

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO THÍ NGHIỆM/ THỰC NGHIỆM:
LẬP TRÌNH JAVA

XÂY DỰNG PHẦN MỀM QUẢN LÝ TRÔNG XE ĐẠI HỌC CÔNG
NGHIỆP HÀ NỘI

GVHD: Ths. Vũ Thị Dương
Thành viên: Phạm Minh Hoàng
Trần Thị Hồng Thắm
Dương Văn Thắng
Lê Bá Thành
Vũ Đức Triệu

Nhóm: 03
Lớp: 20223IT6019001. **Khóa:** 15

Hà Nội, năm 2023

MỤC LỤC

PHẦN 1: MỞ ĐẦU.....	4
1.1. Mô tả tổng quát chủ đề nghiên cứu.....	4
1.1.1. Tóm lược ý tưởng.....	4
1.1.2. Chủ đề nghiên cứu	4
1.1.3. Nội dung học tập	4
1.1.4. Các kiến thức, kỹ năng để giải quyết bài toán	4
1.2. Mục tiêu và chuẩn đầu ra	4
PHẦN 2: KẾT QUẢ NGHIÊN CỨU	6
2.1. Giới thiệu chung.....	6
2.1.1. Quy trình xây dựng sản phẩm.....	6
2.1.2. Hình thức sản phẩm	6
2.2. Khảo sát hệ thống.....	7
2.2.1. Khảo sát sơ bộ.....	7
2.2.1.1. Phương pháp phỏng vấn.....	7
2.2.1.2. Phương pháp lập phiếu điều tra	10
2.2.1.3. Đối tượng khảo sát	11
2.2.2. Mô tả bài toán.....	11
2.3. Phân tích hệ thống.....	13
2.3.1. Mô hình hóa chức năng hệ thống.....	13
2.3.2. Mô hình hóa dữ liệu và giao diện hệ thống	17
2.3.2.1. UseCase Đăng nhập	17
2.3.2.2. UseCase Quản lý ca trực	18

2.3.2.3. UseCase Quản lý vé xe	19
2.3.2.4. UseCase Quản lý trông xe và Thanh toán.....	20
2.3.2.5. UseCase Thống kê	21
2.4. Thực hiện bài toán.....	22
2.4.1. Vũ Đức Triệu – Đăng nhập.....	22
2.4.2. Trần Thị Hồng Thắm – Quản lý ca trực	22
2.4.3. Dương Văn Thắng – Quản lý vé xe	27
2.4.4. Phạm Minh Hoàng – Thống kê.....	30
2.4.5. Lê Bá Thành – Quản lý trông xe.....	33
2.4.6. Vũ Đức Triệu – Thanh toán	36
PHẦN 3: KẾT LUẬN VÀ BÀI HỌC KINH NGHIỆM	38
3.1. Nội dung đã thực hiện	38
3.2. Hướng phát triển	39
TÀI LIỆU THAM KHẢO.....	40

PHẦN 1: MỞ ĐẦU

1.1. Mô tả tổng quát chủ đề nghiên cứu

1.1.1. Tóm lược ý tưởng

Đây là chương trình hỗ trợ trong hoạt động quản lý trông xe của trường Đại học Công Nghiệp Hà Nội.

Chương trình giúp lên lịch phân chia ca trực, quản lý vé xe, thống kê doanh thu theo ngày một cách rõ ràng và thuận tiện.

1.1.2. Chủ đề nghiên cứu

Tên chủ đề: Xây dựng phần mềm quản lý trông xe Đại học Công Nghiệp Hà Nội.

1.1.3. Nội dung học tập

Nội dung học tập cần có để hoàn thành sản phẩm bao gồm:

- Các kiến thức đã học về kỹ thuật lập trình, lập trình hướng đối tượng, trang bị thêm kiến thức về lập trình java cơ sở và xử lý cơ bản giao diện.
- Biết đặt vấn đề, phân tích và thiết kế hệ thống quản lý trông xe cơ bản, có kỹ năng làm việc nhóm và giải quyết vấn đề.

1.1.4. Các kiến thức, kỹ năng để giải quyết bài toán

➤ Các kiến thức cần có

- Có kiến thức về lập trình Java cơ sở.
- Có kiến thức về lập trình Java hướng đối tượng.
- Xử lý ngoại lệ và thao tác với tệp.
- Xử lý giao diện cơ bản.

➤ Các kỹ năng cần có

- Có kỹ năng làm việc nhóm, biết đặt vấn đề và giải quyết vấn đề.
- Có kỹ năng khảo sát sơ bộ để thu thập yêu cầu hệ thống.
- Có kỹ năng phân tích yêu cầu để tìm ra các chức năng của hệ thống.

1.2. Mục tiêu và chuẩn đầu ra

Các mục tiêu và chuẩn đầu ra cần đạt được:

- Phát biểu được bài toán cần xây dựng, phân tích và đưa ra sơ đồ lớp mô tả bài toán theo hướng đối tượng.
- Vận dụng các kỹ thuật đã học trong lập trình Java cài đặt được bài toán theo chủ đề được giao.
- Sản phẩm nghiên cứu: Quyền báo cáo thí nghiệm/ thực nghiệm về chủ đề được giao và sản phẩm mã nguồn.

PHẦN 2: KẾT QUẢ NGHIÊN CỨU

2.1. Giới thiệu chung

Tên chủ đề: Xây dựng hệ thống quản lý trông xe của trường Đại học Công nghiệp Hà Nội.

2.1.1. Quy trình xây dựng sản phẩm

Sử dụng mô hình quy trình thác nước (Waterfall model)

Mô hình thác nước (Waterfall model) là mô hình quy trình phát triển phần mềm đầu tiên được giới thiệu. Trong mô hình thác nước, mỗi giai đoạn phải được hoàn thành trước khi giai đoạn tiếp theo có thể bắt đầu và không có sự chồng chéo trong các giai đoạn.

➤ Các bước tổng quan

- Thu thập và phân tích yêu cầu (Requirement Analysis): Tất cả các yêu cầu có thể có của hệ thống được phát triển đều được ghi lại trong giai đoạn này và được ghi lại trong tài liệu đặc tả yêu cầu để phục vụ cho các giai đoạn sau.
- Thiết kế hệ thống (System Design): Thiết kế hệ thống giúp xác định các yêu cầu phần cứng và hệ thống cũng như giúp xác định kiến trúc hệ thống tổng thể.
- Thực hiện (Implementation): Với đầu vào từ thiết kế hệ thống, tiến hành cài đặt chương trình.
- Tích hợp và Kiểm thử (Integration and Testing).
- Triển khai hệ thống (Deployment of system).
- Bảo trì (Maintenance).

2.1.2. Hình thức sản phẩm

Sản phẩm sau khi hoàn thành là một chương trình quản lý trông xe cơ bản, có thể áp dụng vào hệ thống trông xe tại các trường đại học, cụ thể ở đây là trường Đại học Công nghiệp Hà Nội.

Các bên tham gia sử dụng bao gồm: nhân viên nhà xe, quản lý nhà xe, học sinh, sinh viên trong trường, cán bộ, công nhân viên của trường Đại học Công nghiệp Hà Nội.

Phần mềm được viết bằng ngôn ngữ lập trình java, trên công cụ hỗ trợ viết mã Netbeans do nó dễ dàng sử dụng, dễ thiết kế giao diện và có nhiều tính năng hỗ trợ viết mã lệnh.

2.2. Khảo sát hệ thống

2.2.1. Khảo sát sơ bộ

2.2.1.1. Phương pháp phỏng vấn

➤ Kế hoạch phỏng vấn

Kế hoạch phỏng vấn	
Người được hỏi: Nguyễn Văn A	Người phỏng vấn: Lê Bá Thành, Vũ Đức Triệu
Địa chỉ: Đại học công nghiệp Hà Nội(298 Cầu Diễn, Bắc Từ Liêm, Hà Nội).	Thời gian hẹn: 8h sáng ngày 13/2/2023 Thời điểm bắt đầu: 8h sáng ngày 13/2/2023 Thời điểm kết thúc: 9h sáng ngày 13/2/2023
Đối tượng: <ul style="list-style-type: none"> • Nhân viên trông xe • Khảo sát kỹ lưỡng, không bỏ sót thông tin, Các thông tin thu thập được phải chính xác, không gây ác cảm với người được phỏng vấn 	Các yêu cầu đòi hỏi: <ul style="list-style-type: none"> • Vai trò: Cửa bộ phận trông xe trong nhà trường. • Vị trí: Nhân viên trông xe. • Trình độ: Tốt nghiệp trung học phổ thông.
Chương trình:	Ước lượng thời gian

<ul style="list-style-type: none"> • Giới thiệu: buổi phỏng vấn nhằm thu thập thông tin chi tiết để xây dựng hệ thống quản lí trông xe. 	1 phút
<ul style="list-style-type: none"> • Tổng quan về dự án: thu thập và phân tích các thông tin cần thiết để xây dựng hệ thống quản lí trông xe. 	2 phút
<ul style="list-style-type: none"> • Tổng quan về phỏng vấn: Nội dung các câu hỏi phỏng vấn sẽ xoay quanh các chi tiết và đặc điểm của đơn vị trông xe cũng như các yêu cầu đặt ra. 	1 phút
Xin phép được ghi âm cuộc phỏng vấn	10 phút
Chủ đề 1: Câu hỏi và trả lời	10 phút
Chủ đề 2: Câu hỏi và trả lời	2 phút
Tập hợp các nội dung chính	5 phút
Ý kiến người được hỏi	2 phút
Kết thúc (thỏa thuận)	
	Dự kiến: tổng cộng 33 phút

➤ Phiếu phỏng vấn

Phiếu phỏng vấn	
Tên dự án: Xây dựng hệ thống quản lí trông xe Đại học CNHN	
Tên tiểu dự án: Phân tích đặc tả hệ thống quản lí trông xe Đại học CNHN	
Người được hỏi: Nguyễn Văn A	Ngày: 13/02/2023

	Người hỏi: Vũ Đức Triệu, Lê Bá Thành
Câu hỏi	Ghi chú
Câu 1: Đơn vị quản lí nhà xe thuộc bộ phận nào trong nhà trường.	Bộ phận hỗ trợ sinh viên.
Câu 2: Khách hàng chủ yếu là những đối tượng nào?	Sinh viên, cán bộ nhà trường, giảng viên.
Câu 3: Anh có muốn thống kê số tiền thu vào hằng ngày của từng khu vực không?	Có.
Câu 4: Anh có muốn hệ thống có thể kiểm tra lịch sử ra vào xe không?	Có.
Câu 5: Anh có muốn thống kê về số lượng chi tiết xe vào ra trong ngày không?	Có.
Câu 6: Anh có muốn khách hàng gửi xe sử dụng vé theo thời gian dài hạn không?	Có.
Câu 7: Anh có muốn có loại vé giữ xe riêng cho cán bộ và giáo viên nhà trường hay không?	Có.

Câu 8: Anh có muốn hiển thị thông tin chi tiết về biển số xe khi khách hàng giữ và trả xe không?	Có.
<p>Đánh giá chung:</p> <ul style="list-style-type: none"> • Người được phỏng vấn có vẻ như không có thẩm quyền trả lời một số câu hỏi, họ cần thêm hai đến ba ngày bàn bạc với cấp trên rồi mới có thể tiến hành tiếp. • Chưa kết luận được vấn đề, còn một vài chủ đề chưa đề cập hết. 	

2.2.1.2. Phương pháp lập phiếu điều tra

➤ Phiếu điều tra

<p align="center">Phiếu điều tra về việc gửi xe của khách hàng khi gửi xe tại trường Đại học Công Nghiệp Hà Nội</p> <p>Bạn hãy khoanh tròn vào mục lựa chọn, hoặc bỏ phiếu vào hòm thư.</p> <p>Câu 1: Bạn đã từng gửi xe trong khu vực gửi xe của trường Đại học Công Nghiệp Hà Nội chưa?</p> <p>A. Tôi đã từng</p> <p>B. Tôi chưa từng</p> <p>Câu 2: Bạn cảm thấy dịch vụ gửi xe của trường Đại học Công Nghiệp Hà Nội như thế nào?</p> <p>A. Tốt</p> <p>B. Chưa Tốt</p> <p>Câu 3: Bạn có gặp khó khăn nào khi gửi xe trong trường Đại học Công Nghiệp Hà Nội chưa?</p> <p>A. Có</p> <p>B. Không</p>

Câu 4: Bạn cảm thấy giá gửi xe của trường đã phù hợp hay chưa?

- A. Hợp lý
- B. Chưa hợp lý

Câu 5: Bạn cảm thấy an ninh khi gửi xe trong trường như thế nào?

- A. Tốt
- B. Chưa Tốt

Câu 6: Bạn thấy việc sử dụng thẻ gửi xe thay vì vé gửi xe có thuận tiện và dễ dàng hơn không?

- A. Tốt
- B. Không Tốt

Câu 7: Bạn có góp ý kiến gì cho chúng tôi để cải thiện những bất cập khi gửi xe tại trường Đại học Công Nghiệp Hà Nội không?

Câu 8: Bạn có muốn sử dụng các dịch vụ mới như vé tháng, vé năm không?

Chúc bạn có một ngày tốt lành.

2.2.1.3. Đối tượng khảo sát

- Bảo vệ: có nhiệm vụ kiểm vé, phát vé, thu phí giữ xe.
- Sinh viên, cán bộ, giảng viên: Nhận vé, cất xe đúng vị trí bãi đỗ xe, thanh toán cho bảo vệ đúng số tiền theo loại xe.

2.2.2. Mô tả bài toán

Hệ thống hoạt động bắt đầu từ việc người quản trị phân chia ca trực cho các nhân viên bảo vệ của nhà xe. Các nhân viên được chia ca theo từng khu vực (khu A, khu B, khu C), từng ngày với các mốc thời gian: 6h – 12h, 12h – 17h, 17h – 23h. Dựa theo ca trực được phân chia, nhân viên bảo vệ của nhà xe sẽ phụ trách đúng nhiệm vụ của mình.

Về hoạt động trông giữ xe: Sinh viên, học sinh, cán bộ giảng viên khi vào gửi xe sẽ được nhân viên bảo vệ phát cho 1 vé xe. Trên vé có các thông tin:

mã vé, loại vé, giá tiền. Có 3 loại vé: vé xe đạp, vé xe máy và vé giảng viên với giá vé lần lượt là 2000đ, 3000đ và 0đ. Khi lấy xe, sinh viên, học sinh, cán bộ giảng viên cần đưa lại vé cho nhân viên bảo vệ, trả tiền gửi xe. Nhân viên bảo vệ xác nhận loại vé tương ứng với loại xe hay chưa rồi trả xe. Nhân viên thu tiền gửi xe tương ứng với loại vé, ấn vào nút thanh toán trên màn hình hệ thống để cập nhật số tiền thu được.

Về hoạt động thống kê: Hệ thống sẽ thống kê số tiền thu được của từng khu vực theo từng ngày. Người quản trị nhập ngày muốn thống kê, hệ thống sẽ đưa ra tổng tiền thu được của ngày đó.

➤ Yêu cầu chức năng

- Đăng nhập: Cho phép quản trị viên và nhân viên bảo vệ của nhà xe đăng nhập vào hệ thống với quyền tương ứng.
- Quản lí ca trực: Cho phép quản trị viên thêm, sửa, xóa ca trực.
- Quản lí vé xe: Cho phép quản trị viên sửa lại giá vé.
- Quản lý trông xe: Cho phép nhân viên bảo vệ chọn khu vực, loại vé để thực hiện phát vé và thu tiền vé.
- Thanh toán: Cho phép nhân viên bảo vệ của nhà xe cập nhật số tiền thu được vào hệ thống.
- Thống kê: Khi quản trị viên kích vào nút thống kê, hệ thống sẽ thống kê ra tổng số tiền thu về trong một ngày trên từng khu vực.

➤ Yêu cầu phi chức năng

- Các điều khiển trên hệ thống cần quen thuộc, dễ thao tác với người dùng. Thông tin hình ảnh bao quát, đẹp và dễ dàng lựa chọn tìm kiếm.
- Tên các trường thông tin, chức năng phải thống nhất toàn bộ giao diện.

2.3. Phân tích hệ thống

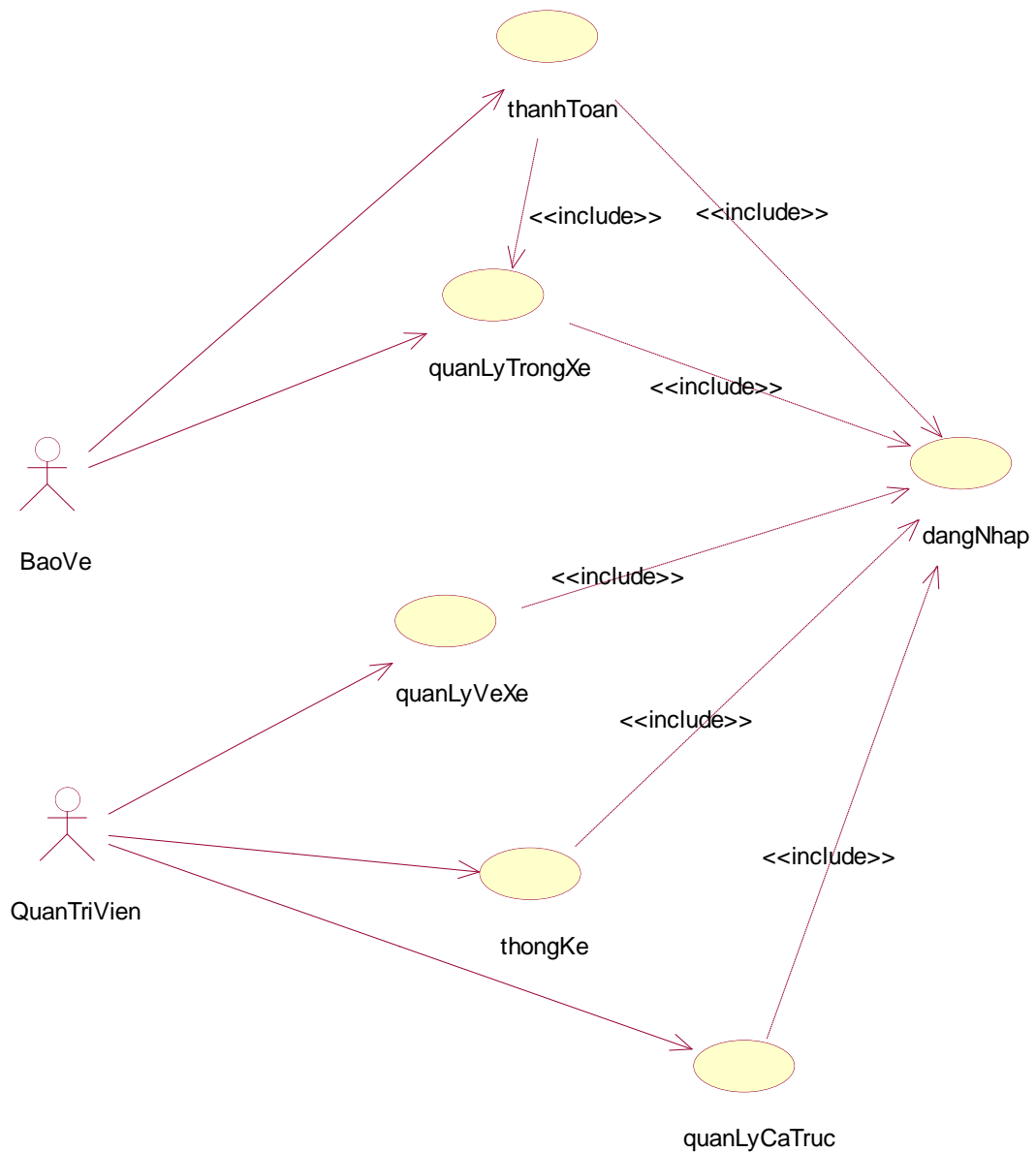
2.3.1. Mô hình hóa chức năng hệ thống

➤ Các actor:

- Quản trị viên: Quản trị viên có nhiệm vụ phân chia ca trực cho nhân viên bảo vệ của nhà xe, quản lý vé xe, thống kê số tiền vé thu được.
- Bảo vệ: Bảo vệ có nhiệm vụ quản lý trông xe: phát vé xe, thu phí gửi xe của học sinh, sinh viên, cán bộ giảng viên của trường.

➤ Các UseCase:

- Đăng nhập: Cho phép quản trị viên và nhân viên bảo vệ của nhà xe đăng nhập vào hệ thống với quyền tương ứng.
- Quản lý ca trực: Cho phép quản trị viên thêm, sửa, xóa ca trực.
- Quản lý vé xe: Cho phép quản trị viên sửa lại giá vé.
- Quản lý trông xe: Cho phép nhân viên bảo vệ chọn khu vực, loại vé để thực hiện phát vé và thu tiền vé.
- Thanh toán: Cho phép nhân viên bảo vệ của nhà xe cập nhật số tiền thu được vào hệ thống.
- Thống kê: Khi quản trị viên kích vào nút thống kê, hệ thống sẽ thống kê ra tổng số tiền thu về trong một ngày trên từng khu vực.



Hình 1. Biểu đồ UseCase tổng quát

➤ **Đặc tả tóm tắt các use-case**

ID	Tên Use case	Mô tả ngắn gọn Use case	Chức năng	Ghi chú
UC1	Đăng nhập	- Đăng nhập với quyền là nhân viên.	- Hệ thống kiểm tra và chuyển sang giao diện cho nhân viên.	Bảo vệ.

		- Đăng nhập với quyền quản trị viên.	- Hệ thống kiểm tra và chuyển sang giao diện quản trị viên.	Quản trị viên.
UC2	Quản lý ca trực	- Chọn “Quản lý ca trực”.	- Hệ thống đọc file “nhanvien.txt”, hiển thị thông tin: Tên nhân viên, khu vực, ngày trực, ca trực lên màn hình.	Quản trị viên.
		- Thêm ca trực: Thêm các thông tin về ca trực: Tên nhân viên, khu vực, ngày trực, ca trực.	- Hệ thống kiểm tra các trường thông tin, ghi dữ liệu vào file “nhanvien.txt”.	
		- Sửa ca trực: Sửa và cập nhật lại các thông tin: Tên nhân viên, khu vực, ngày trực, ca trực.	- Hệ thống kiểm tra các trường thông tin, cập nhật dữ liệu vào file “nhanvien.txt”.	
		- Xóa ca trực: Xóa thông tin của ca trực.	- Hệ thống xóa dữ liệu về ca trực được chọn trong file “nhanvien.txt”.	
UC3	Quản lý vé xe	- Chọn “Quản lý vé xe”.	- Hệ thống đọc file “vexe.txt” và hiển thị thông tin lên màn hình.	Quản trị viên.

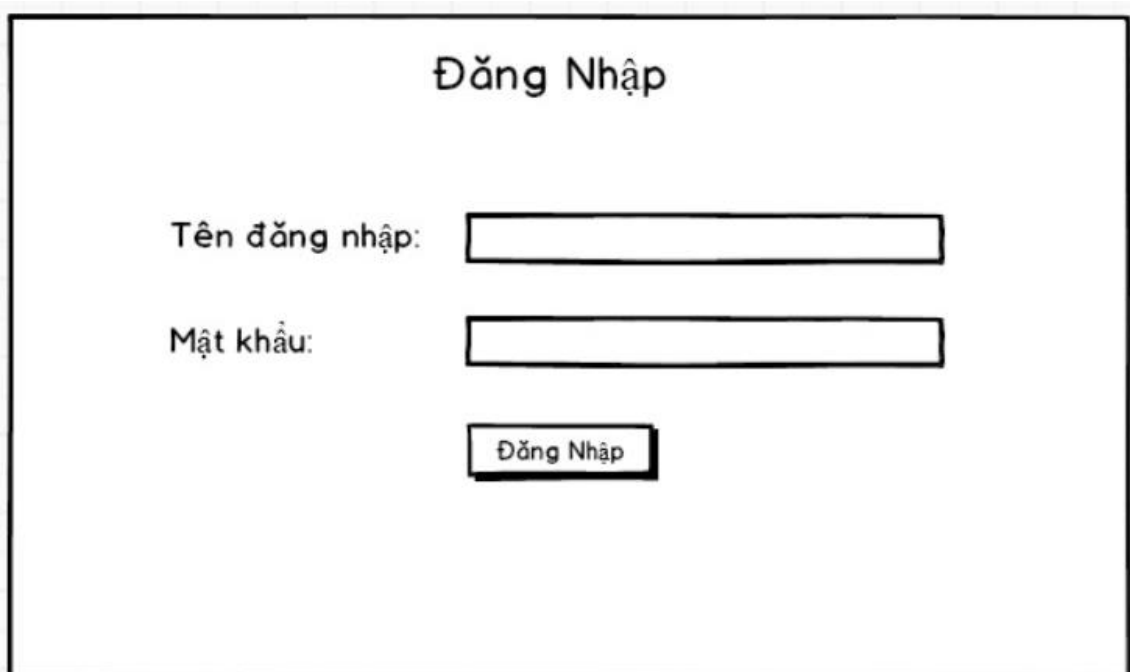
		- Chọn sửa vé xe: Sửa và cập nhật lại giá vé của từng loại vé.	- Hệ thống kiểm tra giá vé, ghi dữ liệu vào file “vexe.txt”. Nếu nhập thiếu thông tin hệ thống báo lỗi và yêu cầu nhập lại.	
UC4	Quản lý trông xe	- Chọn khu vực trông xe, chọn loại vé xe.	- Hệ thống đọc file “khuvc.txt”, “vexe.txt” và hiển thị thông tin ra màn hình.	Bảo vệ.
UC5	Thanh toán	- Chọn “Thanh toán” để xác nhận thanh toán khi khách hàng lấy xe.	- Hệ thống cập nhật số tiền tương ứng với từng loại vé vào file “thongke.txt”, “khuvc.txt”.	Bảo vệ.
UC6	Thống kê	- Chọn “Thống kê”.	- Hệ thống đọc file “thongke.txt” và file “khuvc.txt”, hiển thị thông tin: Khu vực, tổng tiền, ngày trực lên màn hình.	Quản trị viên.
		- Nhập ngày muốn thống kê vào ô “Chọn ngày” và kích “Tìm kiếm”.	- Nếu ngày hợp lệ thì hệ thống đọc file “khuvc.txt” và file “thongke.txt”, hiển thị số tiền thu được theo từng khu vực của ngày đó, đồng thời hiển thị tổng tiền	

			của ngày đó ở ô “Tổng tiền”. - Nếu ngày không hợp lệ, hệ thống hiển thị thông báo lỗi “Ngày không có doanh thu hoặc ngày không hợp lệ”.	
--	--	--	--	--

2.3.2. Mô hình hóa dữ liệu và giao diện hệ thống

2.3.2.1. UseCase Đăng nhập

- Thông tin đầu vào: tên đăng nhập, mật khẩu.
- Thông tin đầu ra: Admin, nhân viên.
- Phác thảo giao diện:



Phác thảo giao diện đăng nhập (Login Form) với tiêu đề "Đăng Nhập".

Form bao gồm các trường sau:

- Tên đăng nhập:
- Mật khẩu:
- Nút "Đăng Nhập" để thực hiện đăng nhập.

Hình 2. Giao diện đăng nhập – Vũ Đức Triệu

2.3.2.2. UseCase Quản lý ca trực

- Thông tin đầu vào: Tên nhân viên, ngày trực, khu trực, ca trực.
- Thông tin đầu ra: Tên nhân viên, ngày trực, khu trực, ca trực.
- Phác thảo giao diện:

Tên Nhân Viên	Khu Trực	Ca Trực	Ngày Trực	Sửa	Xóa
Nhân Viên 1	Khu A	6h - 12h	23/2/2023	Sửa	Xóa
Nhân Viên 2	Khu B	12h - 17h	23/2/2023	Sửa	Xóa
Nhân Viên 3	Khu C	17h - 23h	23/2/2023	Sửa	Xóa

Hình 3. Giao diện quản lý ca trực – Trần Thị Hồng Thắm

Sửa Ca Trực

Tên Nhân Viên:

Khu Trực:

Ca Trực:

Ngày Trực:

Hình 4. Giao diện sửa ca trực – Vũ Đức Triệu

Hình 5. Giao diện thêm ca trực – Trần Thị Hồng Thắm

2.3.2.3. UseCase Quản lý vé xe

- Thông tin đầu vào: Giá tiền.
- Thông tin đầu ra: Mã vé, tên vé, giá tiền.
- Phác thảo giao diện:

Mã Vé	Tên Loại Vé	Giá Tiền
1	Vé Xe Đạp	2000
2	Vé Xe Máy	5000
3	Vé Giảng Viên	0

Sửa

Hình 6. Giao diện quản lý vé xe – Dương Văn Thắng

Sửa Vé Xe

Mã Vé:

Tên Loại Vé:

Giá Tiền:

Hình 7. Giao diện sửa vé xe – Dương Văn Thắng

2.3.2.4. UseCase Quản lý trông xe và Thanh toán

- Thông tin đầu vào: Khu vực, loại vé.
- Thông tin đầu ra: Tổng tiền.
- Phác thảo giao diện:

Quản Lý Trông Xe

Ngày: 23/2/2023

Chọn khu vực:

Loại vé:

 Tổng tiền: 300.000

Hình 8. Giao diện quản lý trông xe và thanh toán – Lê Bá Thành

2.3.2.5. UseCase Thống kê

- Thông tin đầu vào: Tên khu vực, tổng tiền theo khu vực, ngày.
- Thông tin đầu ra: Tên khu vực, tổng tiền theo khu vực, ngày, tổng tiền theo ngày.
- Phác thảo giao diện:

Thống Kê

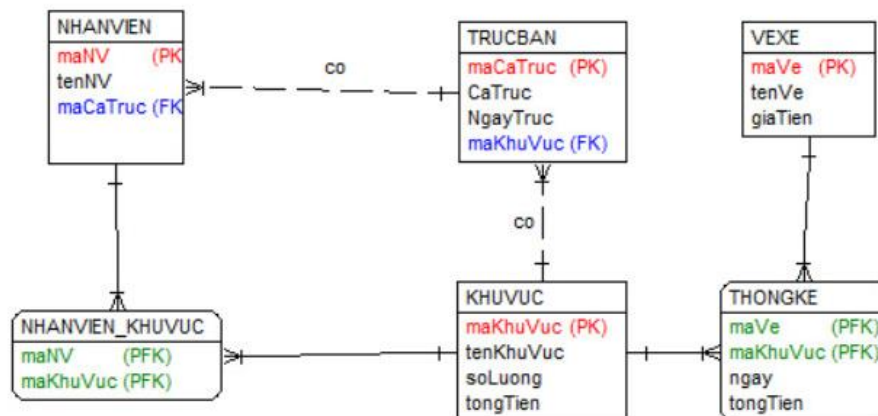
Chọn ngày:

Tên Khu Vực	Tổng Tiền	Ngày
Khu A	1000.000	23/2/2023
Khu B	1000.000	23/2/2023
Khu C	0	23/2/2023

Tổng Tiền: **2.000.000**

Hình 9. Giao diện thống kê – Phạm Minh Hoàng

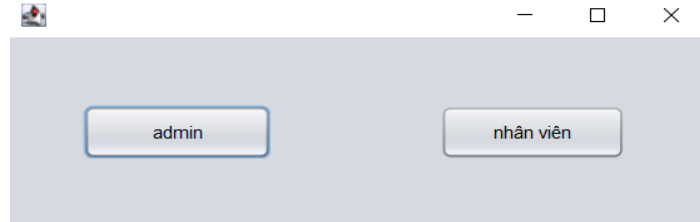
❖ Sơ đồ lớp dữ liệu sẽ cài đặt trong hệ thống



2.4. Thực hiện bài toán

2.4.1. Vũ Đức Triệu – Đăng nhập

- Giao diện “Đăng nhập”



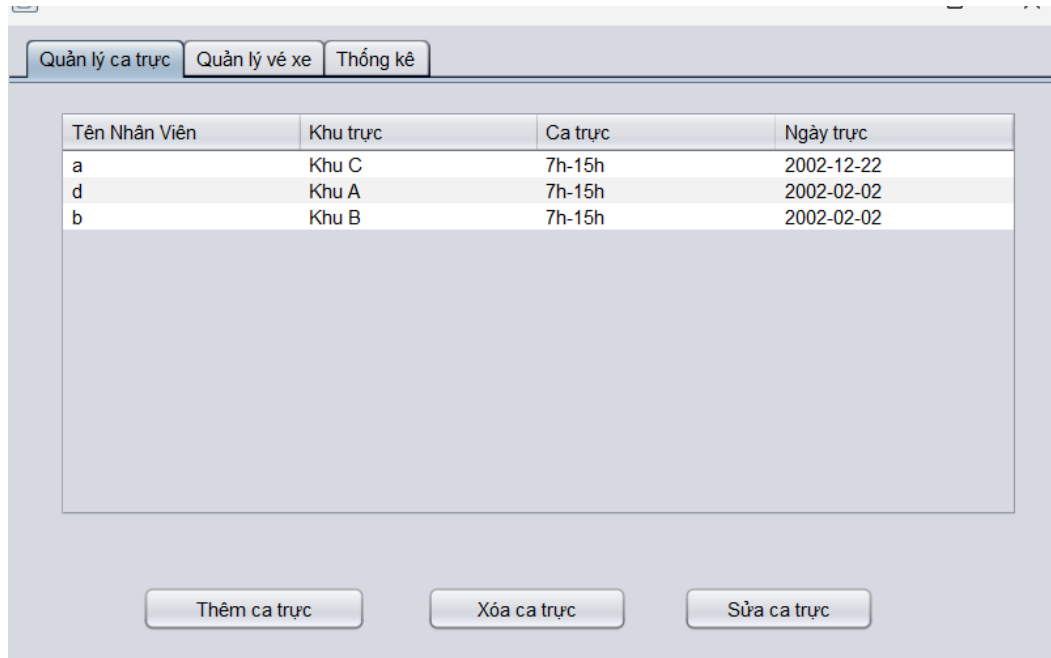
- Mã lệnh:

- Hướng đối tượng: Sử dụng tính kế thừa.

```
public class DangNhapFrm extends javax.swing.JFrame
```

2.4.2. Trần Thị Hồng Thắm – Quản lý ca trực

- Giao diện “Quản lý ca trực”: Hiển thị tên nhân viên, khu trực, ca trực, ngày trực.



Thêm ca trực

Tên Nhân Viên:

Khu Trục:

Ca Trục:

Ngày Trục:

Sửa Ca Trục

Tên nhân viên

Khu trục

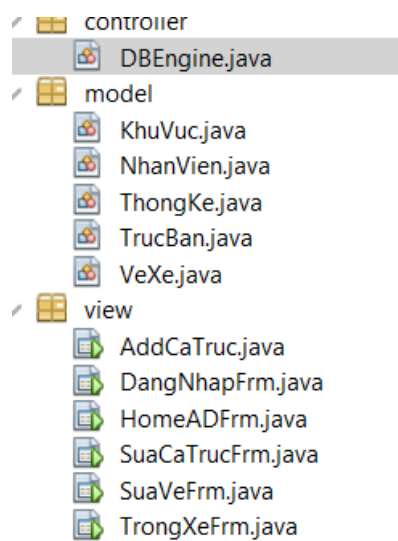
Ca trục

Ngày Trục

➤ Mã lệnh:

- Hướng đối tượng:

- Tính đóng gói: Các file được chia rõ ràng với từng chức năng, nhiệm vụ.



- Tính kế thừa:

```

    */
    public class SuaCaTrucFrm extends javax.swing.JDialog
    {
        /**
         * Creates new form SuaCaTryuc
         */
    }

```

```

    */
    public class AddCaTruc extends javax.swing.JDialog

        /**
         * Creates new form AddCaTruc

```

```

public class NhanVien extends TrucBan implements Serializable{
    private String maNV;
    private String tenNV;

    public NhanVien(String maNV, String tenNV, String caTruc, String khu, String ngayTruc) {
        super(caTruc, khu, ngayTruc);
        this.maNV = maNV;
        this.tenNV = tenNV;
    }

    public NhanVien(String tenNV, String caTruc, String khu, String ngayTruc) {
        super(caTruc, khu, ngayTruc);
        this.tenNV = tenNV;
    }
}

```

- Cơ chế bắt lỗi:

- Bắt lỗi bằng câu lệnh If:

```

boolean isOk = true;
if (tenNV.length() == 0) {
    JOptionPane.showMessageDialog( parentComponent: rootPane,
                                   message: "Họ tên không được để trống!");
    isOk = false;
} else if (ngayTruc.length() == 0) {
    JOptionPane.showMessageDialog( parentComponent: rootPane,
                                   message: "Ngày không được để trống!");
    isOk = false;
} else if (tenNV.matches( regex: "[^a-zA-Z]+")) {
    JOptionPane.showMessageDialog( parentComponent: rootPane,
                                   message: "Họ tên chỉ chứa các chữ cái!");
    isOk = false;
}

```


- Bắt lỗi bằng lệnh try – catch:

```
SimpleDateFormat dateFormat = new SimpleDateFormat( pattern: "yyyy-MM-dd");
dateFormat.setLenient( lenient: false); // Chặn việc chuyển đổi chuỗi không hợp lệ
try {
    Date date = dateFormat.parse( source: ngayTruc);
    System.out.println("Ngày tháng hợp lệ: " + date);
} catch (Exception e) {
    isOk = false;
    JOptionPane.showMessageDialog( parentComponent: rootPane,
        message: "Ngày tháng phải để dạng YYYY-MM-DD!");
    System.out.println("Ngày tháng không hợp lệ: " + ngayTruc);
}
```

- Bắt lỗi đọc file “nhanvien.txt”:

```
public void docFileNV(String fileName) {
    try {
        nhanviens = (ArrayList) db.readDataFromFile(fileName);
        System.out.println( x: " đọc được file");
    } catch (Exception ex) {
        System.out.println( x: "không đọc được file");
    }
}
```

- Tập hợp:

- Sử dụng kiểu tập hợp ArrayList:

```
nhanviens = new ArrayList<>();
tblCaModel = (DefaultTableModel) tblCaTruc.getModel();
```

- Các phương thức:

phương thức thêm nhân viên ở frm add ca trực

```
public void addCaTruc(NhanVien e) {
    nhanviens.add(e);
    showData( list: nhanviens, model: tblCaModel);
}
```

```
//phương thức sửa nhân viên ở frm sửa ca trực
public void suaNhanVien(NhanVien e) {
    nhanviens.remove(index:selectedIndex);
    this.addCaTruc(e);
}
}
```

- Thao tác với file: Sử dụng thao tác đọc và lưu vào file “txt”. Lưu file dưới dạng ArrayList. Hàm đọc và lưu file tổng quát được lưu ở lớp DBEngine.java.

- Lưu file sau khi thêm/ sửa/ xóa ca trực:

```
private void ThemCaActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    AddCaTruc caTrucFrm = new AddCaTruc( parent:this,
        modal:rootPaneCheckingEnabled);
    caTrucFrm.setVisible( b:true);
    lưuFile( T:nhanviens, fileName:fileNameNV);
}

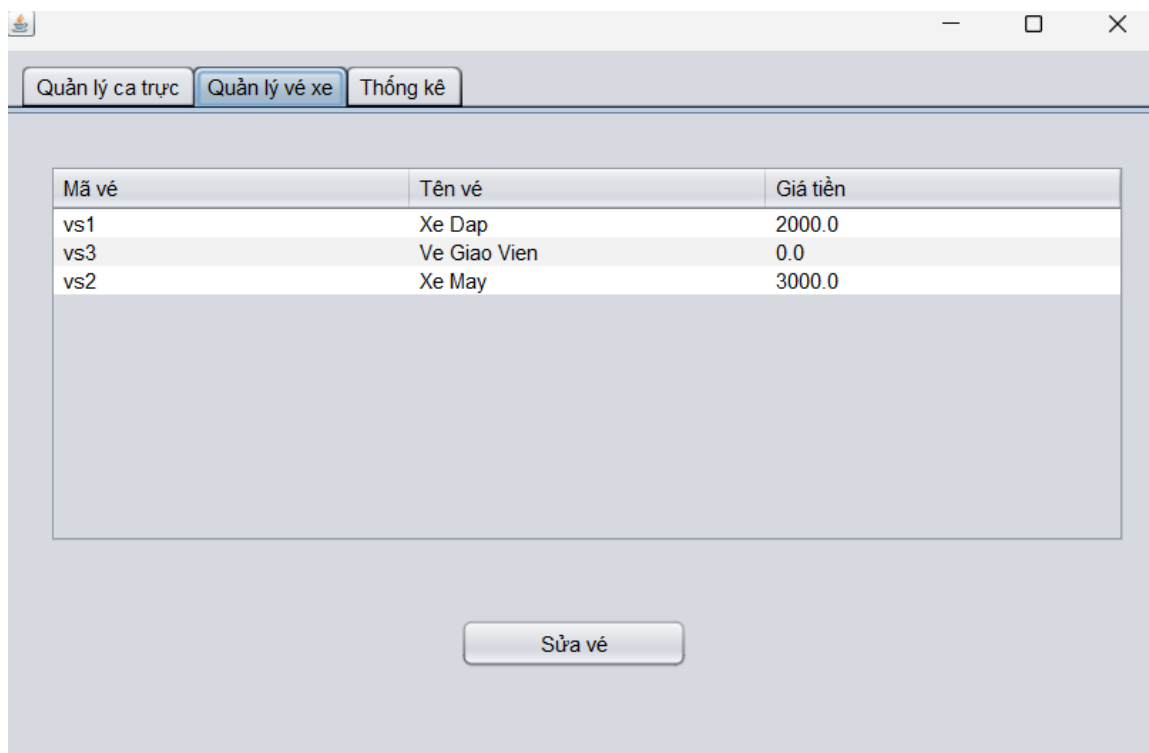
//sửa ca trực
private void SuaCaActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    selectedIndex = tblCaTruc.getSelectedRow();
    if (nhanviens.isEmpty()) {
        JOptionPane.showMessageDialog( parentComponent:rootPane,
            message:"Nhập thêm nhân viên!");
    } else if (selectedIndex == -1) {
        JOptionPane.showMessageDialog( parentComponent:rootPane,
            message:"Hãy chọn sinh viên!");
    } else {
        SuaCaTrucFrm suaCa = new SuaCaTrucFrm( parent:this,
            modal:rootPaneCheckingEnabled);
        suaCa.setSuaCa( nv:nhanviens.get(index:selectedIndex));
        suaCa.setVisible( b:true);
    }
    lưuFile( T:nhanviens, fileName:fileNameNV);
}

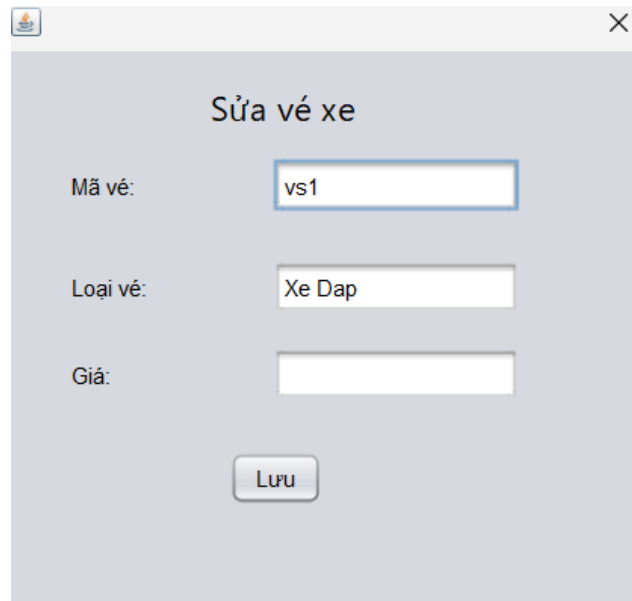
// xóa ca trực nè
```

```
private void XoaCaActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    int removeIndex = tblCaTruc.getSelectedRow();
    if (removeIndex == -1) {
        JOptionPane.showMessageDialog( parentComponent: rootPane,
            message: "Chọn Ca cần xóa!");
    } else if (nhanviens.isEmpty()) {
        JOptionPane.showMessageDialog( parentComponent: rootPane,
            message: "Không có thông tin!");
    } else {
        nhanviens.remove( index: removeIndex);
        showData( list: nhanviens, model: tblCaModel);
    }
    lưuFile( t: nhanviens, fileName: fileNameNV);
}
}
```

2.4.3. Dương Văn Thắng – Quản lý vé xe

- Giao diện “Quản lý vé xe”: Hiển thị mã vé, tên vé, giá tiền của từng loại vé.

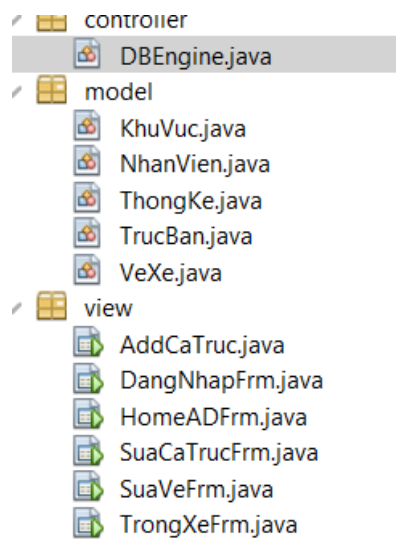




➤ Mã lệnh:

- Hướng đối tượng:

- Tính đóng gói: Các file được chia rõ ràng với từng chức năng, nhiệm vụ.



- Tính kế thừa:

```

    */
    public class SuaVeFrm extends javax.swing.JDialog {

        /**
         * Creates new form QuanLyVeFrm
         */
        private HomeADFrm home;
    }

```

- Cơ chế bắt lỗi:

- Bắt lỗi giá vé không được để trống bằng câu lệnh If:

```
giave = txtGiave.getText();
boolean isOk = true;
//    Bắt lỗi giá vé không được để trống
if (giaVe.length() == 0) {
    JOptionPane.showMessageDialog( parentComponent: rootPane, message: "Giá vé không được để trống!");
    isOk = false;
}
```

- Bắt lỗi dữ liệu của giá vé bằng lệnh try – catch:

```
float x = 0f;
    Bắt lỗi dữ liệu của giá vé
try{
    x = Float.parseFloat( s:giaVe);
}catch(Exception e){
    System.out.println( x: "Giá vé không hợp lệ");
}
```

- Bắt lỗi đọc file “vexe.txt”:

```
public void docFileVS(String fileName) {
/    Bắt lỗi đọc file vé xe
    try {
        vexes = (ArrayList) db.readDataFromFile(fileName);
        System.out.println( x: "đọc được file");
    } catch (Exception ex) {
        System.out.println( x: "không đọc được file");
    }
}
```

- Tập hợp:

- Sử dụng kiểu tập hợp ArrayList

```
//
private ArrayList<VeXe> vexes;
private final DefaultTableModel tblVeModel;
//
```

- Sử dụng phương thức add() và remove().

```
vexes.remove( index: selectedIndex);
vexes.add( e: vs);
```

- Thao tác file: Sử dụng thao tác đọc và lưu vào file “txt”. Lưu file dưới dạng ArrayList. Hàm đọc và lưu file tổng quát được lưu ở lớp DBEngine.java.

- Vé xe sau khi sửa sẽ được lưu như sau:

```
        lưuFile( T: vexes, fileName: fileNameVS);
    }
```

- Để đọc file:

```
public void docFileVS(String fileName) {
    /    Bắt lỗi đọc file vé xe
    try {
        vexes = (ArrayList) db.readDataFromFile(fileName);
        System.out.println(x: "đọc được file");
    } catch (Exception ex) {
        System.out.println(x: "không đọc được file");
    }
}
```

2.4.4. Phạm Minh Hoàng – Thống kê

- Giao diện “Thống kê”: Hiển thị tên khu vực, tổng tiền theo khu vực, ngày, tổng tiền theo ngày.

The screenshot shows a web application interface with three tabs: 'Quản lý ca trực', 'Quản lý vé xe', and 'Thống kê'. The 'Thống kê' tab is active, displaying a table with the following data:

Tên khu vực	Tổng tiền	Ngày
Khu A	8000.0	2023-02-25
Khu B	6000.0	2023-02-25
Khu C	3000.0	2023-02-25
Khu A	66000.0	2023-02-26
Khu C	10000.0	2023-02-26
Khu B	12000.0	2023-02-26

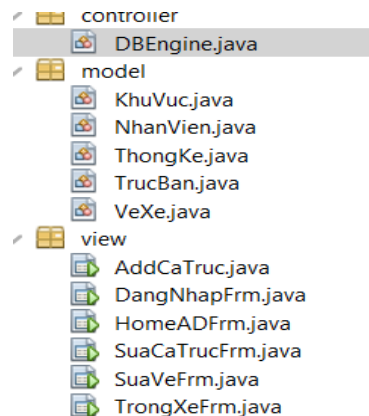
Below the table, there is a search section with the following elements:

- A label 'Chọn Ngày:' followed by a text input field containing 'yyyy-mm-dd'.
- A 'Tìm kiếm' (Search) button.
- A 'Load' button.
- A label 'Tổng tiền:' followed by an empty text input field.

➤ Mã lệnh:

- Hướng đối tượng:

- Tính đóng gói: Các file được chia rõ ràng với từng chức năng nhiệm vụ. Ví dụ: File “ThongKe.java” chỉ có chức năng thống kê.



- Để bảo vệ dữ liệu và tránh các truy cập trái phép, các thuộc tính của các lớp được khai báo với các từ khóa private.

```
private String ngay;
private float tongTien;
private String khuVuc;
```

- Tính kế thừa:

```
public class ThongKe implements Serializable{
```

- Cơ chế bắt lỗi:

- Bắt lỗi đọc file thống kê:

```
public void docFileTK(String fileName) {
    try {
        thongkes = (ArrayList) db.readDataFromFile(fileName);
        System.out.println("đọc được file");
    } catch (Exception ex) {
        System.out.println("không đọc được file");
    }
}
```

- Bắt lỗi nhập ngày thống kê không đúng bằng câu lệnh If:

```
private void TimKiemNgayActionPerformed(java.awt.event.ActionEvent evt) {GEN
    // TODO add your handling code here:
    String ngay;
    float tongTien = 0;
    boolean check = false;
    ngay = txtNgay.getText();
    tblTKModel.setRowCount(0);
    for (ThongKe tk : thongkes) {
        if (ngay.equals(tk.getNgay())) {
            tblTKModel.addRow(new Object[]{
                tk.getKhuVuc(), tk.getTongTien(), tk.getNgay()
            });
            tongTien += tk.getTongTien();
            check = true;
        }
    }
    txtTien.setText(Float.toString(tongTien));

    if (check == false) {
        JOptionPane.showMessageDialog(rootPane
            , "ngày đó không có doanh thu hoặc ngày nhập không hợp lệ!");
        showData(thongkes, tblTKModel);
    }
}
//GEN-LAST:event_TimKiemNgayActionPerformed
```

- Tập hợp: Sử dụng kiểu tập hợp ArrayList:

```
private ArrayList<ThongKe> thongkes;
private final DefaultTableModel tblTKModel;
```

- Thao tác file: Sử dụng thao tác đọc và lưu vào file “txt”. Lưu file dưới dạng ArrayList. Hàm đọc và lưu file tổng quát được lưu ở lớp DBEngine.java.

- Thống kê sau khi được tìm sẽ được lưu vào file như sau:

```
lưuFile(T: thongkes, fileName: filenamethonglke)
```

- Để đọc file:

```
public void docFileTK(String fileName) {
    try {
        thongkes = (ArrayList) db.readDataFromFile(fileName);
        System.out.println(" đọc được file");
    } catch (Exception ex) {
        System.out.println("không đọc được file");
    }
}
```


2.4.5. Lê Bá Thành – Quản lý trông xe

- Giao diện “Quản lý trông xe”: Hiển thị ngày, khu vực trông xe, loại vé.

- Mã lệnh:

- Hướng đối tượng:

- Encapsulation (đóng gói): Để bảo vệ dữ liệu và tránh các truy cập trái phép, các thuộc tính của các lớp được khai báo với các từ khóa private.

```
private String nowDate;
private final DBEngine db = new DBEngine();
```

- Inheritance (kế thừa): Các lớp được tạo ra có thể kế thừa từ các lớp khác thông qua từ khóa "extends". Ví dụ, lớp TrongXeFrm được kế thừa từ lớp JFrame.

```
public class TrongXeFrm extends javax.swing.JFrame {
```

- Polymorphism (đa hình): Các lớp có thể triển khai các phương thức có cùng tên nhưng có cách thức thực hiện khác nhau. Ví dụ, các phương thức "docFileKV()", "docFileVS()", và "docFileTK()" cùng tên nhưng được sử dụng để đọc các tệp khác nhau.

```
public void docFileKV(String fileName) {
    try {
        khuvucs = (ArrayList) db.readDataFromFile(fileName);
        System.out.println( x:" đọc được file");
    } catch (Exception ex) {
        System.out.println( x:"không đọc được file");
    }
}
```

```

public void docFileVS(String fileName) {
    try {
        vexes = (ArrayList) db.readDataFromFile(fileName);
        System.out.println(x:" đọc được file");
    } catch (Exception ex) {
        System.out.println(x:"không đọc được file");
    }
}

public void docFileTK(String fileName) {
    try {
        thongkes = (ArrayList) db.readDataFromFile(fileName);
        System.out.println(x:" đọc được file");
    } catch (Exception ex) {
        System.out.println(x:"không đọc được file");
    }
}

```

- Modularity (tính module): Các chức năng được phân chia thành các phương thức riêng lẻ, giúp cho việc sửa đổi và bảo trì mã nguồn dễ dàng hơn. Ví dụ, các phương thức "luuFile()", "docFileKV()", "docFileVS()", và "docFileTK()" đều đảm nhiệm các chức năng khác nhau và được tách ra thành các phương thức riêng biệt.

```

public void luuFile(ArrayList T, String fileName) {
    try {
        db.writeToFile(list:T, fileName);
        System.out.println(x:"Luu file thành công");
    } catch (Exception ex) {
        System.out.println(x:ex.toString());
    }
}

//đọc file lưu trở lại phòng thi

public void docFileKV(String fileName) {
    try {
        khuvucs = (ArrayList) db.readDataFromFile(fileName);
        System.out.println(x:" đọc được file");
    } catch (Exception ex) {
        System.out.println(x:"không đọc được file");
    }
}

```

- Cơ chế bắt lỗi: Cơ chế bắt lỗi đọc và ghi file sử dụng lệnh try – catch.

```

public void lưuFile(ArrayList T, String fileName) {
    try {
        db.writeToFile(list:T, fileName);
        System.out.println(x:"Lưu file thành công");
    } catch (Exception ex) {
        System.out.println(x:ex.toString());
    }
}

//đọc file lưu trở lại phòng thi

public void docFileKV(String fileName) {
    try {
        khuvucs = (ArrayList) db.readDataFromFile(fileName);
        System.out.println(x:" đọc được file");
    } catch (Exception ex) {
        System.out.println(x:"không đọc được file");
    }
}

```

- Tập hợp:

- Các biến thành viên khuvucs, vexes và thongkes đều là các instance của lớp ArrayList được sử dụng để lưu trữ một tập đối tượng động. khuvucs và vexes lần lượt chứa các thể hiện của lớp KhuVuc và VeXe, trong khi thongkes chứa các instance của lớp ThongKe.

```

-      ^/
      ArrayList<KhuVuc> khuvucs;
      ArrayList<VeXe> vexes;
      ArrayList<ThongKe> thongkes;

```

- Các phương thức sử dụng:

```

VeXe vs1 = new VeXe( maVe: "vs1", tenVe: "Xe Dap", giaTien: 2000);
vexes.add( e: vs1);

```

```

KhuVuc kv1 = new KhuVuc( khuVuc: "Khu A");
khuvucs.add( e: kv1);

```

- Thao tác với file: Sử dụng đọc và lưu file “txt”;

- Lưu lại giao dịch thanh toán đã thực hiện

```

lưuFile( T: khuvucs, fileName: fileNameKV);
lưuFile( T: thongkes, fileName: fileNameTK);

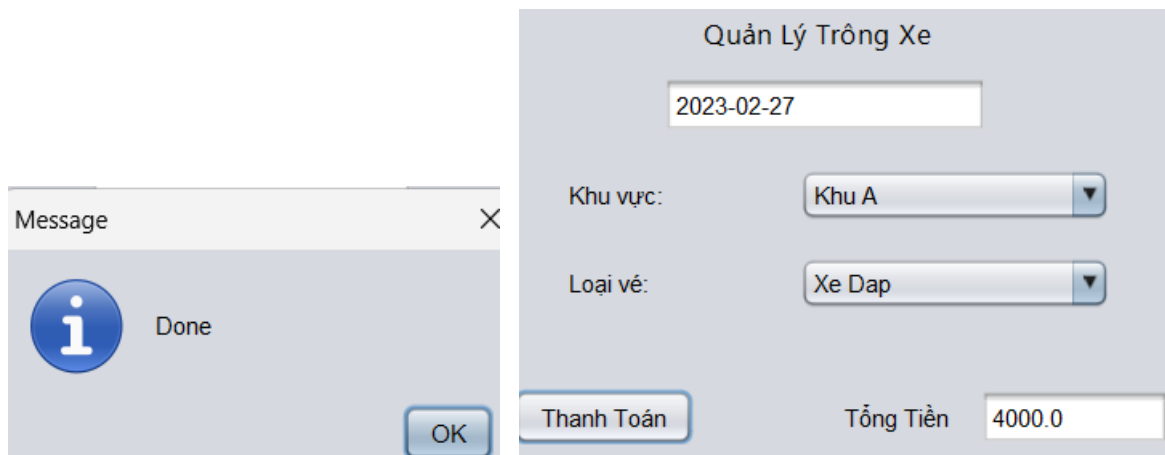
```

- Đọc file:

```
public void docFileTK(String fileName) {
    try {
        thongkes = (ArrayList) db.readDataFromFile(fileName);
        System.out.println(x: "đọc được file");
    } catch (Exception ex) {
        System.out.println(x: "không đọc được file");
    }
}
```

2.4.6. Vũ Đức Triệu – Thanh toán

- Giao diện “Thanh toán”: Hiển thị số tiền thu được của khu vực theo ngày.



- Mã lệnh:

- Hướng đối tượng:

Tính đóng gói: Để bảo vệ dữ liệu và tránh các truy cập trái phép, các thuộc tính của các lớp được khai báo với các từ khóa private.

```
public class ThongKe implements Serializable{
    private String ngay;
    private float tongTien;
    private String khuVuc;
```

- Cơ chế bắt lỗi: Bắt lỗi đọc và ghi file sử dụng lệnh try – catch.

```
public void luuFile(ArrayList T, String fileName) {
    try {
        db.writeToFile(list: T, fileName);
        System.out.println(x: "Lưu file thành công");
    } catch (Exception ex) {
        System.out.println(x: ex.toString());
    }
}
```

```

public void docFileKV(String fileName) {
    try {
        khuvucs = (ArrayList) db.readDataFromFile(fileName);
        System.out.println(x: "đọc được file");
    } catch (Exception ex) {
        System.out.println(x: "không đọc được file");
    }
}

```

- Tập hợp: Sử dụng tập hợp kiểu ArrayList.

```

//
ArrayList<KhuVuc> khuvucs;
ArrayList<VeXe> vexes;
ArrayList<ThongKe> thongkes;

```

- Thao tác với file:

- Lưu lại giao dịch thanh toán đã thực hiện

```

luuFile( T: khuvucs, fileName: fileNameKV);
luuFile( T: thongkes, fileName: fileNameTK);

```

- Đọc file:

```

public void docFileTK(String fileName) {
    try {
        thongkes = (ArrayList) db.readDataFromFile(fileName);
        System.out.println(x: "đọc được file");
    } catch (Exception ex) {
        System.out.println(x: "không đọc được file");
    }
}

```

PHẦN 3: KẾT LUẬN VÀ BÀI HỌC KINH NGHIỆM

3.1. Nội dung đã thực hiện

Sau khi hoàn thành đề tài “Xây dựng phần mềm quản lý trông xe Đại học Công nghiệp Hà Nội”, nhóm chúng em đã có thể nắm được những kiến thức chuyên môn của môn học như:

- Lập trình Java cơ sở.
- Lập trình Java hướng đối tượng.
- Xử lý ngoại lệ, bắt lỗi.
- Thao tác với file.
- Tập hợp.
- Biết cách sử dụng công cụ NetBeans.

Ngoài những kiến thức liên quan đến môn học, chúng em còn tích lũy, trau dồi thêm những kỹ năng mềm khác như:

- Kỹ năng làm việc nhóm.
- Kỹ năng thu thập và chọn lọc thông tin.
- Kỹ năng quản lý thời gian, phân chia công việc.
- Biết lắng nghe và tiếp thu ý kiến của các thành viên.

Những chuẩn đầu ra của học phần đã đạt được:

- Phát biểu được bài toán cần xây dựng, phân tích và đưa ra sơ đồ lớp mô tả bài toán theo hướng đối tượng.
- Vận dụng các kỹ thuật đã học trong lập trình Java cài đặt được bài toán theo chủ đề được giao.
- Sản phẩm nghiên cứu: Quyền thuyết minh chủ đề “Xây dựng phần mềm quản lý trông xe Đại học Công nghiệp Hà Nội” và sản phẩm mã nguồn.

3.2. Hướng phát triển

➤ Ưu điểm:

- Giúp nhà trường dễ dàng quản lý hoạt động gửi xe, trông xe, thống kê doanh thu và phân chia ca trực.
- Giao diện đơn giản, dễ dàng sử dụng.

➤ Nhược điểm:

- Do lượng kiến thức còn hạn chế nên việc tìm hiểu triển khai tính năng vẫn còn nhiều khó khăn, thiếu sót.
- Giao diện còn khá đơn giản.
- Các chức năng của phần mềm còn hạn chế, một số chức năng vẫn còn gặp lỗi.
- Một vài thành phần có thời gian xử lý lâu chưa được tối ưu và tính bảo mật cũng chưa đảm bảo nhiều.

➤ Định hướng phát triển:

- Phần mềm đã đáp ứng thực hiện được các chức năng cơ bản cần thiết để phục vụ quản lý trông xe tại trường học.
- Phần mềm có thể nâng cấp các chức năng khác thành hệ thống lớn hơn theo nhu cầu sử dụng của cá nhân, tổ chức.

TÀI LIỆU THAM KHẢO

1. Tài liệu học tập:

- Giáo trình Lập trình HĐT với Java, Nguyễn Bá Nghiênn, Ngô Văn Bình, Vương Quốc Dũng, Đỗ Sinh Trường; NXB Thống kê, 2020.
- Bộ slide bài giảng lập trình java- Bộ môn CMPM- trường ĐHCN HN.
- The Java Programming Language; Author: K. Arnold, J. Gosling; Published: Addison-Wesley, 1996, ISBN 0-201-63455-4.
- Lập trình hướng đối tượng với Java; Đoàn Văn Ban; NXB Khoa học và Kỹ thuật, Hà Nội 2006 (Tái bản).
- Lập trình Java nâng cao, Đoàn Văn Ban, NXB Khoa học và Kỹ thuật, Hà Nội 2006.

2. Phương tiện, nguyên liệu thực hiện bài tập lớn:

- Máy tính, bộ công cụ phát triển ứng dụng JDK phiên bản 19.
- Phần mềm hỗ trợ viết mã nguồn NetBeans 16.
- Phần mềm soạn thảo văn bản Microsoft word.
- Phần mềm hỗ trợ phân tích thiết kế hệ thống: Rational Rose, CaseStudio 2, Balsamiq Mockups 3.