học phí online!”

* 1. **Trả về kết quả thanh toán**
* Phương thức: GET
* Cấu trúc: http://localhost:5000/apigateway/HocPhi/ketquathanhtoan
* Tham số: Không
* <Url:http://localhost:5000/apigateway/HocPhi/ketquathanhtoan>
* Sử dụng: Sau khi thực hiện thanh toán Vnpay sẽ Redirect về trang này để show kết quả thanh toán và nếu thành công thì lưu hóa đơn vào database.
* Kết quả trả về:

**{**

**"maHD": "14217487",**

**"soTien": 1753016,**

**"thoiDiem": "20231204220950",**

**"noiDung": "000215520541ThanhToanHocPhi, DongPhuc,KTX 17530160"**

**}**

1. Đồng phục
   1. **Get danh sách đồng phục [Auth]**

* Phương thức: GET
* Cấu trúc: http://localhost:5000/apigateway/DongPhuc/danhsachdongphuc
* Tham số: {cccd}: số cccd của sinh viên cần lấy thông tin
* Url:http://localhost:5000/apigateway/DongPhuc/danhsachdongphuc
* Kết quả trả về:

+ Thành công: 200

**[**

**{**

**"maDongPhuc": "dp004",**

**"tenDongPhuc": "Áo khoác PTIT",**

**"donGia": 200000**

**},**

**{**

**"maDongPhuc": "dp001",**

**"tenDongPhuc": "Áo PTIT",**

**"donGia": 120000**

**},**

**{**

**"maDongPhuc": "dp002",**

**"tenDongPhuc": "Áo thể dục",**

**"donGia": 70000**

**},**

**{**

**"maDongPhuc": "dp003",**

**"tenDongPhuc": "Quần thể dục",**

**"donGia": 100000**

**}**

**]**

+ Lỗi khác: 400

* 1. **Mua đồng phục [Auth]**
* Phương thức: GET
* Cấu trúc: http://localhost:5000/apigateway/DongPhuc/muadongphuc/{cccd}
* Tham số: {cccd}: số cccd của sinh viên mua đồng phục
* Body:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Required** | **Default** | **Description** |
| soCCCD | String | Có |  | Số CCCD |
| maDongPhuc | String | Có |  | Mã đồng phục |
| Kích cỡ | String | Có |  | Kích thước áo quần |
| soLuong | int | Có | 1 | Số lượng đồng phục |

* Url:<http://localhost:5000/apigateway/DongPhuc/muadongphuc/000215520541>
* Kết quả trả về:

+ Thành công: 200

+ Không tìm thấy sinh viên: 404

+ Lỗi khác: 400

1. Ký túc xá
   1. **Lấy danh sách phòng [Auth]**

* Phương thức: GET
* Cấu trúc: http://localhost:5000/apigateway/KTX/danhsachktx
* Tham số: không
* Url:http://localhost:5000/apigateway/KTX/danhsachktx
* Kết quả trả về:

+ Thành công: 200

**[**

**{**

**"maPhong": "DV",**

**"tenPhong": "Dãy phòng dịch vụ”,**

**"donGia": 480000**

**},**

**{**

**"maPhong": "TC",**

**"tenPhong": "Dãy phòng tiêu chuẩn",**

**"donGia”:360000**

**}**

**]**

+ Lỗi khác: 400

* 1. **Đăng ký ký túc xá [Auth]**
* Phương thức: POST
* Cấu trúc: http://localhost:5000/apigateway/KTX/dangky/{cccd}
* Tham số: {cccd}: số cccd của sinh viên cần lấy link thanh toán
* Body:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field | Type | Required | Default | Description |
| soCCCD | String | Có |  | Số CCCD |
| maPhong | String | Có |  | Mã phòng |
| Ghi chú | String | Không |  | Ghi chú cho việc đăng ký |

* Url:http://localhost:5000/apigateway/KTX/danhsachktx
* Kết quả trả về:

+ Thành công: 200

+ Lỗi sinh viên đã đăng ký phòng này trong năm: 200, “Lỗi! Sinh viên đã đăng ký phòng này!”

+ Lỗi không tìm thấy sinh viên: 404

+ Lỗi khác: 400

1. Login
   1. **Đăng nhập**

* Phương thức: POST
* Cấu trúc: http://localhost:5000/apigateway/Login
* Tham số: không
* Body:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field | Type | Required | Default | Description |
| soCCCD | String | Có |  | Số CCCD |
| matKhau | String | Có | Ngày sinh sinh viên dưới format ddMMyyyy |  |

* Url:http://localhost:5000/apigateway/Login
* Kết quả trả về: **Token chỉ hiệu lực trong vòng 20 phút**

+ Thành công: 200

**{**

**"success": true,**

**"message": "Authenticate success",**

**"data": "eyJhbGciOiJIUzUxMiIsInR5cCI6IkpXVCJ9.eyJVc2VyTmFtZSI6IjAwMDIxNTUyMDU0MSIsIm5iZiI6MTcwMjQ1OTQxNCwiZXhwIjoxNzAyNDYwNjE0LCJpYXQiOjE3MDI0NTk0MTR9.n1zOhcsprCe9CfqTohnf4XDEP8MEadyoJCTia6K5stv9UZkdFQ3\_ev92mgS11AMfWq3sKvPqOjEHh\_kkpd11CA"**

**}**

+ Lỗi sai cccd hoặc mật khẩu: 200

**{**

**"success": false,**

**"message": "Invalid username/password",**

**"data": null**

**}**

Phụ lục

1. Danh mục các hồ sơ cần cho quá trình nhập học

Bản sao hợp lệ Học bạ Trung học phổ thông.

Bản sao hợp lệ Bằng tốt nghiệp trung học phổ thông (đối với học sinh tốt nghiệp từ năm 2022 trở về trước) hoặc Giấy chứng nhận tốt nghiệp trung học phổ thông tạm thời (đối với học sinh tốt nghiệp năm 2023).

Bản sao hợp lệ Giấy khai sinh.

Bản sao hợp lệ các giấy tờ pháp lý minh chứng đối tượng ưu tiên được hưởng (nếu có).

Giấy triệu tập trúng tuyển của Học viện Công nghệ Bưu chính Viễn thông (bản gốc).

Bản chính Giấy chứng nhận học sinh giỏi quốc gia hoặc cấp tỉnh, thành phố trực thuộc TW; Chứng chỉ quốc tế, Giấy chứng nhận kết quả thi đánh giá năng lực năm 2023 (đối với các thí sinh thuộc đối tượng xét trúng tuyển sớm – Xét tuyển kết hợp hoặc Đánh giá năng lực).

Hồ sơ sinh viên (theo mẫu của Bộ Giáo dục và Đào tạo) có dán ảnh, đóng dấu giáp lai và xác nhận của UBND phường, xã nơi cư trú.

Giấy chuyển sinh hoạt Đảng, Đoàn (nếu là Đảng viên, Đoàn viên) về Đảng bộ hoặc Đoàn Thanh niên Cộng sản Hồ Chí Minh của Học viện Công nghệ Bưu chính Viễn thông.

Giấy giới thiệu di chuyển đăng nghĩa vụ quân sự (đối với Nam giới) do Ban chỉ huy quân sự quận, huyện nơi đăng ký hộ khẩu thường trú cấp.

Bản chụp Thẻ bảo hiểm y tế.

06 ảnh cỡ 3×4, chụp trong vòng 6 tháng trở lại đây (ghi họ tên và ngày sinh vào mặt sau).

Chứng chỉ tiếng Anh (nếu có trong danh mục đính kèm), Chứng chỉ Giáo dục quốc phòng và An ninh (nếu có).