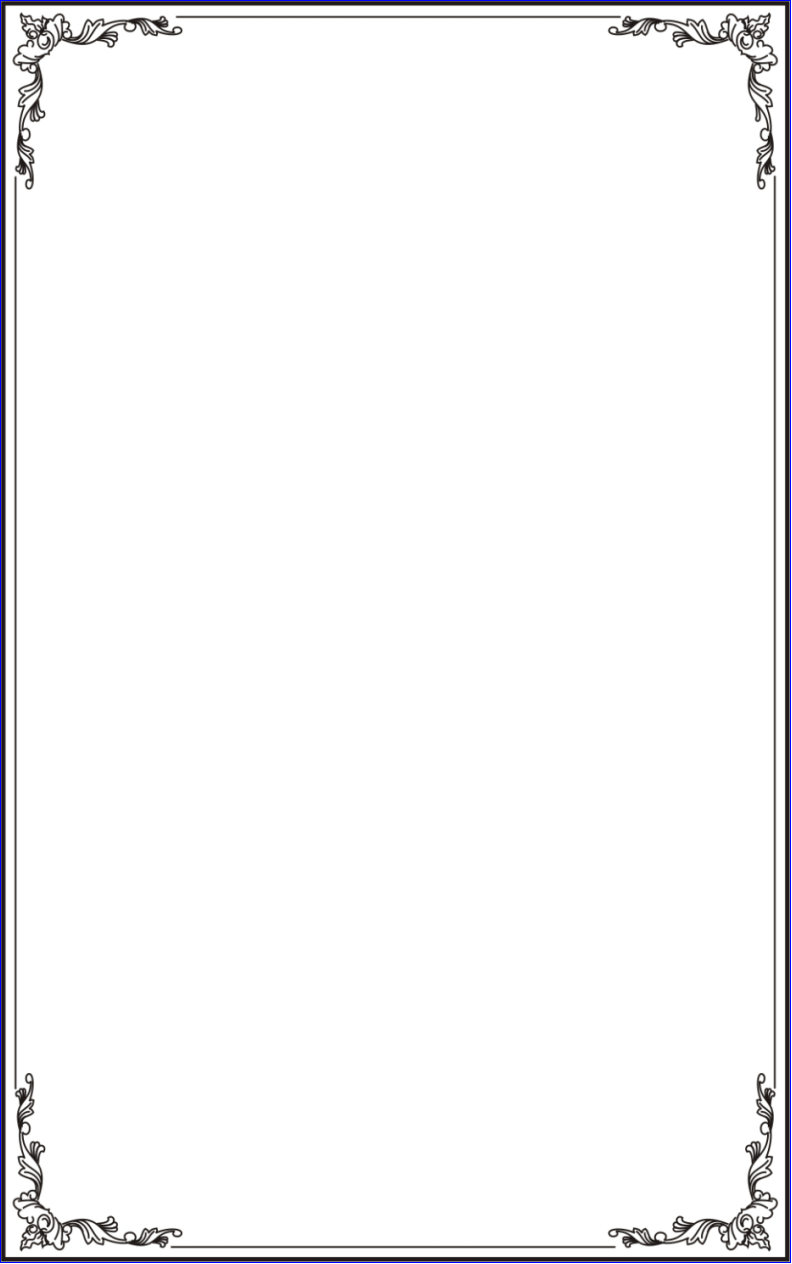
****TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

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

**BÁO CÁO THÍ NGHIỆM/THỰC NGHIỆM**

**LẬP TRÌNH JAVA**

**XÂY DỰNG ỨNG DỤNG GIÚP SINH VIÊN, GIÁO VIÊN ĐĂNG KÝ THAM QUAN DOANH NGHIỆP KHOA CNTT, HAUI**

| **GVHD:** | **ThS. Nguyễn Ngọc Quang.** |
| --- | --- |
| **Sinh viên:** |  |
|  | **1. Nguyễn Trọng Tuấn – 2021606069.** |
|  | **2. Nguyễn Tài Tuệ - 2021607117.** |
|  | **3. Võ Thị Quỳnh – 2021604331.** |
|  | **4. Vũ Thị Thanh Tâm – 2021603841.** |
|  | **5. Nguyễn Đông Bắc - 2021603451.** |

**Nhóm: 4**

**Lớp: 20233IT6019001 - Khóa: 16**

***Hà Nội – Năm 2024***

# MỤC LỤC

[MỤC LỤC 2](#_Toc159769706)

[Danh mục hình ảnh 4](#_Toc159769707)

[PHẦN 1: MỞ ĐẦU 5](#_Toc159769708)

[1.1. Lý do chọn đề tài 5](#_Toc159769709)

[1.2. Mục tiêu và ý nghĩa 5](#_Toc159769710)

[1.3. Nội dung học tập và các kỹ năng, kiến thức then chốt 6](#_Toc159769711)

[1.4. Bố cục chính 7](#_Toc159769712)

[PHẦN 2: KẾT QUẢ NGHIÊN CỨU 8](#_Toc159769713)

[2.1. Giới thiệu 8](#_Toc159769714)

[2.2. Khảo sát hệ thống 9](#_Toc159769715)

[2.2.1. Khảo sát sơ bộ 9](#_Toc159769716)

[2.2.2. Tài liệu đặc tả yêu cầu 10](#_Toc159769717)

[2.3. Thực hiện bài toán 14](#_Toc159769718)

[2.3.1. Võ Thị Quỳnh – Giao diện quản lý lớp học 14](#_Toc159769719)

[2.3.2. Nguyễn Tài Tuệ – Giao diện quản lý thông tin giáo viên, giao diện quản lý thông tin sinh viên, giao diện trang chủ (với tài khoản quản trị viên), giao diện đăng nhập hệ thống, giao diện quản lý tham quan doanh nghiệp. 18](#_Toc159769720)

[2.3.3 Nguyễn Đông Bắc – Màn hình giao diện dành cho sinh viên. 38](#_Toc159769721)

[2.3.4 Nguyễn Trọng Tuấn- Màn hình giao diện dành cho giáo viên. 42](#_Toc159769722)

[2.3.5 Võ Thị Quỳnh, Vũ Thị Thanh Tâm – Giao diện quản lý hệ thống. 44](#_Toc159769723)

[PHẦN 3: KẾT LUẬN VÀ BÀI HỌC KINH NGHIỆM 47](#_Toc159769724)

[3.1. Nội dung đã thực hiện 47](#_Toc159769725)

[3.2 Hướng phát triển tương lai 47](#_Toc159769726)

[TÀI LIỆU THAM KHẢO 49](#_Toc159769727)

Danh mục hình ảnh

[Hình 1: Giao diện quản lý thông tin lớp học 15](#_Toc159769802)

[Hình 2: Giao diện danh sách sinh viên của lớp học 16](#_Toc159769803)

[Hình 3: Giao diện quản lý thông tin giáo viên 19](#_Toc159769804)

[Hình 4: Giao diện danh sách sinh viên của lớp học 20](#_Toc159769805)

[Hình 5: Giao diện quản lý thông tin sinh viên. 24](#_Toc159769806)

[Hình 6: Giao diện quản lý chuyến tham quan của sinh viên. 25](#_Toc159769807)

[Hình 7: Giao diện trang chủ cho quản trị viên 29](#_Toc159769808)

[Hình 8: Giao diện đăng nhập 33](#_Toc159769809)

[Hình 9: Giao diện quản lý tham quan doanh nghiệp 36](#_Toc159769810)

[Hình 10: Giao diện danh sách sinh viên tham quan doanh nghiệp 37](#_Toc159769811)

[Hình 11: Giao diện trang chủ 39](#_Toc159769812)

[Hình 12: Giao diện xem các chuyến tham quan hôm nay 40](#_Toc159769813)

[Hình 13: Giao diện xem các chuyến tham quan sắp diễn ra 40](#_Toc159769814)

[Hình 14: Giao diện quản lý tài khoản cá nhân 41](#_Toc159769815)

[Hình 15: Giao diện các chuyến tham quan đã đăng ký 42](#_Toc159769816)

[Hình 16: Giao diện trang chủ của giáo viên 43](#_Toc159769817)

[Hình 17: Giao diện Các chuyến tham quan đã đăng ký 43](#_Toc159769818)

[Hình 18: Giao diện Tài khoản cá nhân. 44](#_Toc159769819)

[Hình 19: Giao diện chuyến tham quan hôm nay của bạn 44](#_Toc159769820)

[Hình 20: Giao diện quản lý thông tin hệ thống. 46](#_Toc159769821)

PHẦN 1: MỞ ĐẦU

1.1. Lý do chọn đề tài

Tham quan doanh nghiệp là một trong nhiều chương trình mà trường HaUI tổ chức cho sinh viên nhằm tăng sự hiểu biết của mình về doanh nghiệp và ngành nghề mình đang theo học. Sự tăng cường giao lưu giữa sinh viên và doanh nghiệp qua các chương trình tham quan không chỉ mang lại trải nghiệm thực tế mà còn giúp sinh viên hiểu rõ hơn về môi trường làm việc và yêu cầu của ngành nghề.

Một trong những vấn đề lớn mà nhiều tổ chức đang phải đối mặt là khó khăn trong việc tổ chức và quản lý thông tin sinh viên tham quan. Thông tin này bao gồm các chi tiết quan trọng như tên, lớp, thông tin liên lạc, cũng như các thông tin về doanh nghiệp, ngày tham quan, và xác nhận tham gia từ cả sinh viên và doanh nghiệp. Sự phức tạp của quy trình này thường dẫn đến những vấn đề như mất mát thông tin, sai sót trong quá trình ghi nhận, và khả năng theo dõi chưa linh hoạt.

Do đó, việc phát triển một hệ thống quản lý thông tin sinh viên tham quan doanh nghiệp trở thành một ưu tiên. Hệ thống này không chỉ giúp giảm thiểu những khó khăn trong việc quản lý thông tin mà còn tạo điều kiện cho việc tự động hóa một số công việc như xác nhận tham gia và tạo báo cáo tổng kết. Điều này không chỉ nâng cao hiệu suất làm việc mà còn đảm bảo tính chính xác và minh bạch trong quá trình quản lý sinh viên tham quan doanh nghiệp.

1.2. Mục tiêu và ý nghĩa

* **Mục tiêu:**
* Xây dựng và triển khai một hệ thống quản lý thông tin sinh viên tham quan doanh nghiệp
* Tích hợp các giải pháp vào một phần mềm quản lý và tổ chức chuyến tham quan doanh nghiệp một cách hiệu quả.
* Giải quyết các khó khăn trong quá trình tổ chức và quản lý thông tin liên quan đến hoạt động tham quan, từ việc thu thập thông tin đến việc tạo báo cáo tổng kết.
* Tối ưu hóa quy trình làm việc, giảm thiểu sai sót và mất mát thông tin, cung cấp một công cụ linh hoạt và dễ sử dụng cho cả sinh viên và doanh nghiệp.
* **Ý nghĩa:**
* Cải tiến chất lượng phần mềm quản lý chuyến tham quan doanh nghiệp để có thể nâng cái trình độ và kỹ năng áp dụng vào thực tế của sinh viên khoa Công nghệ thông tin trường Đại học Công nghiệp Hà Nội.
* Đóng góp về mặt phương pháp luận và thực nghiệm vào lĩnh vực quản lý và tổ chức chuyến tham quan doanh nghiệp Trường Đại học Công nghiệp Hà Nội một cách thực tế và hiệu quả.

1.3. Nội dung học tập và các kỹ năng, kiến thức then chốt

* **Nội dung học tập:**
* Cấu trúc của một chương trình Java
* Các kiểu dữ liệu và chuyển kiểu dữ liệu
* Các toán tử
* Các cấu trúc điều khiển
* Lớp và đối tượng trong Java
* Các hàm khởi tạo
* Phương thức tĩnh static
* Kế thừa, kết tập
* Tính trừu tượng, đa hình và interface
* Ghi đè phương thức
* Xử lý ngoại lệ
* I/O theo luồng và thao tác với tệp
* Collection
* Giao diện Java Swing
* **Kiến thức đã trang bị:**
* Lập trình Java cơ sở
* Lập trình Java hướng đối tượng
* Xử lý ngoại lệ và thao tác với tệp
* Lập trình với cấu trúc Collection
* Lập trình giao diện Java Swing
* **Kỹ năng then chốt:**
* Kỹ năng làm việc nhóm
* Kỹ năng phân tích và xử lý tình huống
* Kỹ năng thu thập và chuẩn hóa thông tin
* Kỹ năng xây dựng ý tưởng đề tài

1.4. Bố cục chính

- Bản báo cáo gồm 3 phần chính:

* Phần I: Mở đầu
* Phần II: Kết quả nghiên cứu
* Phần III: Kết luận và bài học kinh nghiệm

PHẦN 2: KẾT QUẢ NGHIÊN CỨU

2.1. Giới thiệu

Đối với sinh viên khoa Công nghệ Thông tin, việc tham gia hoạt động tham quan doanh nghiệp là vô cùng quan trọng giúp sinh viên nắm bắt thực tế về ngành nghề. Chúng tôi quyết định xây dựng một ứng dụng hỗ trợ sinh viên và giảng viên đăng ký tham quan doanh nghiệp một các hiệu quả. Đối với đề tài này chúng tôi sẽ sử dụng NetBeans làm môi trường phát triển chính.

Với những yêu cầu của đề tài chúng tôi quyết định áp dụng phương pháp phát triển phần mềm theo mô hình Agile để linh hoạt và có thể thích ứng nhanh chóng khi có những yêu cầu thay đổi. Mô hình Agile là một phương pháp phát triển phần mềm linh hoạt và tập trung vào việc tạo ra giá trị ngày từ giai đoạn sớm của dự án. Thay vì tiếp cận theo kiểu “Waterfall” truyền thống, mô hình Agile chia dự án thành các đoạn nhỏ giúp tăng tốc quá trình phát triển và tối ưu hóa sự linh hoạt

Để xây dựng lên ứng dụng chúng tôi đã sử dụng ngôn ngữ lập trình Java và các công cụ như NetBeans và Github. Trong đó NetBeans là một môi trường phát triển tích hợp (IDE) mã nguồn mở được thiết kế để hỗ trợ việc phát triển ứng dụng Java, cũng như nhiều ngôn ngữ khác. Netbeans hỗ trợ đa nền tảng, bao gồm Windows, macOS và Linux, điều này đã giúp cho đội ngũ phát triển làm việc trên nhiều hệ điều hành mà không gặp vấn đề. Ngoài ra giao diện của Netbeans được thiết kế để giúp người phát triển làm việc một các hiệu quả với các tính năng như code completion, debugging và quản lý dự án. NetBeans còn cung cấp các công cụ quản lý dự án giúp đơn giản hóa việc tổ chức mã nguồn, quản lý thư viện và kiểm soát phiên bản. Netbeans cũng giúp đội phát triển làm thuận tiện hiệu quả khi tích hợp sâu rộng với hệ thống kiểm soát phiên bản Git. GitHub là một dịch vụ lưu trữ mã nguồn trực tuyến, cho phép các nhóm phát triển cùng nhau làm việc trên dự án, theo dõi thay đổi trong mã nguồn, và quản lý các version của mã nguồn. Việc kết hợp sử dụng Netbeans cùng với GitHub đã giúp cho đội ngũ phát triển dễ dàng quản lý và theo dõi mã nguồn, quản lý các phiên bản hiệu quả.

Kết quả của quá trình phát triển sẽ là một ứng dụng vận hành trên nền tảng desktop, được xây dựng để hoạt động mượt mà trên các máy tính cá nhân. Điều này giúp cho sinh viên dễ dàng và sử dụng một các thuật tiện. Chương trình sẽ giúp cho sinh viên có thể xem và nắm bắt các thông tin về các buổi tham quan và các doanh nghiệp, cung cấp tính năng giúp sinh viên có thể đăng ký đi trải nghiệm tham quan và đưa ra những ý kiến cá nhân về buổi tham quan.

2.2. Khảo sát hệ thống

2.2.1. Khảo sát sơ bộ

Khoa Công Nghệ Thông Tin - Trường Đại học Công nghiệp Hà Nội (Hanoi University of Industry) là khoa trực thuộc Trường Đại học Công nghiệp Hà Nội được thành lập ngày 05/07/1999, có chức năng đào tạo nguồn nhân lực CNTT, hiện tại khoa có các hệ Cao đẳng, Đại học, liên thông từ Trung cấp lên Đại học và liên thông từ Cao đẳng lên Đại học, Thạc sỹ. Cùng với việc đào tạo, khoa còn có nhiệm vụ tổ chức các chuyến tham quan doanh nghiệp cho sinh viên nhằm hỗ trợ sinh viên trong việc hiểu rõ về nhiệm vụ của từng vị trí công việc thông qua các buổi tham quan, đồng thời gia tăng sự hợp tác chặt chẽ giữa doanh nghiệp và nhà trường nói chung, cũng như giữa doanh nghiệp và khoa nói riêng.

Hiện nay, khoa có khoảng 8000 sinh viên thuộc 5 các chuyên ngành khác nhau, khoảng hơn 100 cán bộ giáo viên, 20 doanh nghiệp đã hợp tác và trung bình một năm khoa tổ chức khoảng 40 - 50 chuyến tham quan doanh nghiệp cho sinh viên trong khoa, đến tham quan các doanh nghiệp mà trường hợp tác.

Để dễ dàng quản lý và tổ chức cho sinh viên và giáo viên của khoa Công Nghệ Thông Tin - Trường Đại học Công Nghiệp Hà Nội thì việc sử dụng phần mềm này là vô cùng cần thiết. Qua quá trình khảo sát và phân tích các dữ liệu thu được, phần mềm giúp sinh viên đăng ký tham quan doanh nghiệp đặt ra các vấn đề cơ bản sau:

* Quản lý các chuyến tham quan doanh nghiệp: đúng thông tin các chuyến tham quan (như thời gian, số lượng sinh viên tham gia,..), không sai sót trong quá trình sắp xếp giáo viên đại diện tham quan doanh nghiệp(VD: 1 giáo viên đại diện tham gia nhiều chuyến tham quan cùng lúc…)
* Quản lý giáo viên: giáo viên đại diện đúng với chuyến tham quan của mình.
* Quản lý sinh viên: sinh viên tại một thời điểm chỉ tham gia một chuyến tham quan và phải tham gia đúng với chuyến tham quan mà mình đã đăng ký.

2.2.2. Tài liệu đặc tả yêu cầu

* **Mô tả hệ thống:**
* Hệ thống quản lý đăng ký tham quan doanh nghiệp của sinh viên khoa Công Nghệ Thông Tin, trường Đại học Công nghiệp Hà Nội cho phép nhiều Actor có các chức năng khác nhau: Actor Quản trị viên toàn hệ thống có thể có toàn bộ các chức năng của Actor quản trị viên thông thường, ngoài ra có thể quản lý tài khoản các Actor khác. Actor Giáo viên có thể nhận, hủy , xem thông tin các đoàn tham quan của bản thân giáo viên, Actor có thể xem danh sách các sinh viên tham gia chuyến tham quan. Actor Sinh viên có quyền đăng ký, hủy , xem danh sách các chuyến tham quan sắp diễn ra, xem danh sách các sinh viên cùng tham gia chuyến tham quan, cập nhật thông tin cá nhân.
* **Giao diện và chức năng của hệ thống:**
* Hệ thống gồm 16 giao diện chính:
  + Giao diện Đăng nhập
  + Giao diện Trang chủ (admin và user)
  + Giao diện Quản lý giáo viên
  + Giao diện xem danh sách chuyến tham quan của giáo viên
  + Giao diện Quản lý sinh viên
  + Giao diện Chuyến tham quan sinh viên tham gia
  + Giao diện Quản lý các chuyến tham quan doanh nghiệp
  + Giao diện danh sách sinh viên tham gia chuyến tham quan
  + Giao diện Quản lý doanh nghiệp liên kết
  + Giao diện các chuyến tham quan tới doanh nghiệp liên kết
  + Giao diện Quản lý lớp học
  + Giao diện danh sách sinh viên của lớp học
  + Giao diện tài khoản cá nhân
  + Quản lý tài khoản.
  + Giao diện đánh giá sinh viên sau mỗi chuyến tham quan
* Chức năng:

Chức năng chung :

Ứng dụng cho phép sinh viên và giáo viên tham gia đăng ký tham gia chuyến du lịch sắp diễn ra. Đồng thời Admin của ứng dụng có vai trò quản lý, cập nhật thông tin các tài khoản, dữ liệu có thể được thống kê và xuất file PDF, EXCEL.

* + Giao diện trang chủ: cho phép lựa chọn các mục quản lý, đồng thời hiển thị số lượng doanh nghiệp đang liên kết với nhà trường, số lượng giáo viên, sinh viên của khoa, và số lượng chuyến tham quan đã, đang và sắp được tổ chức, ngoài ra thì ở giao diện trang chủ, quản trị có thể biết được danh sách các chuyến tham quan được tổ chức trong ngày .

Đối với Actor Quản trị viên :

* + Quản lý lớp học: hiển thị danh sách các lớp học của sinh viên khoa Công nghệ thông tin, trường Đại học Công Nghiệp Hà Nội, cho phép quản trị thêm, sửa, xóa thông tin lớp học và xem danh sách sinh viên của lớp học và tại màn xem danh sách sinh viên của lớp học thì người quản trị có thể thêm sinh viên vào cho lớp học đó .
  + Quản lý các chuyến tham quan du lịch : cho phép quản trị xem các chuyến tham quan doanh nghiệp, thêm, sửa, xóa các chuyến tham quan, và đồng thời cũng cho người quản trị xem danh sách sinh viên tham gia chuyến tham quan của một chuyến tham quan bất kỳ, và khi ở màn xem danh sách sinh viên của một chuyến tham quan, tìm kiếm, thêm, hoặc xóa sinh viên ra khỏi danh sách.
  + Quản lý doanh nghiệp: cho phép người quản trị xem danh sách các doanh nghiệp đang liên kết với khoa và nhà trường, quản trị có quyền thêm, sửa, xóa thông tin của doanh nghiệp, đồng thời người quản trị có thể xem và xuất pdf hoặc excel danh sách các chuyến tham quan doanh nghiệp mà doanh nghiệp đó tổ chức.
  + Quản lý giáo viên: cho phép hiển thị danh sách giáo viên của khoa, quản trị có thể thêm, sửa, xóa thông tin giáo viên. Quản trị cũng được cấp quyền thêm, xóa, xem và xuất danh sách các chuyến tham quan của giáo viên dưới định dạng PDF hoặc Excel. Trong màn xem danh sách các chuyến tham quan mà giáo viên đã đăng ký, người quản trị có thể xem danh sách sinh viên tham gia từng chuyến tham quan cụ thể. Ngoài ra, quản trị có khả năng tìm kiếm sinh viên và xuất danh sách sinh viên đó dưới dạng PDF hoặc Excel.
  + Quản lý tài khoản: cho phép người quản trị chính xem danh sách tài khoản của các người quản trị và cho phép, thêm, sửa xóa thông tin của các người quản trị khác. Có hai loại quản trị là quản trị có toàn quyền quản lý hệ thống và quản trị thông thường (không có quyền thêm, sửa, xóa thông tin của các quản trị khác)
  + Quản lý sinh viên: cho phép người quản trị xem danh sách sinh viên của khoa. Quản trị có quyền thêm, sửa, xóa thông tin, sinh viên và đồng thời có thể xem danh sách những chuyến tham quan mà một sinh viên bất kỳ tham gia.
  + Xuất file PDF, EXCEL: người quản trị có quyền xuất thông tin các danh sách mà hệ thống có : Chuyến tham quan, danh sách Giáo viên, sinh viên, lớp học, doanh nghiệp dưới dạng pdf hoặc excel.

Đối với Actor Giáo viên :

* + Xem thông tin các chuyến tham quan: Giáo viên có quyền xem danh sách chuyến tham quan hôm nay hoặc chuyến tham quan sắp diễn ra. Đăng ký, hủy đăng ký dẫn đoàn chuyến tham quan: Các chuyến thăm diễn ra sắp tới Giáo viên có quyền đăng ký hoặc hủy đăng ký chúng. Cập nhật thông tin cá nhân: Đối với Actor là Giáo viên sẽ có quyền sửa thông tin cá nhân. Xem danh sách sinh viên: Xem danh sách sinh viên tham gia tham quan doanh nghiệp đã đăng ký.

Đối với Actor Sinh viên :

* + Xem thông tin các chuyến tham quan: Sinh viên có quyền xem danh sách chuyến tham quan hôm nay hoặc , chuyến tham quan sắp diễn ra. Đăng ký, hủy đăng ký tham gia chuyến tham quan: Các chuyến thăm diễn ra sắp tới sinh viên có quyền đăng ký hoặc hủy đăng ký chúng. Xem danh sách sinh viên: Xem danh sách sinh viên tham gia tham quan doanh nghiệp đã đăng ký. Cập nhật thông tin: Cập nhật thông tin cá nhân các thông tin đã cung cấp từ trước.

2.3. Thực hiện bài toán

2.3.1. Võ Thị Quỳnh – Giao diện quản lý lớp học

* Quản lý lớp học có giao diện là màn hình quản lý lớp học, trong đó dữ liệu được lưu trong cơ sở dữ liệu, ClassroomDao được tạo để lấy dữ liệu từ cơ sở dữ liệu, ClassroomService được tạo để thực hiện câu lệnh với dữ liệu ở dao. Ở tầng giao diện gọi tới đối tượng ClassroomService.

**a. Giao diện**

A screenshot of a computer

Description automatically generated

Hình 1: Giao diện quản lý thông tin lớp học

A screenshot of a computer

Description automatically generated

Hình 2: Giao diện danh sách sinh viên của lớp học

* Giao diện quản lý thông tin lớp học được thực hiện bằng cơ chế kéo thả các labels: Tên lớp học , mã lớp học, tên Trang. Sử dụng 1 table để hiển thị danh sách giám thị. Các nút: Danh sách sinh viên , thêm, sửa , xóa, nhập lại để thực hiện quản lý thông tin lớp học . Nút “Quay lại trang chủ” để trở về trang chủ.

**b. Kiểu đối tượng**

* Các yếu tố hướng đối tượng:
* Lớp (class) Classroom:
  + Đây là một đơn vị cơ bản của hướng đối tượng. Nó đóng gói dữ liệu (thông tin về lớp học) và các phương thức liên quan vào một đối tượng.
* Thuộc tính (private int id, private String code, private String name, private List<Student> students):
  + Các thuộc tính được khai báo là private, chỉ có thể truy cập trực tiếp từ bên trong lớp Classroom. Điều này thể hiện tính đóng gói (encapsulation).
* Phương thức khởi tạo (public Classroom (), public Classroom (int id, String code, String name), ...):
  + Cung cấp cách để tạo ra và khởi tạo đối tượng Classroom. Đây là một ví dụ của tính đa hình (polymorphism) với các phương thức khác nhau dựa trên số lượng và loại tham số.
* Phương thức getter và setter (public int getId(), public void setId(int id), ...):
  + Cung cấp cách truy cập và thay đổi giá trị của các thuộc tính. Điều này là một triển khai của tính đóng gói.
* Biến tĩnh (public static int count = 0):
  + Biến tĩnh được sử dụng để theo dõi số lượng đối tượng Classroom đã được tạo. Điều này có thể được xem xét là một ví dụ của tính chia sẻ (shared state) giữa tất cả các đối tượng của lớp.
* Phương thức toString() (@Override public String toString()):
  + Override phương thức toString để cung cấp một biểu diễn chuỗi của đối tượng. Điều này thường được sử dụng để in ra thông tin của đối tượng khi cần thiết.
* Các lớp ClassroomDao, ClassroomService, ManageClassroom chứa các phương thức tương tác với các đối tượng của lớp Classroom để thực hiện các chức năng của chương trình.

**c. Bắt lỗi**

* Bắt lỗi khi chuyển trang:

A screenshot of a computer screen

Description automatically generated

* Bắt lỗi khi không thể cập nhật dữ liệu lớp học:

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

**d. Tập hợp**

Hệ thống sử dụng ArrayList, một cấu trúc dữ liệu được sử dụng để lưu trữ và quản lý một tập hợp các phần tử động. Cho phép thêm dữ liệu vào cơ sở dữ liệu, truy cập dữ liệu, …

A computer screen shot of a program code

Description automatically generated

2.3.2. Nguyễn Tài Tuệ – Giao diện quản lý thông tin giáo viên, giao diện quản lý thông tin sinh viên, giao diện trang chủ (với tài khoản quản trị viên), giao diện đăng nhập hệ thống, giao diện quản lý tham quan doanh nghiệp.

2.3.2.1. Giao diện quản lý thông tin giáo viên.

* Quản lý thông tin giáo viên có giao diện là màn hình quản lý lớp học, trong đó dữ liệu được lưu trong cơ sở dữ liệu, TeacherDao được tạo để lấy dữ liệu từ cơ sở dữ liệu, TeacherService được tạo để thực hiện câu lệnh với dữ liệu ở dao. Ở tầng giao diện gọi tới đối tượng TeacherService.

**a. Giao diện**

A screenshot of a computer

Description automatically generated

Hình 3: Giao diện quản lý thông tin giáo viên

A screenshot of a computer

Description automatically generated

Hình 4: Giao diện danh sách sinh viên của lớp học

* Giao diện được xây dựng bằng cách kéo thả các labels, swing control: Mã giáo viên, họ tên, địa chỉ, số điện thoại, ngày sinh, email, ảnh đại diện của giáo viên. 1 table dùng để hiển thị thông tin của danh sách các giáo viên. Chức năng của các nút trong giao diện. ngoài ra nó còn chứa các logic mà có liên quan đến giao diện và hành động của người quản trị trong chương trình. Các nút: “Xem danh sách chuyến tham quan của giáo viên”, “Thêm”, “Sửa” , “Xóa”, “Nhập lại” để thực hiện quản lý thông tin giáo viên . Nút “Quay lại” để trở về trang chủ.

**b. Kiểu đối tượng .**

* Các yếu tố hướng đối tượng:
* Lớp (class) Person:
  + Đây là một đơn vị cơ bản của hướng đối tượng. Nó đóng gói dữ liệu (thông tin về con người) và các phương thức liên quan vào một đối tượng.
* Thuộc tính (public static int count = 0, protected int id, protected String code, protected String firstName, protected String lastName, protected String address, protected String phoneNumber, protected String email, protected String birthDate;):
  + Các thuộc tính được khai báo là protected, chỉ có thể truy cập trực tiếp từ bên trong lớp Person. Điều này thể hiện tính đóng gói.
* Lớp (class) Teacher:
  + Đây là một đơn vị cơ bản của hướng đối tượng. Nó đóng gói dữ liệu (thông tin về giáo viên) được kế thừa từ class person và các phương thức liên quan vào một đối tượng.
* Thuộc tính (private List<Tour> tours = new ArrayList<>(), private String imagePath, private int accountId;):
  + Các thuộc tính được khai báo là private, chỉ có thể truy cập trực tiếp từ bên trong lớp Teacher. Điều này thể hiện tính đóng gói.
* Phương thức khởi tạo (public Teacher(), public Teacher( int id,String imagePath, String code, String firstName, String lastName, String address, String phoneNumber, String email, String birthDate, int accountID), ...):
  + Cung cấp cách để tạo ra và khởi tạo đối tượng Teacher. Đây là một ví dụ của tính đa hình với các phương thức khác nhau dựa trên số lượng và loại tham số.
* Phương thức getter và setter (public String getImagePath(), public String setImagePath (int id), ...):
  + Cung cấp cách truy cập và thay đổi giá trị của các thuộc tính. Điều này là một triển khai của tính đóng gói.
* Biến tĩnh (public static int count = 0):
  + Biến tĩnh được sử dụng để theo dõi số lượng đối tượng Teacher đã được tạo. Điều này có thể được xem xét là một ví dụ của tính chia sẻ (shared state) giữa tất cả các đối tượng của lớp.
* Phương thức toString() (@Override public String toString()):
  + Override phương thức toString để cung cấp một biểu diễn chuỗi của đối tượng. Điều này thường được sử dụng để in ra thông tin của đối tượng khi cần thiết.
* Các lớp TeacherDao, TeacherService, ManageTeacher chứa các phương thức tương tác với các đối tượng của lớp Teacher để thực hiện các chức năng của chương trình.

**c. Bắt lỗi**

* Bắt lỗi khi chuyển trang: A screenshot of a computer

  Description automatically generated
* Bắt lỗi khi không thể cập nhật dữ liệu giáo viên:

A screen shot of a computer

Description automatically generated

A screen shot of a computer code

Description automatically generated

**d. Tập hợp**

Hệ thống sử dụng ArrayList, một cấu trúc dữ liệu được sử dụng để lưu trữ và quản lý một tập hợp các phần tử động. Cho phép thêm dữ liệu vào cơ sở dữ liệu, truy cập dữ liệu, …

A screenshot of a computer code

Description automatically generatedA computer screen shot of a program

Description automatically generated

2.4.2.2. Giao diện quản lý thông tin sinh viên.

* Để chuyển đến phần giao diện quản lý thông sinh viên trong hệ thống, trước hết tại màn hình trang chủ (xuất hiện khi đăng nhập với tài khoản có quyền quản trị viên), chọn vào nút “Quản lý thông tin sinh viên”
* Phần chương trình cho giao diện quản lý thông tin sinh viên được chia thành nhiều file và được đặt trong các package khác nhau:
* StudentDao.java tại package dao: Chứa các phương thức đọc ghi dữ liệu tương tác trực tiếp với cơ sở dữ liệu của hệ thống.
* StudentService.java tại package services: Chứa các phương thức xử lý các logic của hệ thống mà độc lập với giao diện, tương tác với các phương thức đọc ghi dữ liệu tại StudentDao.java.
* ManageStudent.java tại package views: Là file chứa các thành phần GUI của màn hình quản lý tài khoản hệ thống, ngoài ra chứa các logic mà có liên quan đến giao diện và hành động của người dùng trong chương trình, tương tác với các phương thức được định nghĩa tại StudentService.java để xử lí các logic phức tạp hơn và độc lập với giao diện.

**a, Giao diện**

A screenshot of a computer

Description automatically generated

Hình 5: Giao diện quản lý thông tin sinh viên.

A screenshot of a computer

Description automatically generated

Hình 6: Giao diện quản lý chuyến tham quan của sinh viên.

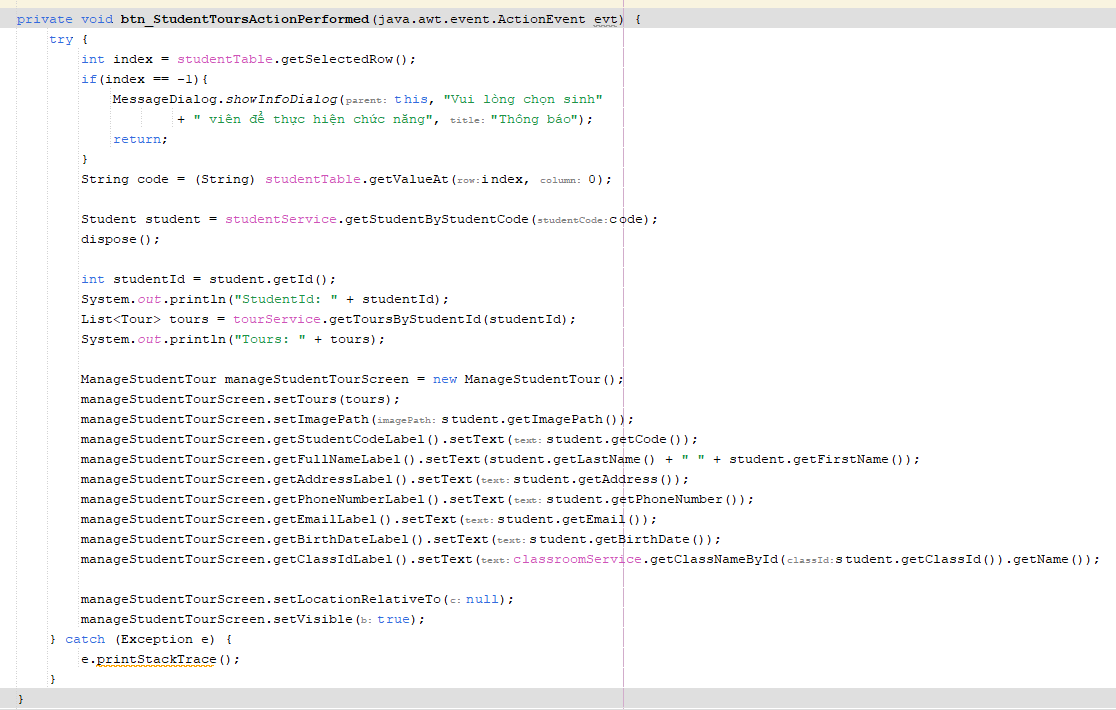
* Giao diện quản lý thông tin sinh viên được thực hiện bằng cơ chế kéo thả các label, swing control. Sử dụng 1 table để hiển thị danh sách sinh viên. Chức năng của các nút trong giao diện. ngoài ra nó còn chứa các logic mà có liên quan đến giao diện và hành động của người quản trị trong chương trình. Các nút: “Chuyến tham quan sinh viên tham gia”, “Thêm”, “Sửa” , “Xóa”, “Nhập lại” để thực hiện quản lý thông tin sinh viên . Nút “Quay lại trang chủ” để trở về trang chủ .

**b. Hướng đối tượng**

* Các yếu tố hướng đối tượng:
* Lớp (class) Person:
  + Đây là một đơn vị cơ bản của hướng đối tượng. Nó đóng gói dữ liệu (thông tin về con người) và các phương thức liên quan vào một đối tượng.
  + Thuộc tính (public static int count = 0, protected int id, protected String code, protected String firstName, protected String lastName, protected String address, protected String phoneNumber, protected String email, protected String birthDate;):
  + Các thuộc tính được khai báo là protected, chỉ có thể truy cập trực tiếp từ bên trong lớp Person. Điều này thể hiện tính đóng gói.
* Lớp (class) Student:
  + Đây là một đơn vị cơ bản của hướng đối tượng. Nó đóng gói dữ liệu (thông tin về sinh viên) được kế thừa từ class person và các phương thức liên quan vào một đối tượng.
  + Thuộc tính (private String imagePath, private List<StudentTour> studentTours, private int classId, private int accountId;):
  + Các thuộc tính được khai báo là private, chỉ có thể truy cập trực tiếp từ bên trong lớp Student. Điều này thể hiện tính đóng gói.
* Phương thức khởi tạo (public Student(), public Student(int id), ...):
  + Cung cấp cách để tạo ra và khởi tạo đối tượng Student. Đây là một ví dụ của tính đa hình với các phương thức khác nhau dựa trên số lượng và loại tham số.
* Phương thức getter và setter (public int getAccountId (), public void setAccountId (int id), ...):
  + Cung cấp cách truy cập và thay đổi giá trị của các thuộc tính. Điều này là một triển khai của tính đóng gói.
* Biến tĩnh (public static int count = 0):
  + Biến tĩnh được sử dụng để theo dõi số lượng đối tượng Sudent đã được tạo. Điều này có thể được xem xét là một ví dụ của tính chia sẻ (shared state) giữa tất cả các đối tượng của lớp.
* Phương thức toString() (@Override public String toString()):
  + Override phương thức toString để cung cấp một biểu diễn chuỗi của đối tượng. Điều này thường được sử dụng để in ra thông tin của đối tượng khi cần thiết.
* Các lớp StudentDao, StudentService, ManageStudent chứa các phương thức tương tác với các đối tượng của lớp Student để thực hiện các chức năng của chương trình.

**c. Bắt lỗi**

* Bắt lỗi khi chuyển trang .



* Bắt lỗi khi không thể cập nhật dữ liệu sinh viên.

A screenshot of a computer program

Description automatically generated

**d. Tập hợp**

* Phần quản lý thông tin sinh viên sử dụng ArrayList, một cấu trúc dữ liệu được sử dụng để lưu trữ và quản lý một tập hợp các phần tử động. Cho phép thêm phần tử vào cuối danh sách, truy cập vào phần tử tại một vị trí bất kì… Tập hợp này được sử dụng ở tất các các file liên quan đến phần quản lý này.



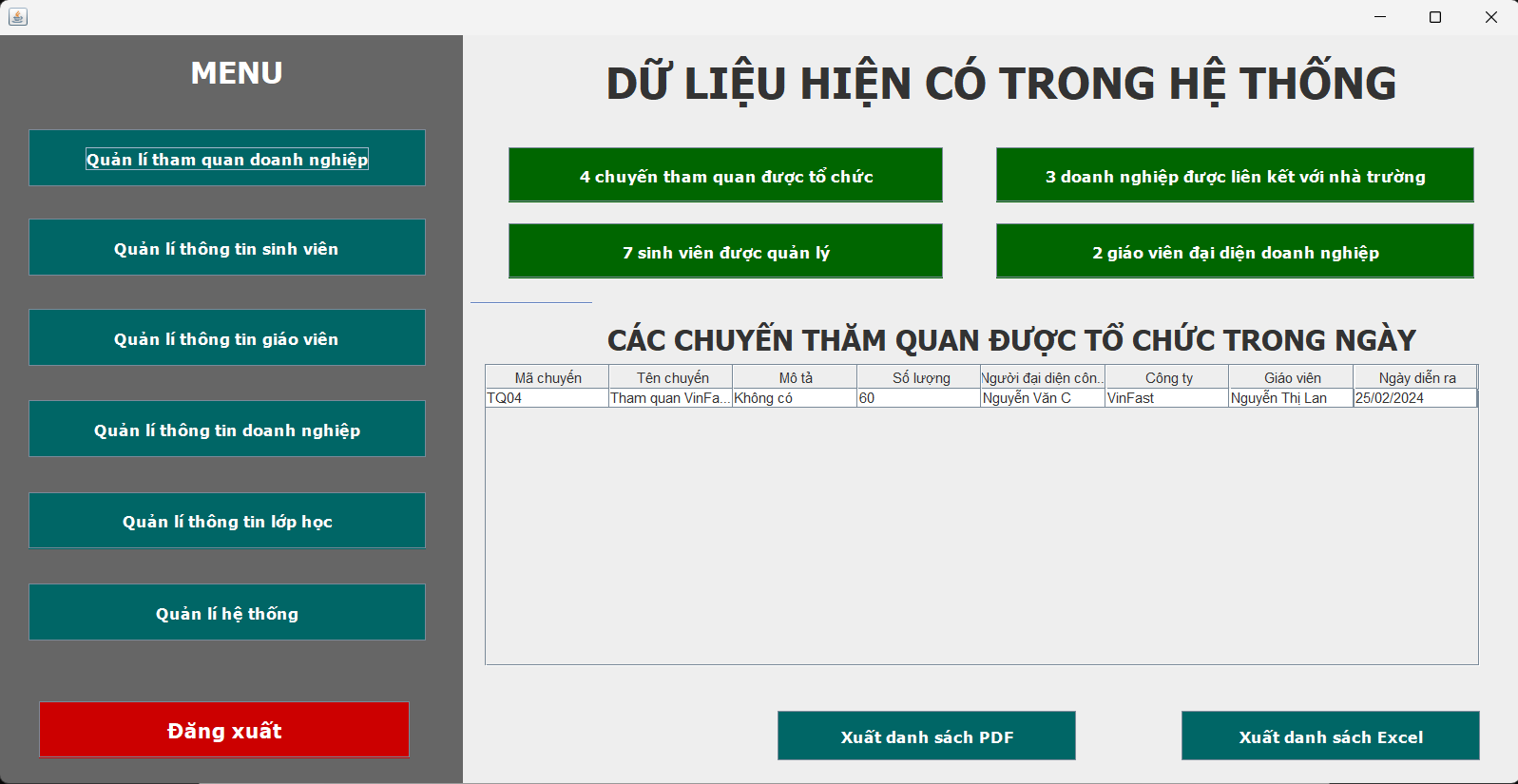
A screen shot of a computer code

Description automatically generated

2.4.2.3. Giao diện trang chủ (phía quản trị viên).

* Phần chương trình cho màn hìnhtrang chủ được chia thành nhiều file và được đặt trong các package khác nhau, các nút trên màn hình liên kết tới các giao diện khác của chương trình.
* AdminHome.java tại package views: Là file chứa các thành phần GUI của màn hình quản lý tài khoản hệ thống, ngoài ra chứa các logic mà có liên quan đến giao diện và hành động của người dùng trong chương trình, tương tác với các phương thức được định nghĩa tại AccountService.java để xử lí các logic phức tạp hơn và độc lập với giao diện.

**a. Giao diện**



Hình 7: Giao diện trang chủ cho quản trị viên

* Giao diện trang chủ được thực hiện bằng cơ chế kéo thả các label, swing control. Sử dụng 1 table để hiển thị danh sách sinh viên.
* Các nút “Quản lý chuyến tham quan doanh nghiệp”, “Quản lý thông tin sinh viên”, “Quản lý thông tin doanh nghiệp”, “Quản lí thông tin giáo viên”, “Quản lý thông tin lớp học”, “Quản lý tài khoản hệ thống”, khi quản trị viên ấn nút sẽ mở ra màn hình quản lý tương ứng với nút.
* Nút “Xuất danh sách PDF” dùng để xuất danh sách các chuyến tham quan được diễn ra trong ngày hiện tại (tính theo đồng hồ hệ thống của máy tính người dùng), file được xuất ra tại vị trí người dùng chọn khi sử dụng chức năng.
* Nút “Xuất danh sách Excel” dùng để xuất danh sách các chuyến tham quan được diễn ra trong ngày hiện tại (tính theo đồng hồ hệ thống của máy tính người dùng), file được xuất ra tại vị trí người dùng chọn khi sử dụng chức năng.
* Nút “Đăng xuất” để trở về trang đăng nhập với nhiều quyền khác nhau trong hệ thống.

**b. Bắt lỗi và gom rác**

* Bắt lỗi khi không hiển dữ liệu các chuyến tham quan diễn ra trong ngày vào bảng. A screenshot of a computer code

  Description automatically generated

A screen shot of a computer code

Description automatically generated

* Bắt lỗi khi không thực hiện được chức năng xuất file Excel.

A screenshot of a computer

Description automatically generated

A screenshot of a computer code

Description automatically generated

A screenshot of a computer program

Description automatically generated

* Bắt lỗi khi không thực hiện được chức năng xuất file PDF.

A screen shot of a computer code

Description automatically generated

* Bắt lỗi khi không hiển thị được dữ liệu hiện có trong hệ thống.

A screenshot of a computer code

Description automatically generated

2.4.2.4. Giao diện đăng nhập.

* + Phần chương trình cho màn hìnhtrang chủ được chia thành nhiều file và được đặt trong các package khác nhau:
* AccountDao.java tại package dao: Chứa các phương thức đọc ghi dữ liệu tương tác trực tiếp với cơ sở dữ liệu.
* AccountService.java tại package services: Chứa các phương thức xử lý các logic của hệ thống mà độc lập với giao diện, tương tác với các phương thức đọc ghi dữ liệu tại AccountDao.java.
* Login.java tại package views: Là file chứa các thành phần GUI của màn hình quản lý tài khoản hệ thống, ngoài ra chứa các logic mà có liên quan đến giao diện và hành động của người dùng trong chương trình, tương tác với các phương thức isExisted được định nghĩa tại AccountService.java để xử lí các logic phức tạp hơn và độc lập với giao diện.

**a. Giao diện.**

A screenshot of a computer screen

Description automatically generated

Hình 8: Giao diện đăng nhập

* Giao diện trang chủ được thực hiện bằng cơ chế kéo thả các label, swing control. Sử dụng 1 table để hiển thị danh sách.
* Nút “Đăng nhập” dùng để xác nhận tài khoản đăng nhập vào hệ thống. Khi đăng nhập thành công sẽ hiển thị giao diện phù hợp với quyền của tài khoản (Tài khoản sinh viên, Tài khoản giáo viên, Quản trị viên, …)

**b. Hướng đối tượng.**

* Các yếu tố hướng đối tượng:
* Lớp (class) Account:
  + Đây là một đơn vị cơ bản của hướng đối tượng. Nó đóng gói dữ liệu (thông tin về tài khoản) và các phương thức liên quan vào một đối tượng.
  + Thuộc tính (private int id, private String username, String password, private String role;):
  + Các thuộc tính được khai báo là private, chỉ có thể truy cập trực tiếp từ bên trong lớp Account. Điều này thể hiện tính đóng gói.
* Phương thức khởi tạo (public Account (), public Account (String username, String password), ...):
  + Cung cấp cách để tạo ra và khởi tạo đối tượng Acount. Đây là một ví dụ của tính đa hình với các phương thức khác nhau dựa trên số lượng và loại tham số.
* Phương thức getter và setter (public int getId (), public void setAId (int id), ...):
  + Cung cấp cách truy cập và thay đổi giá trị của các thuộc tính. Điều này là một triển khai của tính đóng gói.
* Biến tĩnh (public static int count = 0):
  + Biến tĩnh được sử dụng để theo dõi số lượng đối tượng Sudent đã được tạo. Điều này có thể được xem xét là một ví dụ của tính chia sẻ (shared state) giữa tất cả các đối tượng của lớp.
* Phương thức toString() (@Override public String toString()):
  + Override phương thức toString để cung cấp một biểu diễn chuỗi của đối tượng. Điều này thường được sử dụng để in ra thông tin của đối tượng khi cần thiết.
* Các lớp AccountDao, AccountService, ManageAccount chứa các phương thức tương tác với các đối tượng của lớp Account để thực hiện các chức năng của chương trình.

**c. Bắt lỗi và gom rác.**

* Bắt lỗi khi người dùng ấn nút “Đăng nhập”.

A screenshot of a computer

Description automatically generated

**A screenshot of a computer screen

Description automatically generated**

**d. Tập hợp.**

* Phần quản lý thông tin đăng nhập sử dụng ArrayList, một cấu trúc dữ liệu được sử dụng để lưu trữ và quản lý một tập hợp các phần tử động. Cho phép thêm phần tử vào cuối danh sách, truy cập vào phần tử tại một vị trí bất kì… .



2.4.2.5. Giao diện quản lý tham quan doanh nghiệp.

* + Phần chương trình cho màn hình quản lý tham quan doanh nghiệp được chia thành nhiều file và được đặt trong các package khác nhau:
* TourDao.java tại package dao: Chứa các phương thức đọc ghi dữ liệu tương tác trực tiếp với cơ sở dữ liệu.
* TourService.java tại package services: Chứa các phương thức xử lý các logic của hệ thống mà độc lập với giao diện, tương tác với các phương thức đọc ghi dữ liệu tại TourDao.java.
* ManageTour.java tại package views: Là file chứa các thành phần GUI của màn hình quản lý tham quan doanh nghiệp, ngoài ra chứa các logic mà có liên quan đến giao diện và hành động của người dùng trong chương trình, tương tác với các phương thức isExisted được định nghĩa tại TourService.java để xử lí các logic phức tạp hơn và độc lập với giao diện.

**a, Giao diện.**

A screenshot of a computer

Description automatically generated

Hình 9: Giao diện quản lý tham quan doanh nghiệp

A screenshot of a computer

Description automatically generated

Hình 10: Giao diện danh sách sinh viên tham quan doanh nghiệp

* Giao diện trang chủ được thực hiện bằng cơ chế kéo thả các label, swing control. Sử dụng 1 table để hiển thị danh sách.

**b. Hướng đối tượng**

* Các yếu tố hướng đối tượng:
* Lớp (class) Tour:
  + Đây là một đơn vị cơ bản của hướng đối tượng. Nó đóng gói dữ liệu (thông tin về tài khoản) và các phương thức liên quan vào một đối tượng.
  + Thuộc tính (private int id, private String code, private String name, private String description, private String startDate, private int availables, private List<StudentTour> studentTours, private int companyId, private int teacherId, private String presentator;):
  + Các thuộc tính được khai báo là private, chỉ có thể truy cập trực tiếp từ bên trong lớp Tour. Điều này thể hiện tính đóng gói.
* Phương thức khởi tạo (public Tour (), public Tour (String code, String name, String description, String startDate, int availables, int companyId, int teacherId, String presentator), ...):
  + Cung cấp cách để tạo ra và khởi tạo đối tượng Tour . Đây là một ví dụ của tính đa hình với các phương thức khác nhau dựa trên số lượng và loại tham số.
* Phương thức getter và setter (public int getId (), public void setAId (int id), ...):
  + Cung cấp cách truy cập và thay đổi giá trị của các thuộc tính. Điều này là một triển khai của tính đóng gói.
* Biến tĩnh (public static int count = 0):
  + Biến tĩnh được sử dụng để theo dõi số lượng đối tượng Tour đã được tạo. Điều này có thể được xem xét là một ví dụ của tính chia sẻ (shared state) giữa tất cả các đối tượng của lớp.
* Phương thức toString() (@Override public String toString()):
  + Override phương thức toString để cung cấp một biểu diễn chuỗi của đối tượng. Điều này thường được sử dụng để in ra thông tin của đối tượng khi cần thiết.
* Các lớp AccountDao, AccountService, ManageAccount chứa các phương thức tương tác với các đối tượng của lớp Account để thực hiện các chức năng của chương trình.

**c. Bắt lỗi và gom rác.**

* Bắt lỗi khi người dùng chuyển trang.

**A screenshot of a computer program

Description automatically generated**

**d. Tập hợp.**

* Phần quản lý thông tin đăng nhập sử dụng ArrayList, một cấu trúc dữ liệu được sử dụng để lưu trữ và quản lý một tập hợp các phần tử động. Cho phép thêm phần tử vào cuối danh sách, truy cập vào phần tử tại một vị trí bất kì… .



A screenshot of a computer program

Description automatically generated

2.3.3 Nguyễn Đông Bắc – Màn hình giao diện dành cho sinh viên.

**a. Giao diện:**

* Giao diện trang chủ, các chuyến tham quan sắp diễn ra, các chuyến tham quan hôm nay của bạn:

A screenshot of a computer

Description automatically generated

Hình 11: Giao diện trang chủ

A screenshot of a computer

Description automatically generated

Hình 12: Giao diện xem các chuyến tham quan hôm nay

*A screenshot of a computer

Description automatically generated*

Hình 13: Giao diện xem các chuyến tham quan sắp diễn ra

* Giao diện “Trang chủ”, “Các chuyến tham quan hôm nay của bạn”, “Các chuyến tham quan sắp diễn ra” được thực hiện bằng cơ chế kéo thả các labels:Tiêu đề. Sử dụng 1 table để hiển thị danh sách các chuyến tham quan sắp diễn ra. Các nút: “Các chuyến tham quan đã đăng ký”, “ Tài khoản cá nhân”, “Chuyến tham quan hôm nay của bạn” , “ Các chuyến tham quan sắp diễn ra” để thực hiện quản lý tài khoản sinh viên.
* Nút “Các chuyến tham quan đã đăng ký” để xem các chuyến tham quan đã đăng ký, nút “Tài khoản cá nhân” để chuyển tới giao diện quản lý các thông tin tài khoản bao gồm thông tin người dùng và thông tin tài khoản. Nút “Chuyến tham quan hôm nay của bạn” dùng để hiển thị các chuyến tham quan sẽ diễn ra hôm nay lên bảng. Nút “Tìm kiếm” dùng để tìm kiếm theo tên chuyến tham quan. Để đăng ký các chuyến tham quan sắp diễn ra, sinh viên chọn chuyến tham quan cần đăng ký và nhấn “Đăng ký tham gia chuyến tham quan”.
* **Giao diện tài khoản cá nhân:**

A screenshot of a computer

Description automatically generated

Hình 14: Giao diện quản lý tài khoản cá nhân

* Giao diện “Tài khoản cá nhân” được thực hiện bằng cơ chế kéo thả các labels:Chủ tài khoản, tên chủ tài khoản, tiêu đề thông tin cơ bản, các thông tin cơ bản gồm: Mã sinh viên, Họ và tên, Ngày sinh, Email, Số điện thoại, Lớp, Tài khoản và mật khẩu.
* Nút “Chỉnh sửa” dùng để chỉnh sửa thông tin cá nhân của sinh viên và nút đổi mật khẩu dùng để thay đổi mật khẩu. Sau khi click vào nút đổi mật khẩu sẽ hiển thị một màn hình cho người dùng nhập mật khẩu cũ và mật khẩu mới, để hoàn tất quá trình thay đồi mật khẩu, click vào nút “Lưu thay đổi” hoặc click vào nút “Hủy bỏ” để hủy bỏ quá trình này.
* **Giao diện “Các chuyến tham quan đã đăng ký”:**

**A screenshot of a computer

Description automatically generated**

Hình 15: Giao diện các chuyến tham quan đã đăng ký

* Giao diện “Các chuyến tham quan đã đăng ký” bao gồm tiêu đề, một bảng hiển thị các chuyến tham quan đã đăng ký tham gia. Nút “Tìm kiếm” để tìm kiếm chuyến tham quan theo tên. Để hủy đăng ký, cần nhấn vào nút “Hủy chuyến tham quan”. Ngoài ra, còn có thể xuất danh sách Pdf hoặc exel các chuyến tham quan đã đăng ký bằng các nút tương ứng.

**b. Bắt lỗi:**

* Bắt lỗi không lấy được dữ liệu từ file:

A screen shot of a computer code

Description automatically generated

2.3.4 Nguyễn Trọng Tuấn- Màn hình giao diện dành cho giáo viên.

**a. Giao diện.**

~~A screenshot of a computer

Description automatically generated~~

Hình 16: Giao diện trang chủ của giáo viên

* Cũng giống như giao diện trang chủ của sinh viên, giao diện trang chủ của giáo viên, cho phép giáo viên xem các chuyến tham quan mà giáo viên đã đăng ký, sửa thông tin giáo viên, xem chuyến tham quan ngày hôm nay của giáo viên(nếu có) và xem các chuyến tham quan sắp diễn ra và cuối cùng là đánh giá sinh viên sau khi tham gia chuyến tham quan về. Ngoài ra ở giao diện này thì giáo viên còn có thể đăng ký tham gia một chuyến tham quan và hủy không tham gia nữa và xem danh sách sinh viên của một chuyến tham quan.

A screenshot of a computer

Description automatically generated

Hình 17: Giao diện Các chuyến tham quan đã đăng ký

* Giao diện khi nhấn vào nút “Tài khoản cá nhân”

A screenshot of a computer

Description automatically generated

Hình 18: Giao diện Tài khoản cá nhân.

* Giao diện khi nhấn vào nút “Chuyến tham quan hôm nay của bạn”

A screenshot of a computer

Description automatically generated

Hình 19: Giao diện chuyến tham quan hôm nay của bạn*.*

* Giao diện khi nhấn vào nút “Các chuyến tham quan sắp diễn ra” giống giao diện trang chủ khi mới truy cập.

**b. Bắt lỗi và gom rác.**

* Bắt lỗi chuyển trang khi giáo viên muốn xem hoặc thay đổi thông tin của mình.

A screenshot of a computer program

Description automatically generated

* Ngoài ra còn nhiều loại bắt lỗi khác như bắt lỗi khi chưa chọn một chuyến tham quan mà đã nhấn nút “Xem danh sách sinh viên”, bắt lỗi khi một không thể thêm một chuyến tham quan do chuyến tham quan có số lượng người tham gia đã đầy, hoặc chuyến tham quan được chọn để đăng ký có thời gian trùng với thời gian của một chuyến tham quan đã đăng ký…

2.3.5 Võ Thị Quỳnh, Vũ Thị Thanh Tâm – Giao diện quản lý hệ thống.

* + Phần chương trình cho màn hìnhtrang chủ được chia thành nhiều file và được đặt trong các package khác nhau:
* AccountDao.java tại package dao: Chứa các phương thức đọc ghi dữ liệu tương tác trực tiếp với cơ sở dữ liệu.
* AccountService.java tại package services: Chứa các phương thức xử lý các logic của hệ thống mà độc lập với giao diện, tương tác với các phương thức đọc ghi dữ liệu tại AccountDao.java.
* Login.java tại package views: Là file chứa các thành phần GUI của màn hình quản lý tài khoản hệ thống, ngoài ra chứa các logic mà có liên quan đến giao diện và hành động của người dùng trong chương trình, tương tác với các phương thức isExisted được định nghĩa tại AccountService.java để xử lí các logic phức tạp hơn và độc lập với giao diện.
  1. Giao diện.

A screenshot of a computer

Description automatically generated

Hình 20: Giao diện quản lý thông tin hệ thống.

* Giao diện trang chủ được thực hiện bằng cơ chế kéo thả các label, swing control. Sử dụng 1 table để hiển thị danh sách.

**b. Hướng đối tượng.**

* Giống quản lý đăng nhập.

**c. Bắt lỗi và gom rác.**

* Ta sử dụng try – catch để xử lý ngoại lệ. Nếu ngoại lệ xảy ra, hệ thống sẽ hiển thị thông báo lên màn hình. Ngược lại nếu không có ngoại lệ, hệ thống sẽ hoạt động bình thường. Toàn bộ các lỗi được bắt tại file AdminHome.java, vì lớp Login chứa các GUI cũng như là tầng gần với thao tác của người dùng nhất, bắt lỗi tại đây sẽ trả được về các lỗi sát với tương tác của người dùng nhất, lỗi được hiển thị sẽ rõ ràng hơn. Các tầng thấp hơn như DAO, service khi có lỗi sẽ được đưa lên tầng cao nhất này (AdminHome.java) qua cách đáp lỗi qua phương thức (throws Exception)
* Bắt lỗi khi người dùng ấn nút “Đăng nhập”.

A screenshot of a computer

Description automatically generated**A screenshot of a computer screen

Description automatically generated**

**d. Tập hợp.**

* Giống quản lý đăng nhập .

PHẦN 3: KẾT LUẬN VÀ BÀI HỌC KINH NGHIỆM

3.1. Nội dung đã thực hiện

1. Áp dụng những kiến thức đã được cung cấp trên trường như cấu trúc của một chương trình java, các kiểu dữ liệu và toán tử, các kĩ thuật xử lý, lập trình đối tượng trong java, các kĩ thuật với cấu trúc collection, giao diện java swing. Qua đó xây dựng được chương trình quản lý sinh viên tham gia doanh nghiệp.
2. Tìm hiểu và áp dụng các kỹ thuật mới như kết nối chương trình với cơ sở dữ liệu thông qua MySQL , xuất file pdf và excel để hoàn thiện các chức năng cho chương trình.

3.2 Hướng phát triển tương lai

Trong quá trình thực hiện đề tài xây dựng phần mềm giúp sinh viên đăng ký tham quan doanh nghiệp trường Đại học Công Nghiệp Hà Nội, chúng em nhận thấy sự quan trọng và hữu ích của việc áp dụng công nghệ thông tin trong việc đăng ký tham quan doanh nghiệp. Phần mềm giúp sinh viên đăng ký tham quan doanh nghiệp đã giúp nâng cao hiệu quả và khâu đăng ký đồng thời cung cấp thông tin toàn diện và chính xác để hỗ trợ quyết định và lập kế hoạch của sinh viên và giáo viên. Kinh nghiệm từ việc thực hiện đề tài này đã đem lại những bài học quý giá.

Bài học rút ra:

* Sự cần thiết của việc nắm vững yêu cầu: Trước khi bắt đầu xây dựng phần mềm, việc hiểu rõ yêu cầu từ phía trường Đại học Công Nghiệp Hà Nội và người dùng là rất quan trọng. Điều này giúp đảm bảo rằng sản phẩm cuối cùng đáp ứng đúng nhu cầu và mong đợi của người sử dụng.
* Quy trình phát triển phần mềm: Áp dụng quy trình phát triển phần mềm hợp lý, như mô hình thác nước hay agile, giúp quản lý dự án hiệu quả hơn. Phân chia công việc thành các giai đoạn, theo dõi tiến độ và tương tác thường xuyên với mọi người giúp đảm bảo chất lượng và sự ổn định của dự án.

Tổng kết lại, việc thực hiện đề tài xây dựng phần mềm giúp sinh viên đăng ký tham quan doanh nghiệp trường Đại học Công Nghiệp Hà Nội đã mang lại nhiều lợi ích và kinh nghiệm quý báu cho nhóm chúng em. Trong quá trình thực hiện đề tài, nhận thấy nhóm đã cố gắng hết sức nhưng vì kiến thức vẫn còn hạn hẹp nên vẫn còn nhiều thiếu sót, mong thầy cô bổ sung để dự án trên được hoàn thiện hơn.

TÀI LIỆU THAM KHẢO

[1]. Giáo trình Lập trình HĐT với Java, Nguyễn Bá Nghiễn, Ngô Văn Bình, [2]. Vương Quốc Dũng, Đỗ Sinh Trường; NXB Thống kê, 2020.

[3]. Bộ slide bài giảng lập trình java- Bộ môn CMPM- trường ĐHCN HN

[4]. Lập trình hướng đối tượng với Java; Đoàn Văn Ban; NXB Khoa học và Kỹ thuật, Hà Nội 2006 (Tái bản).

[5]. Lập trình Java nâng cao, Đoàn Văn Ban, NXB Khoa học và Kỹ thuật, Hà Nội 2006

[6]. The Java Programming Language; Author: K. Arnold, J. Gosling; Published: Addison-Wesley, 1996, ISBN 0-201-63455-4

# 