TRƯỜNG ĐẠI HỌC ĐÀ LẠT  
KHOA CÔNG NGHỆ THÔNG TIN  
🙠🙠🕮🙢🙢



TIỂU LUẬN HỌC PHẦN  
NGÔN NGỮ LẬP TRÌNH JAVA

XÂY DỰNG ỨNG DỤNG QUẢN LÝ BÁN HÀNG – SỬ DỤNG JAVA - SWING

**Giáo viên hướng dẫn:** Đoàn Minh Khuê

**Sinh viên thực hiện:** 2115239 – Trần Văn Nam,

2113024 – Nông Đức Trí.

*Đà Lạt, ngày 28, tháng 12, năm 2023*

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Đà Lạt, ngày . . . . tháng năm 2023

Giáo viên hướng dẫn   
[Ký tên và ghi rõ họ tên]

LỜI CẢM ƠN

Đầu tiên, chúng em xin gửi lời cảm ơn chân thành đến những người đã giúp đỡ chúng em trong suốt quá trình học tập bộ môn Ngôn ngữ lập trình Java. Đặc biệt, chúng em xin gửi lời cảm ơn sâu sắc đến giảng viên bộ môn – thầy Đoàn Minh Khuê đã dạy dỗ, truyền đạt, góp ý những kiến thức quý báu cho chúng em trong suốt thời gian thực hiện đề tài vừa qua. Trong thời gian thực hiện đề tài, chúng em đã có thêm nhiều kiến thức bổ ích, tinh thần học tập, làm việc nhóm hiệu quả, nghiêm túc. Đây chắc chắn sẽ là những kiến thức quý báu, là hành trang để chúng em có thể vững bước sau này

Bộ môn Ngôn ngữ lập trình Java là một môn học thú vị, vô cùng bổ ích và có tính thực tế cao. Đảm bảo cung cấp đủ kiến thức. Tuy nhiên, do vốn kiến thức còn nhiều hạn chế và khả năng tiếp thu thực tế còn nhiều bỡ ngỡ. Mặc dù chúng em đã cố gắng hết sức nhưng chắc chắn bài tiểu luận này khó có thể tránh khỏi những thiếu sót. Kính mong thầy xem xét góp ý để bài tiểu luận của chúng em được hoàn thiện hơn.

Chúng em xin chân thành cảm ơn.

MỤC LỤC

[NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN 3](#_Toc153952473)

[LỜI CẢM ƠN 4](#_Toc153952474)

[MỤC LỤC 5](#_Toc153952475)

[DANH MỤC HÌNH ẢNH 7](#_Toc153952476)

[TỔNG QUAN 8](#_Toc153952477)

[TÓM TẮT 9](#_Toc153952478)

[MỞ ĐẦU 10](#_Toc153952479)

[Chương 1: KHÁI QUÁT VỀ NGÔN NGỮ LẬP TRÌNH JAVA VÀ THƯ VIỆN JAVA SWING 11](#_Toc153952480)

[1.1 Khái quát ngôn ngữ lập trình Java 11](#_Toc153952481)

[1.1.1 Khái niệm 11](#_Toc153952482)

[1.1.2 Lịch sử phát triển 11](#_Toc153952483)

[1.1.3 Hạn chế của ngôn ngữ lập trình Java 11](#_Toc153952484)

[1.2 Khái quát thư viện hỗ trợ đồ họa Java Swing 12](#_Toc153952485)

[1.2.1 Khái niệm 12](#_Toc153952486)

[1.2.2 So sánh giữa Java Swing và AWT 12](#_Toc153952487)

[Chương 2: CÁC NỀN TẢNG, MÔ HÌNH VÀ THƯ VIỆN SỬ DỤNG 13](#_Toc153952488)

[2.1 IDE 13](#_Toc153952489)

[2.2 Cơ sở dữ liệu 13](#_Toc153952490)

[2.3 Mô hình MVC 14](#_Toc153952491)

[2.4 Các nền tảng và thư viện khác 16](#_Toc153952492)

[2.3.1 Xampp 16](#_Toc153952493)

[2.3.2 FlatLaf 16](#_Toc153952494)

[2.3.3 Imgscalr 16](#_Toc153952495)

[2.3.4 MySQL Connector/J 16](#_Toc153952496)

[2.3.5 JFreeChart 17](#_Toc153952497)

[2.3.6 Jcalendar 17](#_Toc153952498)

[Chương 3: CÁC CHỨC NĂNG CỦA CHƯƠNG TRÌNH VÀ PHÂN TÍCH THIẾT KẾ 18](#_Toc153952499)

[3.1 Các chức năng của chương trình 18](#_Toc153952500)

[3.2 Phân tích thiết kế 20](#_Toc153952501)

[3.2.1 Màn hình chính 21](#_Toc153952502)

[3.2.2 Quản lý nhân viên 22](#_Toc153952503)

[3.2.3 Quản lý loại hàng hóa 22](#_Toc153952504)

[3.2.4 Quản lý đặt hàng 23](#_Toc153952505)

[3.2.5 Chỉnh sửa hóa đơn 24](#_Toc153952506)

[3.2.6 Quản lý khách hàng 25](#_Toc153952507)

[3.2.7 Quản lý bàn 27](#_Toc153952508)

[3.2.8 Thống kê 28](#_Toc153952509)

[3.2.9 Thiết lập 30](#_Toc153952510)

[Chương 4: QUÁ TRÌNH THỰC HIỆN VÀ MÃ NGUỒN 33](#_Toc153952511)

[4.1 Xây dựng dữ liệu 33](#_Toc153952512)

[4.1.2 Các bảng dữ liệu 33](#_Toc153952513)

[4.1.3 Lược đồ quan hệ 36](#_Toc153952514)

[4.2 Xây dựng mã nguồn 36](#_Toc153952515)

[4.2.1 Kết nối cơ sở dữ liệu 36](#_Toc153952516)

[4.2.2 Các truy vấn 40](#_Toc153952517)

[Chương 5: KIỂM THỬ VÀ KẾT QUẢ 47](#_Toc153952518)

[5.1 Kết quả đạt được 47](#_Toc153952519)

[5.2 Hạn chế 47](#_Toc153952520)

[Chương 6: KẾT LUẬN 49](#_Toc153952521)

[TÀI LIỆU THAM KHẢO 50](#_Toc153952522)

DANH MỤC HÌNH ẢNH

[1. Hình 2.3. Minh họa mô hình MVC 15](#_Toc153951088)

[2. Hình 2.3.1. Triển khai mô hình MVC trong dự án 15](#_Toc153951089)

[3.Hình 3.2.2. Màn hỉnh quản lý nhân viên 22](#_Toc153951090)

[4.Hình 3.2.3. Màn hình quản lý loại món ăn 23](#_Toc153951091)

[5.Hình 3.2.3. Màn hình quản lý tất cả món ăn 23](#_Toc153951092)

[6.Hình 3.2.4. Màn hình quản lý tất cả đơn đặt hàng 24](#_Toc153951093)

[7. Hình 3.2.5. Giao diện chỉnh sửa hóa đơn 25](#_Toc153951094)

[8. Hình 3.2.6. Giao diện tổng danh sách khách hàng 25](#_Toc153951095)

[9. Hình 3.2.6.2. Cửa sổ bật lên khi nhấn nút thêm khách hàng 26](#_Toc153951096)

[10. Hình 3.2.6.3. Cửa sổ bật lên khi nhấn sửa khách hàng 26](#_Toc153951097)

[11. Hình 3.2.7. Giao diện quản lý bàn 27](#_Toc153951098)

[12. Hình 3.2.8. Giao diện thống kê 28](#_Toc153951099)

[13. Hình 3.2.8.1. Giao diện tổng thống kê 29](#_Toc153951100)

[14. Hình 3.2.9. Giao diện thông tin cá nhân 30](#_Toc153951101)

[15. Hình 3.2.9.1. Cửa số bật lên khi đổi mật khẩu 30](#_Toc153951102)

[16. Hình 3.2.9.2. Cửa số bật lên khi xem lịch sử đăng nhập 31](#_Toc153951103)

[17. Hình 3.2.10. Tất cả các quyền của nhân viên 32](#_Toc153951104)

[18. Hình 4.1.2. Các bảng dữ liệu 35](#_Toc153951105)

[19. Hình 4.1.3. Quan hệ giữa các bảng. 36](#_Toc153951106)

TỔNG QUAN

**Tên đề tài:** Xây dựng ứng dụng quản lý bán hàng – sử dụng Java Swing

**Sinh viên thực hiện:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Lớp** | **Công việc** |
| 1 | Trần Văn Nam | 2115239 | CTK45PM | Thiết kể cơ sở dữ liệu, tìm hiểu mô hình MVC, quản lý phân quyền, lập trình các chức năng thêm, xóa, sửa của các chức năng, chức năng theo dõi phiên làm việc, toàn bộ thống kê, hỗ trợ viết báo cáo. |
| 2 | Nông Đức Trí | 2113024 | CTK45PM | Viết báo cáo, lập trình chức năng tìm kiếm, hiển thị các danh sách. |

**Giáo viên hướng dẫn:** Đoàn Minh Khuê

**Mục tiêu đề tài:** Áp dụng những kiến thức đã tìm hiểu về ngôn ngữ lập trình Java, và các thư viện hỗ trợ, từ đó, phát triển thành một ứng dụng quản lý bán hàng hoàn chỉnh.

TÓM TẮT

Báo cáo này đi qua những tìm hiểu về ngôn ngữ lập trìn Java và những thư viện hỗ trợ, đặc biệt là thư viện hỗ trợ đồ họa Java Swing, từ đó có thể phát triển thành một ứng dụng quản lý bán hàng sử dụng Java Swing và các nền tảng hỗ trợ khác. Từ đó rút ra được điểm mạnh khi sử dụng công cụ này.

MỞ ĐẦU

Trong thời đại số hóa ngày càng phát triển, việc sử dụng phần mềm quản lý là một phần quan trọng trong quá trình hoạt động của các doanh nghiệp. Điều này giúp họ tối ưu hóa quá trình quản lý hàng hóa, dịch vụ, nhân sự, tương tác với thông tin của doanh nghiệp và khách hàng một cách dễ dàng. Với những lợi ích như vậy, những ứng dụng dùng để quản lý việc kinh doanh nói chung và việc quản lý bán hàng nói riêng đã trở thành một công cụ hữu ích trong việc quản lý kinh doanh của doanh nghiệp.

Điều đáng chú ý là không chỉ các doanh nghiệp lớn mới có thể sử dụng những ứng dụng như vậy, mà các doanh nghiệp vừa vả nhỏ cũng có thể tiếp cận các lợi ích này.

Vì vậy, việc tìm hiểu về các ngôn ngữ lập trình nói chung và ngôn ngữ lập trình Java nói riêng là một điều rất có ích, và dây cũng chính là nguyên nhân chúng tôi chọn xây dựng ứng dụng dựa trên ngôn ngữ lập trình Java.

Trong báo cáo này, chúng tôi đã tìm hiểu một cách tổng quan về ngôn ngữ lập trình Java và một số kỹ thuật lập trình, sau đó đã tiến hành so sánh giữa các thư viện hỗ trợ đồ họa với nhau và đã quyết định chọn thư viện hỗ trợ đồ họa Java Swing để hoàn thành đề tài này, từ đó rút ra được điểm mạnh và điểm yếu của các thư viện.

Chương 1:   
KHÁI QUÁT VỀ NGÔN NGỮ LẬP TRÌNH JAVA VÀ THƯ VIỆN JAVA SWING

* 1. Khái quát ngôn ngữ lập trình Java
     1. Khái niệm

Java là ngôn ngữ lập trình hướng đối tượng [1] được phát triển ời James Gosling vào đầu những năm 1990. Nhóm đã khởi xướng dự án này để phát triển ngôn ngữ cho các thiết bị kỹ thuật số như hộp giải mã tín hiệu, tivi,…

* + 1. Lịch sử phát triển

Lịch sử của Java rất thú vị. Java ban đầu được thiết kế cho truyền hình tương tác, nhưng nó là công nghệ quá tiên tiến đối với ngành truyền hình cáp kỹ thuật số vào thời điểm đó. Lịch sử của Java bắt đầu với Đội Xanh. Các thành viên nhóm Java (còn được gọi là Green Team), đã khởi xướng dự án này để phát triển ngôn ngữ cho các thiết bị kỹ thuật số như hộp giải mã tín hiệu, tivi, v.v. Tuy nhiên, nó phù hợp nhất cho lập trình internet. Sau đó, công nghệ Java được Netscape kết hợp.

Các nguyên tắc để tạo lập trình Java là "Đơn giản, Mạnh mẽ, Di động, Độc lập với nền tảng, Bảo mật, Hiệu suất cao, Đa luồng, Trung tính về kiến ​​trúc, Hướng đối tượng, Diễn giải và Động". [Java](https://www.javatpoint.com/java-tutorial) được phát triển bởi James Gosling, người được mệnh danh là cha đẻ của Java, vào năm 1995. James Gosling và các thành viên trong nhóm của ông đã bắt đầu dự án này từ rất sớm.

* + 1. Hạn chế của ngôn ngữ lập trình Java

Là một ngôn ngữ cấp cao, nó phải xử lý các cấp độ biên dịch và trừu tượng của một máy ảo. Java thể hiện hiệu xuất kém, nguyên nhân chính là do bộ thu gom rác, cấu hình bộ nhớ đệm không hợp lệ và bế tắc giữa các quy trình.

Java có rất ít trình tạo Gui – Swing, SWT, JSF và JavaFX là những trình xây dựng phổ biến hơn.

Để viết mã để thực hiện một tập hợp các hoạt động đơn giản, chúng ta có thể phải viết những đoạn mã dài và phức tạp. Điều này có thể ảnh hưởng đến khả năng đọc nhưng đảm bảo rằng các lập trình viên nhập chính xác những gì phải làm.

* 1. Khái quát thư viện hỗ trợ đồ họa Java Swing
     1. Khái niệm

Swing trong Java là một bộ công cụ Giao diện Người dùng Đồ họa (GUI) bao gồm các thành phần GUI. Swing cung cấp một bộ widget và gói phong phú để tạo ra các thành phần GUI tinh vi cho các ứng dụng Java

* + 1. So sánh giữa Java Swing và AWT

Bảng so sánh giữa AWT và Java Swing

|  |  |  |
| --- | --- | --- |
| **Các tiêu chí so sánh** | **AWT** | **Java Swing** |
| Nền tảng phụ thuộc | Phụ thuộc vào nền tảng; dựa vào các thành phần hệ diều hành gốc. | Nền tảng độc lập; cung cấp giao diện nhất quán trên các nền tảng |
| Nhẹ và nặng | Sử dụng các thành phần nặng gắn liền với GUI của nền tảng gốc; có thể kém hiệu quả hơn | Sử dụng các thành phần nhẹ, hoàn toàn dựa trên Java và hiệu quả hơn. |
| Xem và cảm nhận | Kế thừa giao diện của nền tảng gốc, có khả năng dẫn đến sự không nhất quán. | Cung cấp giao diện nhất quán, có thể tùy chỉnh thông qua các tùy chọn “Giao diện và cảm nhận (L&F). |
| Hiệu quả | Có thể có những hạn chế về hiệu suất do các thành phần phụ thuộc vào nền tảng. | Nói chung cấp hiệu suất tốt hơn do các thành phần nhẹ. |
| Tùy biến | Tùy chọn, tùy chỉnh hạn chế; dựa vào khả năng của nền tảng gốc | Khả năng tùy biến cao, với sự hỗ trợ rộng rãi để tạo các thành phần giao diện người dùng tùy chỉnh |
| Các thành phần bổ sung | Cung cấp một bộ giới hạn các thành phần GUI cơ bản | Cung cấp nhiều thành phần giao diện người dùng hơn, bao gồm bảng, cây và ngăn theo thẻ. |
| Phổ biến | Cũ hơn và ít phổ biến hơn trong phát triển Java hiện đại | Được áp dụng rộng rãi để tạo ra các ứng dụng GUI Java. |

Chương 2:  
CÁC NỀN TẢNG, MÔ HÌNH VÀ THƯ VIỆN SỬ DỤNG

2.1 IDE

Netbeans [2] là Môi trường phát triển tích hợp (IDE) mã nguồn mở. Nó được sử dụng để phát triển các ứng dụng với Java, PHP, C++, HTML và nhiều ngôn ngữ lập trình khác. Các ứng dụng được phát triển bằng cách sử dụng các mô-đun trong Java. NetBeans có thể chạy trên mọi hệ điều hành như Windows, MacOS, Linux, v.v.

2.2 Cơ sở dữ liệu

**Sử dụng PhpMyAdmin, là phần mềm mã nguồn mở được viết bằng ngôn ngữ PHP giúp quản trị cở sở dữ liệu MySQL thông qua giao diện web**.

**MySQL là một hệ thống quản trị cơ sở dữ liệu mã nguồn mở (gọi tắt là RDBMS) hoạt động theo mô hình client-server**. Với **RDBMS** là viết tắt của **Relational Database Management System**. **MySQL** được **tích hợp apache**, **PHP**. **MySQL** quản lý dữ liệu thông qua các cơ sở dữ liệu. Mỗi cơ sở dữ liệu có thể có nhiều bảng quan hệ chứa dữ liệu. **MySQL** cũng có cùng một cách truy xuất và mã lệnh tương tự với ngôn ngữ SQL. MySQL được phát hành từ thập niên 90s.

Một số tính năng chung thường được sử dụng trên phpMyAdmin:

* Quản lý user(người dùng): thêm, xóa, sửa(phân quyền).
* Quản lý cơ sở dữ liệu: tạo mới, xóa, sửa, thêm bảng, hàng, trường, tìm kiếm đối tượng.
* Nhập xuất dữ liệu(Import/Export): hỗ trợ các định dạng SQL, XML và CSV.
* Thực hiện các truy vấn MySQL, giám sát quá trình và theo dõi.
* Sao lưu và khôi phục(Backup/Restore): Thao tác thủ công.

**Điểm yếu trong việc sao lưu dữ liệu của phpMyAdmin.**

Dù có nhiều ưu điểm song phpMyAdmin vẫn khó tránh khỏi một vài điểm yếu cố hữu. Đặc biệt, trong việc sao lưu dữ liệu thủ công sẽ không có một vài tính năng cần thiết.

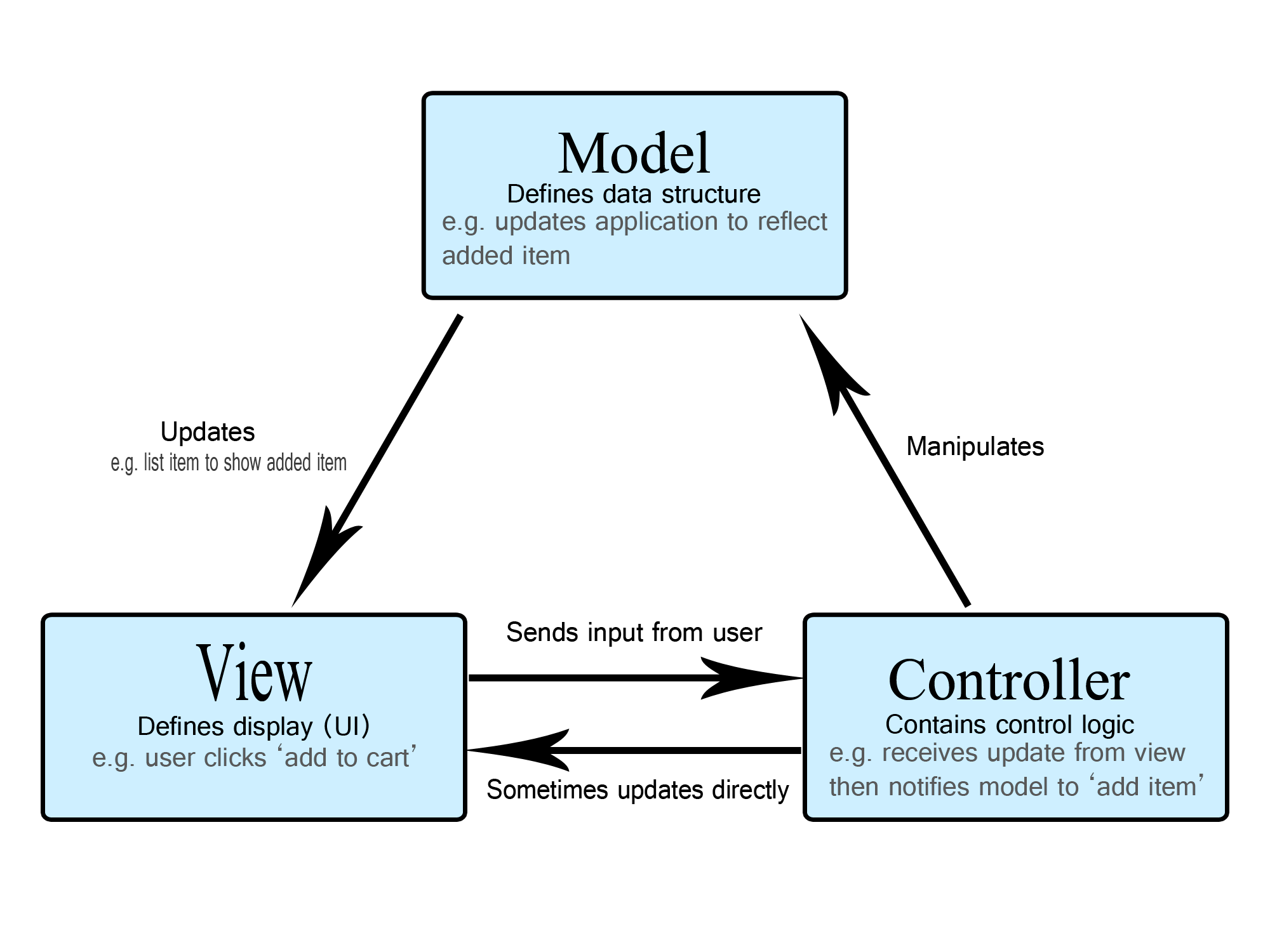
* Scheduling(sao lưu tự động theo lịch đặt trước): Một tính năng khá phổ biến ở những công cụ quản trị cơ sở dữ liệu.
* Storage media support(hỗ trợ lưu trữ các phương tiện truyền thông): phpMyAdmin chỉ cho phép lưu các bản sao lưu vào các local drive có sẵn trên hệ thống, qua hộp thoại Save as của trình duyệt.

2.3 Mô hình MVC

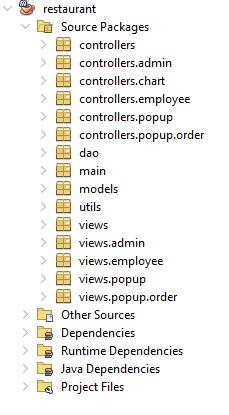
MVC (Model-View-Controller) [3] là một mẫu trong thiết kế phần mềm thường được sử dụng để triển khai giao diện người dùng, dữ liệu và logic điều khiển. Nó nhấn mạnh sự tách biệt giữa logic nghiệp vụ và hiển thị của phần mềm. Sự "tách biệt các mối quan tâm" này mang lại sự phân công lao động tốt hơn và cải thiện việc bảo trì. Một số mẫu thiết kế khác dựa trên MVC, chẳng hạn như MVVM (Model-View-Viewmodel), MVP (Model-View-Presenter) và MVW (Model-View-Whatever).

Ba phần của mẫu thiết kế phần mềm MVC có thể được mô tả như sau:

* Model: Quản lý dữ liệu và logic nghiệp vụ.
* View: Xử lý bố cục và hiển thị.
* Controller: Định tuyến các lệnh đến mô hình và xem các bộ phận.



1. Hình 2.3. Minh họa mô hình MVC



2. Hình 2.3.1. Triển khai mô hình MVC trong dự án

2.4 Các nền tảng và thư viện khác

2.3.1 Xampp

**XAMPP** [4] là một phần mềm cho phép **giả lập** môi trường server hosting ngay trên máy tính của bạn, cho phép bạn **chạy demo website** mà **không**cần phải mua[hosting](https://www.thegioididong.com/hoi-dap/hosting-la-gi-co-may-loai-kinh-nghiem-chon-mua-hosting-1337771)hay[VPS](https://www.thegioididong.com/hoi-dap/vps-la-gi-dung-de-lam-gi-co-uu-nhuoc-diem-nhu-the-nao-1337214). Chính vì vậy, XAMPP hay được phục vụ cho hoạt động học tập giảng dạy thực hành và phát triển web.

**XAMPP** được viết tắt của **X**+**Apache**+**MySQL**+**PHP**+**Perl** vì nó được tích hợp sẵn [Apache](https://vi.wikipedia.org/wiki/Apache_(HTTP)), [MySQL](https://www.thegioididong.com/game-app/huong-dan-cach-tai-cai-dat-mysql-ban-moi-nhat-chi-tiet-tung-1299084), [PHP](https://vi.wikipedia.org/wiki/PHP), [FTP server](https://www.thegioididong.com/hoi-dap/ftp-la-gi-3-dieu-co-ban-ban-can-biet-ve-giao-thuc-ftp-1339006), Mail Server. Còn X thể hiện cho sự đa nền tảng của XAMPP vì nó có thể dùng được cho 4 hệ điều hành khác nhau: [Windows](https://www.thegioididong.com/phan-mem/windows-10-home-32-bit-64-bit-all-languages-kw9-0), [MacOS](http://www.thegioididong.com/hoi-dap/he-dieu-hanh-macos-la-gi-956642), [Linux](https://www.thegioididong.com/hoi-dap/he-dieu-hanh-linux-la-gi-uu-nhuoc-diem-cua-he-dieu-hanh-1312530) và [Solaris](https://vi.wikipedia.org/wiki/Solaris_(h%E1%BB%87_%C4%91i%E1%BB%81u_h%C3%A0nh)).

2.3.2 FlatLaf

FlatLaf là Giao diện đa nền tảng mã nguồn mở hiện đại dành cho các ứng dụng máy tính để bàn Java Swing.

Nó trông gần như phẳng (không có bóng hoặc độ dốc), sạch sẽ, đơn giản và thanh lịch. FlatLaf đi kèm với các chủ đề Sáng, Tối, IntelliJ và Darcula, chia tỷ lệ trên màn hình HiDPI và chạy trên Java 8 hoặc mới hơn.

Giao diện được lấy cảm hứng rất nhiều từ các chủ đề Darcula và IntelliJ từ IntelliJ IDEA 2019.2+ và sử dụng hầu hết các màu sắc và biểu tượng giống nhau.

2.3.3 Imgscalr

Thư viện này cung cấp các phương thức để thao tác với hình ảnh trong Java, đặc biệt là thay đổi kích thước hình ảnh.

2.3.4 MySQL Connector/J

JDBC (Java Database Connectivity) là một chuẩn API (Application Program Interface) cho phép kết nối các chương trình viết bởi Java với các hệ quản trị cơ sở dữ liệu (MySQL, MS SQL, Postgre SQL, Oracle, DB2…)

JDBC chỉ là một tập các interface, các định nghĩa, thông báo lỗi, đặc tả chứ không phải là thư viện. Với mỗi hệ quản trị cơ sở dữ liệu ta sẽ có một cài đặt JDBC riêng cho nó, ví dụ JDBC cho MySQL, JDBC cho MS SQL …

Các thành phần của JDBC

* **DriverManager**: Dùng để quản lý danh sách các **Driver**(database drivers).
* **Driver**: Dùng để liên kết các kết nối tới cơ sở dữ liệu, điều khiển các liên kết.
* **Connection**: Biểu thị kết nối tới cơ sở dữ liệu. Dùng để tạo ra Statement, PreparedStatement và CallableStatement.
* **Statement, PreparedStatement, CallableStatement**: Chứa lệnh SQL gửi tới cơ sở dữ liệu để thực thi.
* **ResultSet** – biểu diễn một tập kết quả trong cơ sở dữ liệu tạo ra bởi việc sử dụng một câu lệnh SQL là SELECT.
* **SQLException** – một lớp xử lý lỗi ngoại lệ chứa các lỗi truy cập cơ sở dữ liệu.

2.3.5 JFreeChart

FreeChart là một FrameWork mã nguồn mở cho ngôn ngữ lập trình Java, cho phép tạo ra nhiều loại biểu đồ tương tác và không tương tác. Sau đây là hướng dẫn tích hợp biểu đồ JFreeChart trong Java Swing.

2.3.6 Jcalendar

JCalendar là một công cụ chọn ngày trong Java để chọn ngày theo đồ họa. JCalendar bao gồm một số Java Beans khác, JDayChooser, JMonthChooser và JYearChooser. Tất cả các loại Beans này đều có thuộc tính miền địa phương, cung cấp một số biểu tượng (Màu 16×16, Màu 32×32, Mono 16×16 và Mono 32×32) và trình chỉnh sửa thuộc tính miền địa phương của riêng chúng. Vì vậy, chúng có thể dễ dàng được sử dụng trong các trình xây dựng GUI. Ngoài ra, một phần của gói này còn có JDateChooser, một Bean bao gồm IDateEditor (để chỉnh sửa ngày trực tiếp) và một nút để mở JCalendar để chọn ngày.

Chương 3:  
CÁC CHỨC NĂNG CỦA CHƯƠNG TRÌNH VÀ PHÂN TÍCH THIẾT KẾ

3.1 Các chức năng của chương trình

**Chức năng phân quyền:** Phân quyền của nhân viên và quản lý

**Quản lý nhân viên:** Cho phép thêm, sửa, xóa thông tin về nhân viên và lưu vào cơ sở dữ liệu

**Quản lý hàng hóa:** Cho phép thêm, xóa, sửa thông tin về các loại sản phẩm, hàng hóa, bao gồm giá, và mô tả. Quản lý hàng hóa bao gồm 2 phần:

* Quản lý loại món: Chức năng này sẽ cho phép phân loại của loại hàng hóa đang kinh doanh trong cửa hàng, và có thể thêm, xóa, loại hàng hóa ở chức năng này.
* Quản lý món ăn: Liệt kê đa dạng các món ăn đang kinh doanh của cửa hàng và có phân loại. Ví dụ: Hotdog (Hotdog phô mai kéo sợi, Hotdog xúc xích chà bông,…)

**Quản lý đặt hàng:** Chia ra từng mục quản lý khác nhau.

* Quản lý bàn: Cho phép thêm, xóa, sửa các bàn và hiển thị trạng thái của bàn trong quán (Đang sử dụng hoặc trống).
* Quản lý khách hàng: Cho phép thêm, xóa, sửa thông tin của khách hàng.
* Quản lý đơn hàng: Cho phép người quản lý biết được đơn hàng nào đang được phục vụ, đã thanh toán, và đơn hàng nào đang được giao, trạng thái của tất cả các hóa dơn, ngày giờ, người thanh toán, người tạo hóa đơn, tổng số tiền thanh toán. Tài khoản có quyền quản lý thì có thể xem tất cả các hóa đơn được tạo bởi bất kỳ nhân viên, quản lý nào, nhưng tài khoản có quyền Nhân viên chỉ có thể xem được hóa đơn mà mình tạo nên.
* Quản lý giao hàng: Bao gồm thông tin của khách hàng, địa chỉ khách hàng, tên nhân viên giao hàng, thời gian bắt đầu và kết thúc của một đơn giao, trạng thái của đơn giao.

**Thống kê:** Hiển thị tổng doanh thu của quán và số hóa đơn dựa vào một khoảng thời gian được chọn, bao gồm cả việc hiển thị số nhân viên, số khách hàng của quán, việc thống kê được chia là hai phần.

* Thống kê nhân viên: Bao gồm thống kê phiên làm việc và thống kê số nhân viên, hiển tị ngày giờ nhân viên làm việc và hiển thị thông số của từng nhân viên, số hóa đơn tạo được trong ca làm việc và số tiền thưởng khi tạo được một hóa đơn.
* Thống kê doanh thu: Sẽ bao gồm việc thống kê doanh thu theo nhân viên (Nhân viên tạo được tổng bao nhiêu hóa đơn và số tiền bán ra) và thống kê theo doanh thu sản phẩm (Sẽ có bao nhiêu sản phẩm được bán ra trong khoảng thời gian và tổng tiền).

**Chức năng chấm công:** Khi người dùng đăng nhập, hệ thống sẽ bắt được sự kiện hoạt động của người dùng đó, và chương trình sẽ bắt đầu đếm thời gian làm việc của nhân viên, khi nhân viên đăng xuất tài khoản hoặc tắt chương trình thì bộ đếm thời gian cũng sẽ dừng lại và thống kê được phiên làm việc của nhân viên đó.

**Tìm kiếm:** Ở mỗi chức năng quản lý như Quản lý nhân viên, quản lý bàn, quản lý món ăn, quản lý đơn đặt hàng, quản lý khách hàng,… đều có chức năng tìm kiếm dựa trên nhiều tiêu chí khác nhau.

**Thiết lập:** Ở chức năng này, sẽ hiển thị thông tin cá nhân, thời gian làm việc, chức năng đổi mật khẩu và xem lại lịch sử đăng nhập.

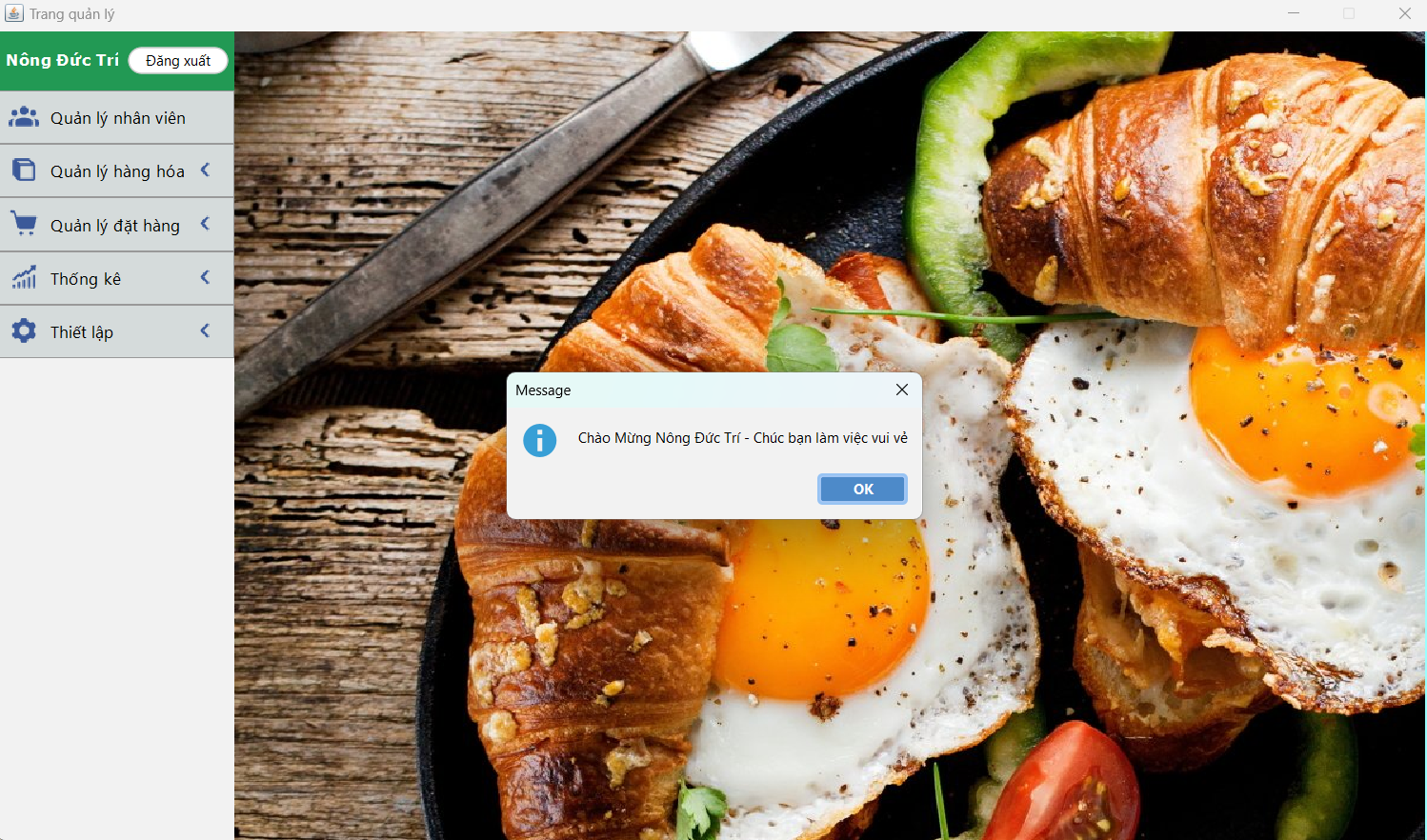
3.2 Phân tích thiết kế

Giao diện người dùng của ứng dụng quản lý bán hàng được thiết kế để đảm bảo tính thân thiện và dễ sử dụng. Với sự kết hợp giữa tính thẩm mỹ và hiệu suất, giao diện người dùng này, về cơ bản, cung cấp các tiện lợi để quản lý thông tin.

**ADMIN**

3.2.1 Màn hình chính

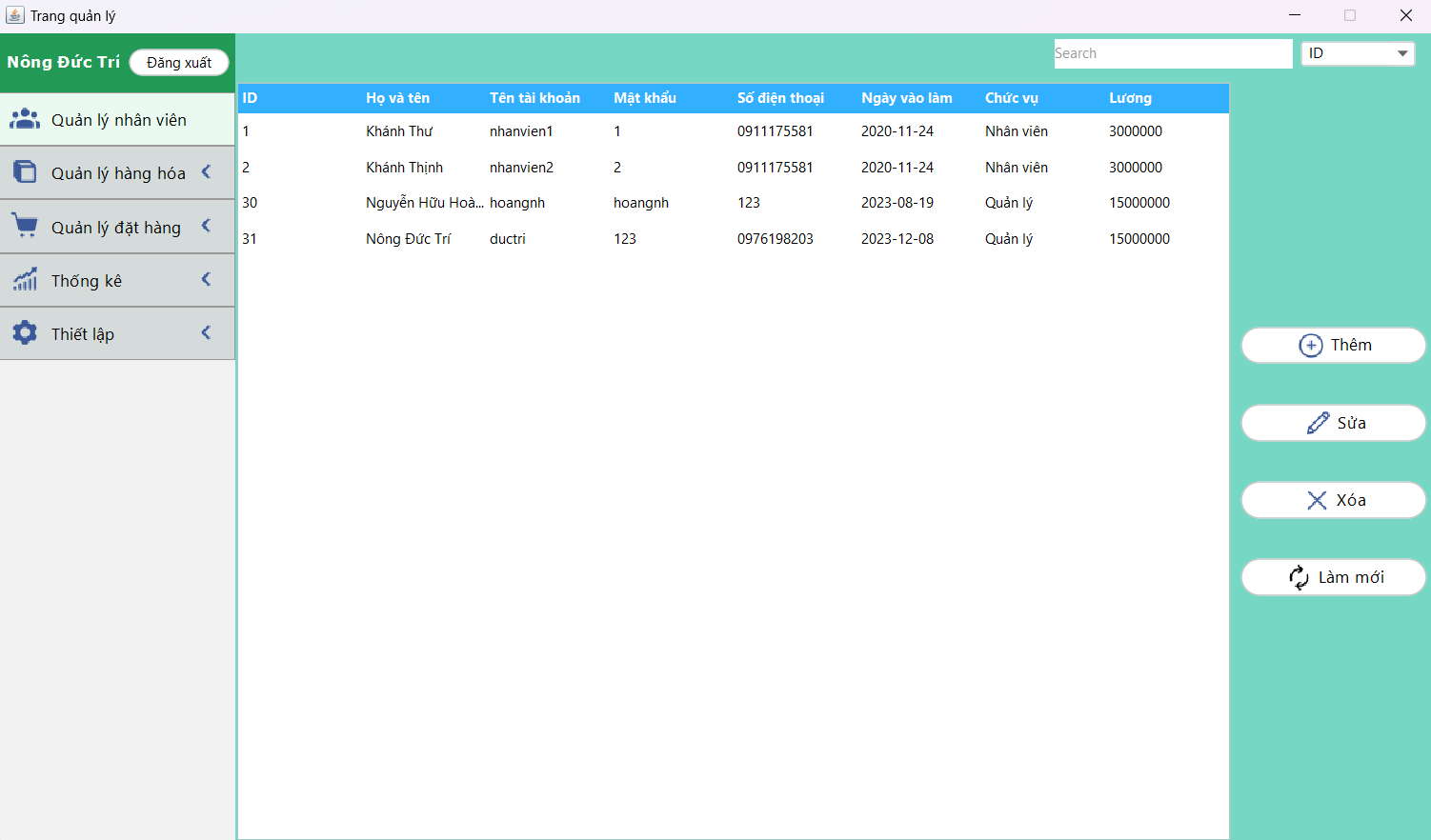
Màn hình chính là trung tâm điều hành của ứng dụng, là nơi có thể truy cập các tính năng quan trọng và xem thông tin tổng quan về tình trạng của doanh nghiệp. Trên màn hình này, người dùng sẽ dễ dàng truy cập vào các tính năng như quản lý thông tin khách hàng.



Hình 3.2.1. Màn hình chính

3.2.2 Quản lý nhân viên

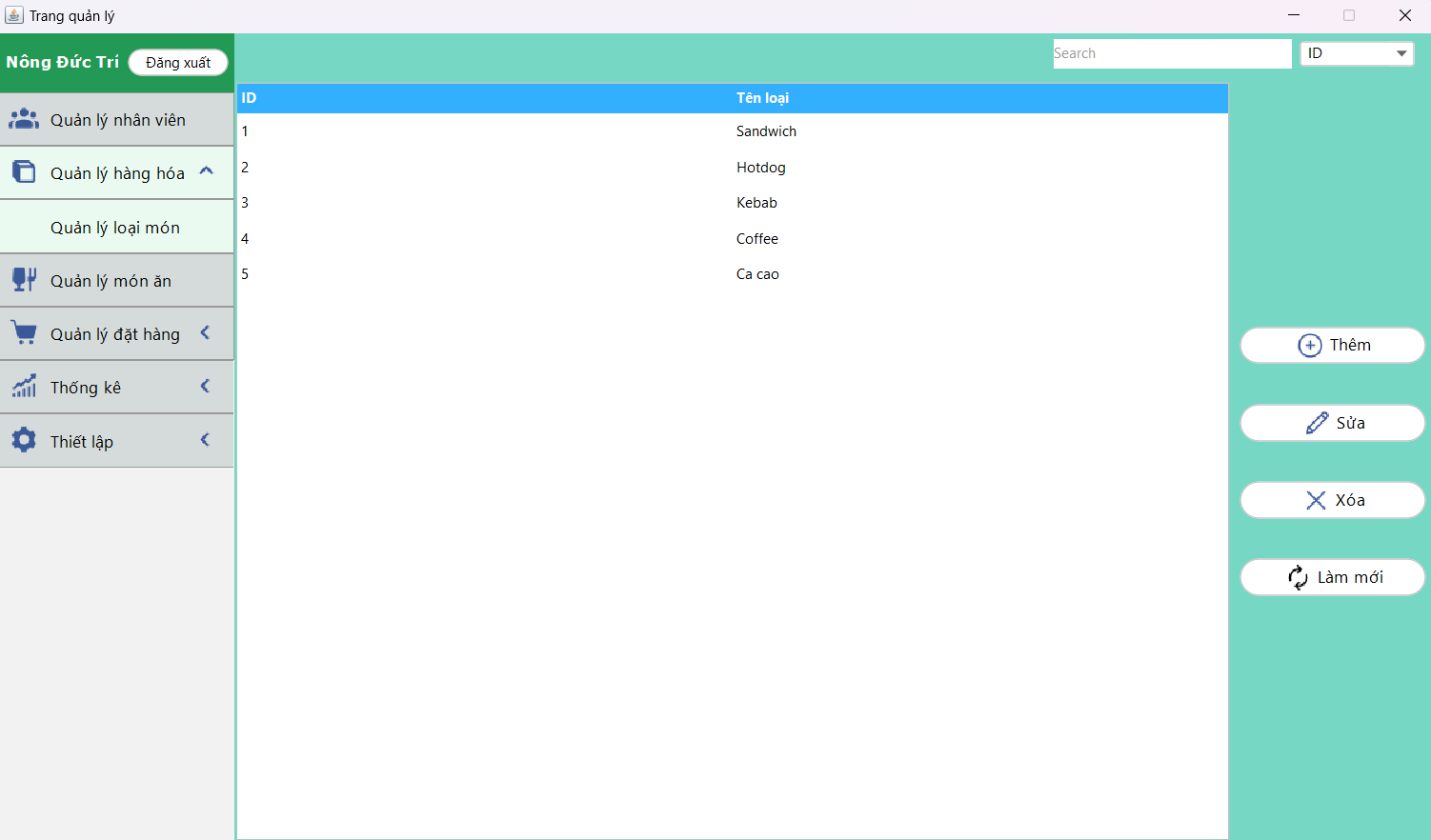
Tại đây, admin có thể thêm, sửa, xóa thông tin cá nhân của nhân viên, bao gồm tên, tài khoản, số điện thoại, thông tin cá nhân.



3.Hình 3.2.2. Màn hỉnh quản lý nhân viên

3.2.3 Quản lý loại hàng hóa

Màn hình quản lý loại hàng hoá giúp người dùng dễ dàng chọn loại món ăn, quản lý món ăn. Trong phần này admin cũng có thể thêm, sửa, xoá các loại món ăn. Thêm phần mô tả và hình ảnh cho từng loại món ăn.



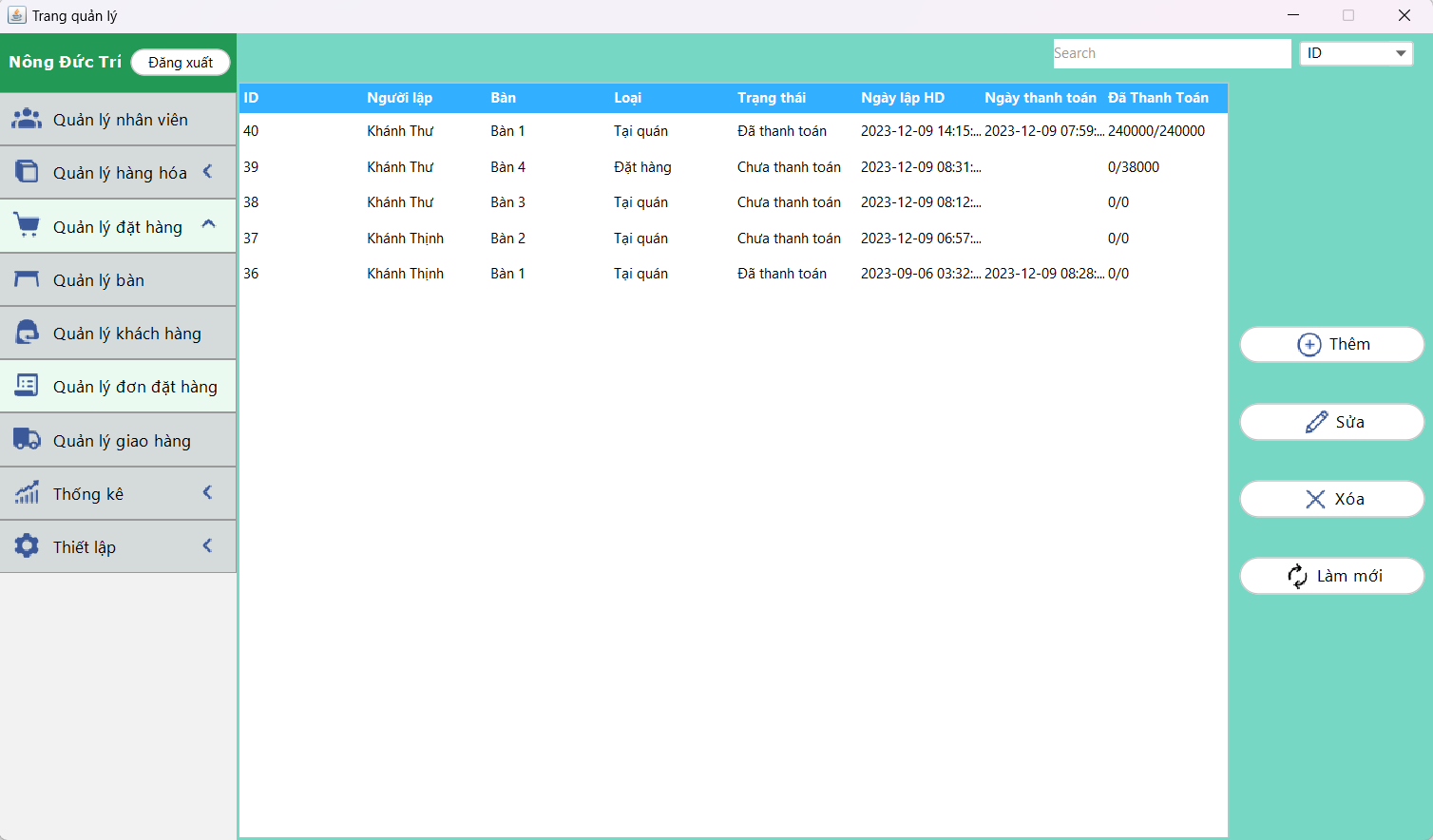
4.Hình 3.2.3. Màn hình quản lý loại món ăn



5.Hình 3.2.3. Màn hình quản lý tất cả món ăn

3.2.4 Quản lý đặt hàng

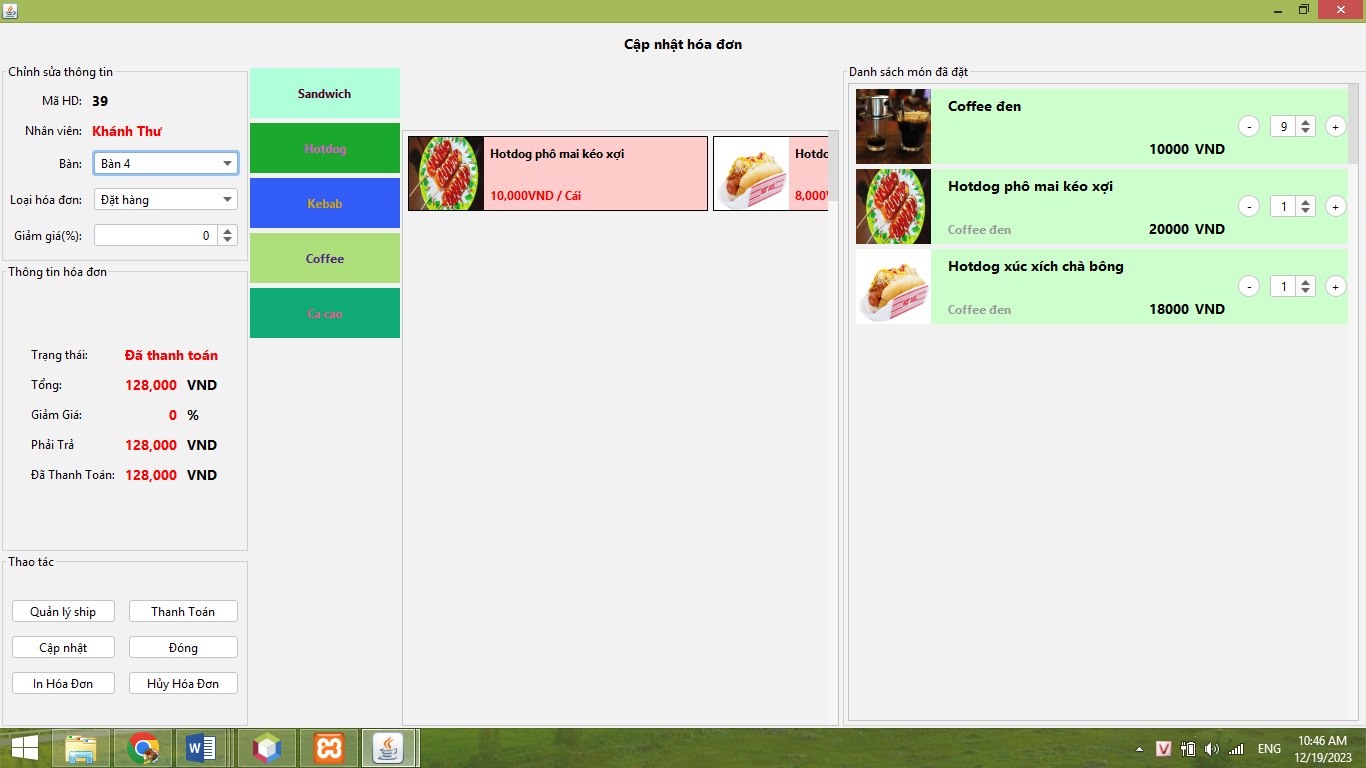
Trang quản lý đặt hàng cho phép người thêm, sửa, và xóa thông tin về bàn phục vụ, khách hàng, đơn đặt hàng. Mỗi trang quản lý có mô tả và tình trạng hiện tại. Thông tin được cập nhật tại quản lý đặt hàng cũng sẽ được cập nhật về phía quản lý bán hàng để người dùng xem được thông tin, trạng thái mặt hàng, dịch vụ nào hiện đang phục vụ và giao đi, thông tin của khách hàng trong hệ thống và phục vụ cho mục đích bán hàng.



6.Hình 3.2.4. Màn hình quản lý tất cả đơn đặt hàng

3.2.5 Chỉnh sửa hóa đơn

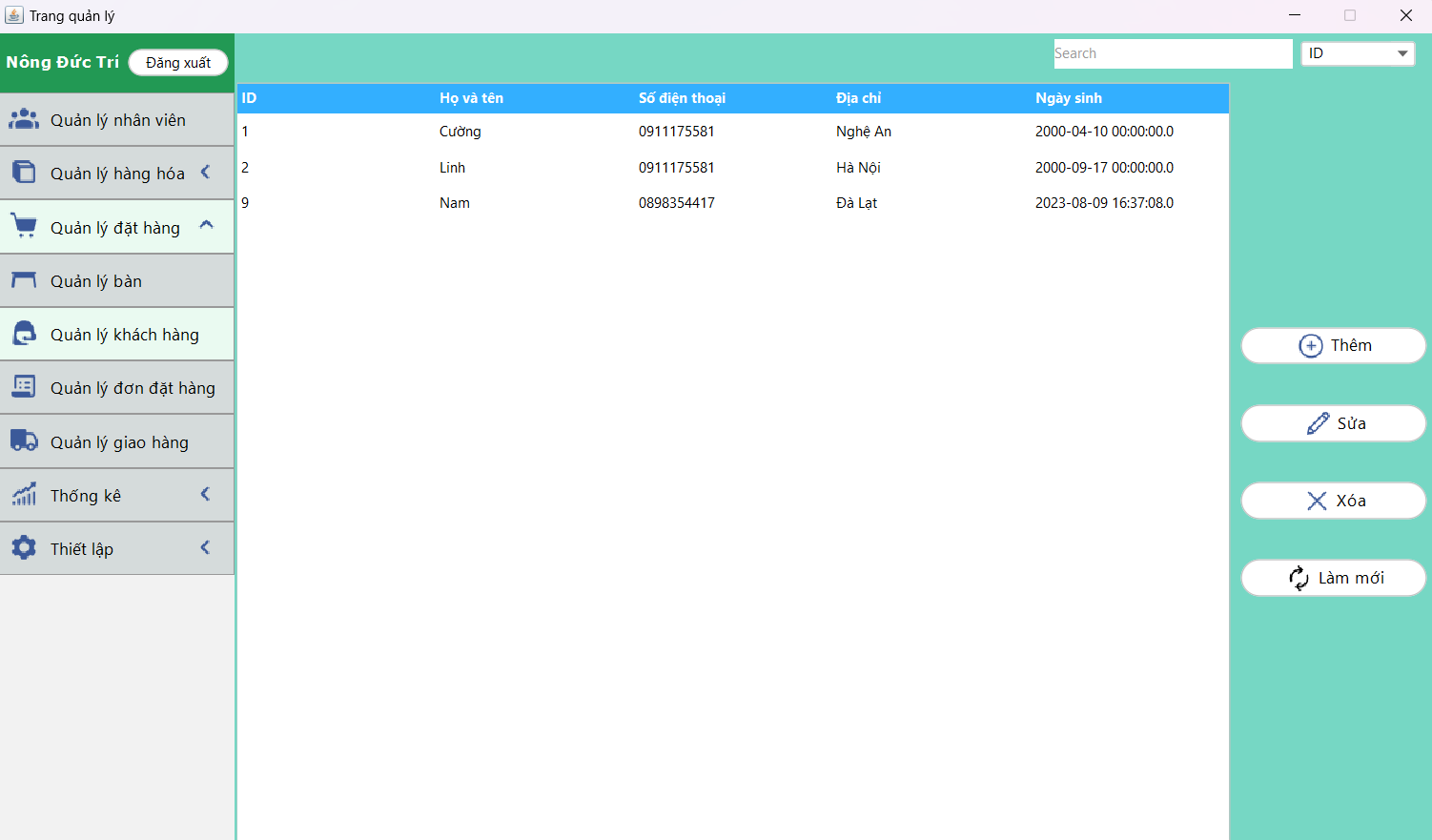
Tại đây, nhân viên có thể xem thông tin của hóa đơn được chọn, người lập hóa đơn, trạng thái hóa đơn, và ở trung tâm của màn hình sẽ có phần chọn món cho hóa đơn, khi nhấn chọn thì danh sách món được chọn sẽ hiển thị ở bên phải màn hình và sẽ thống kê tạm thời tổng giá của hóa đơn đó, và trong giao diện này, người dùng có thể hủy hóa đơn, cập nhật, in hóa đơn, quản lý giao hàng,…



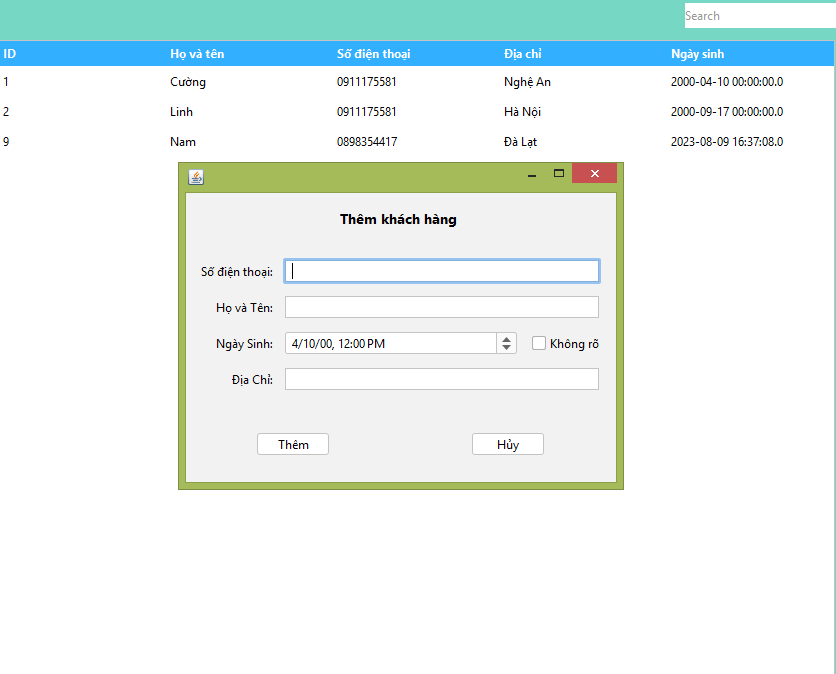
7. Hình 3.2.5. Giao diện chỉnh sửa hóa đơn

3.2.6 Quản lý khách hàng

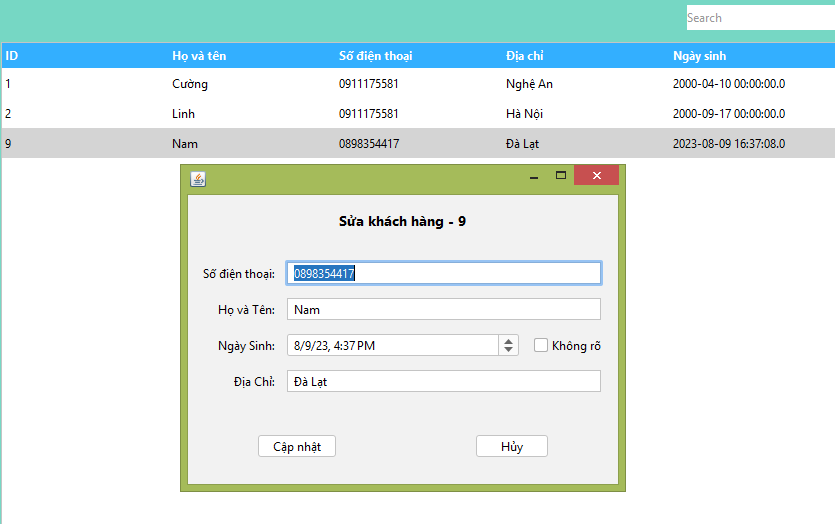
Tại đây, nhân viên có thể thêm một khách hàng mới, sửa thông tin của khách hàng, xóa khách hàng và danh sách tổng khách hàng sẽ hiển thị như hình dưới.



8. Hình 3.2.6. Giao diện tổng danh sách khách hàng



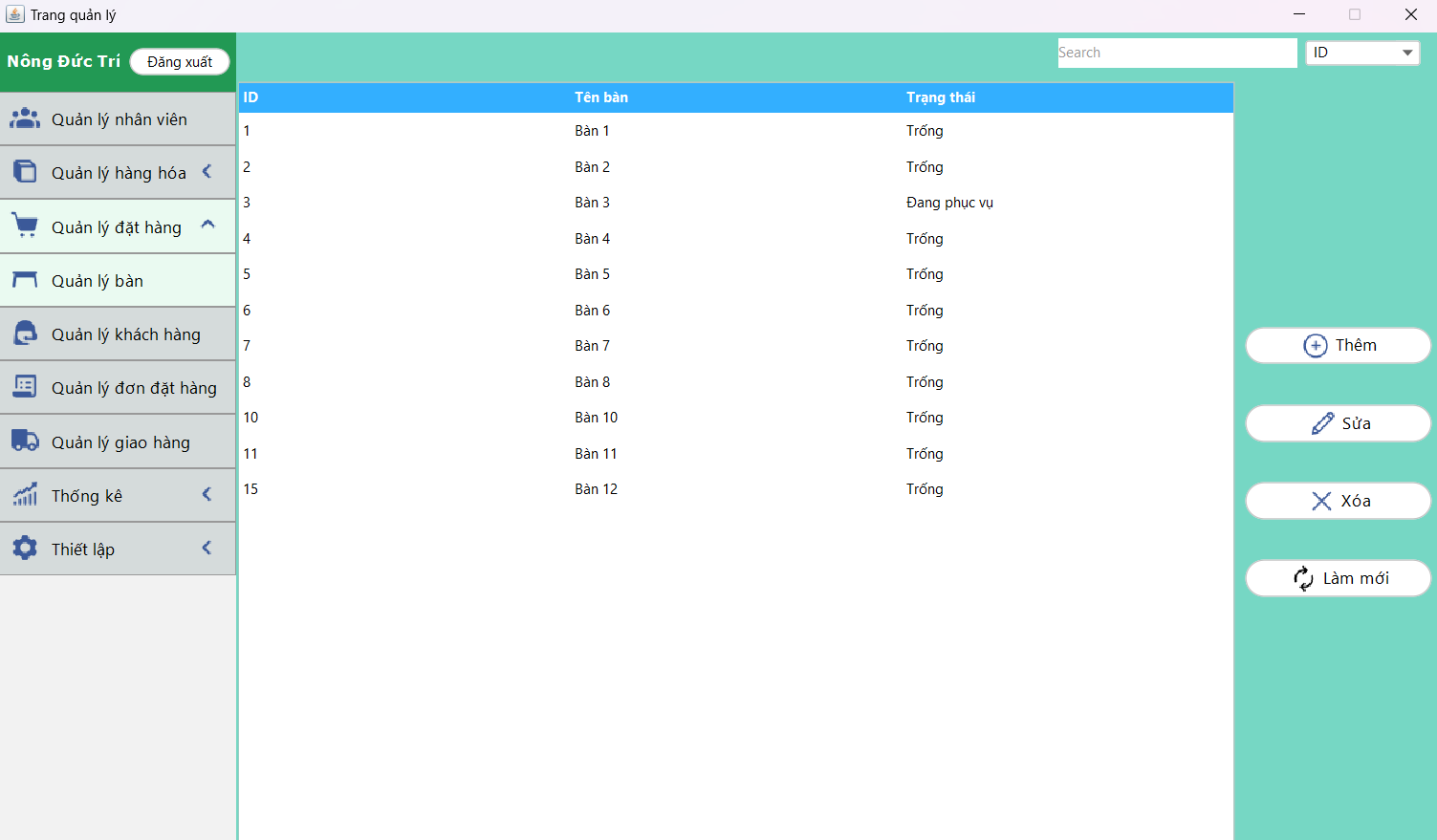
9. Hình 3.2.6.2. Cửa sổ bật lên khi nhấn nút thêm khách hàng



10. Hình 3.2.6.3. Cửa sổ bật lên khi nhấn sửa khách hàng

3.2.7 Quản lý bàn

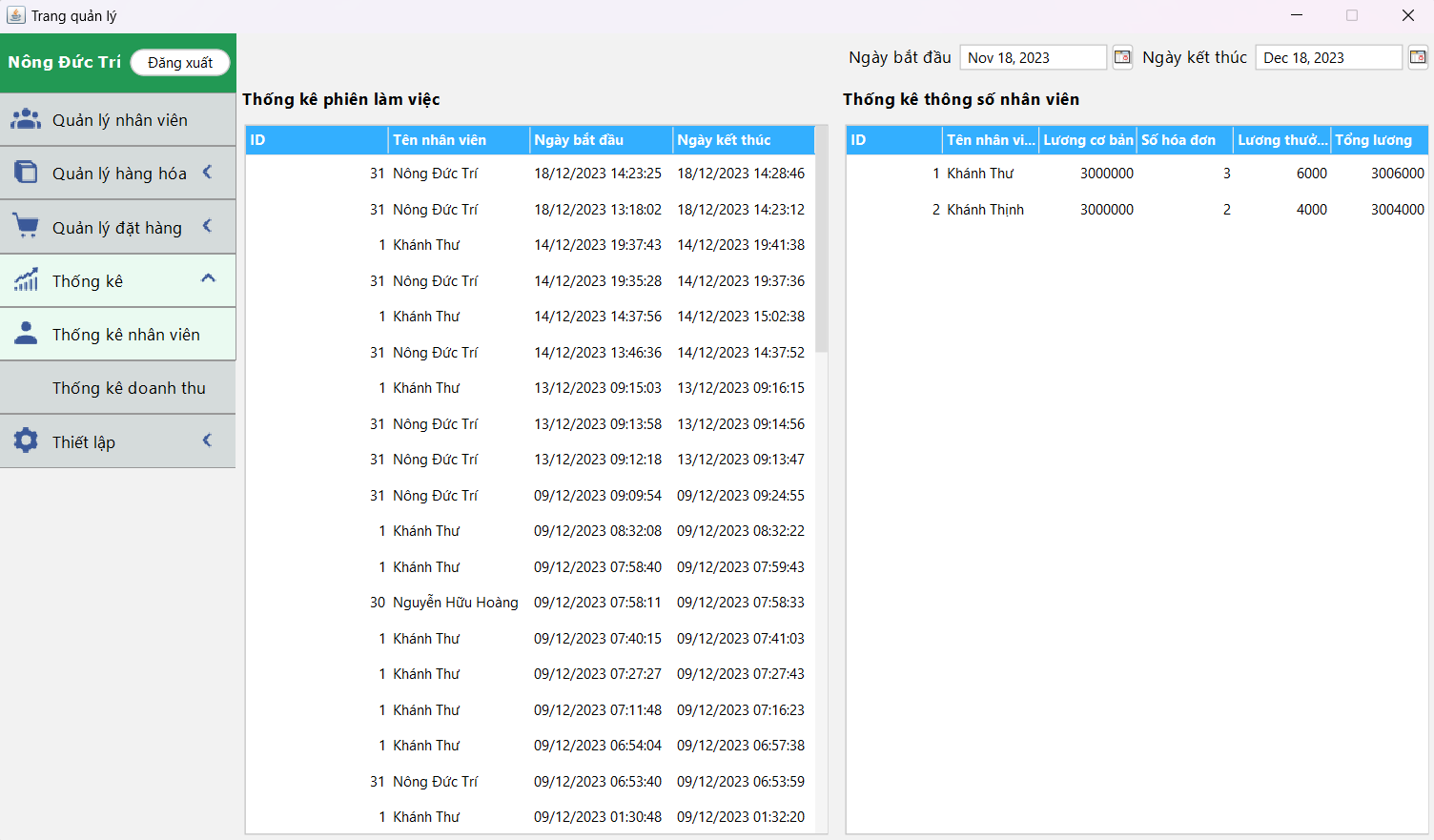
Tại đây, có thể thao tác với các chức năng như thêm bàn mới, xóa bàn và hiển thị tổng số bàn vào danh sách, đồng thời có thể xem được trạng thái của bàn (Trống hoặc đang phục vụ) và có thể tìm kiếm dựa vào điều kiện.



11. Hình 3.2.7. Giao diện quản lý bàn

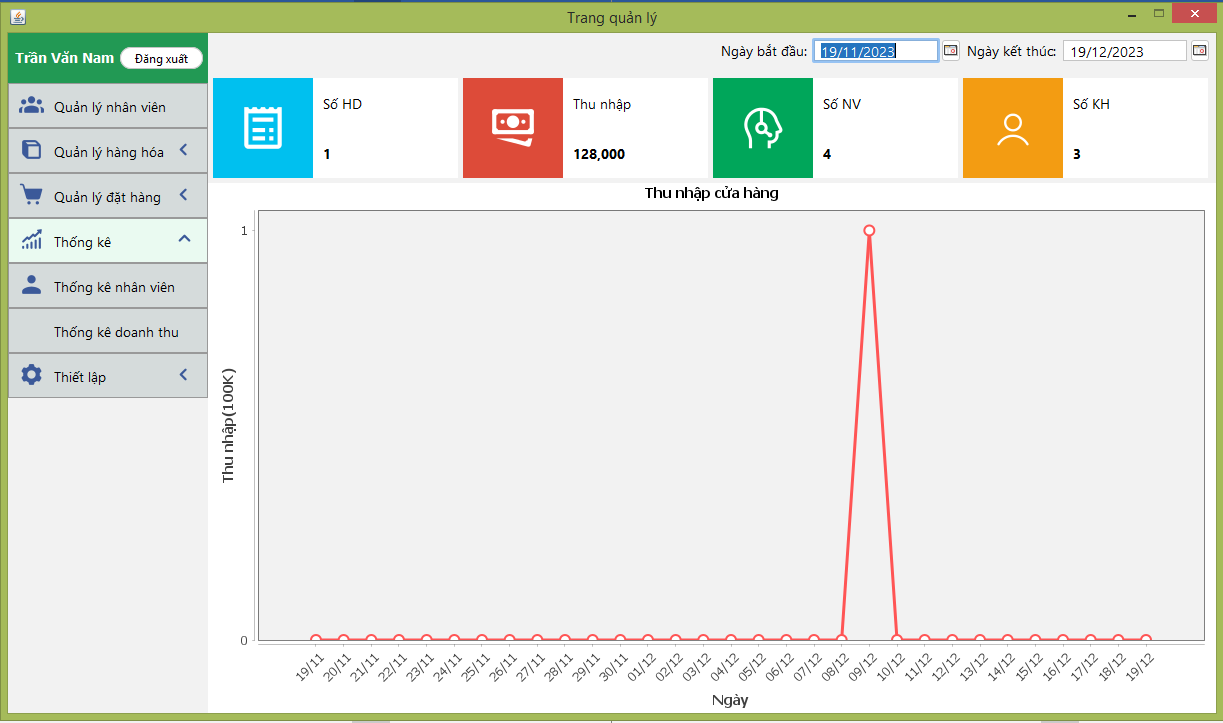
3.2.8 Thống kê

Trang thống kê cho phép cho phép người dùng xem được số hoá đơn bán trong ngày, doanh thu, số nhân viên và khách hàng của quán. Ngoài ra còn hiển thị mục thống kê nhân viên, xem được thống kê phiên làm việc của nhân viên và thông số nhân viên. Phần thống kê doanh thu hiển thị doanh thu theo nhân viên và doanh thu theo sản phẩm.



12. Hình 3.2.8. Giao diện thống kê

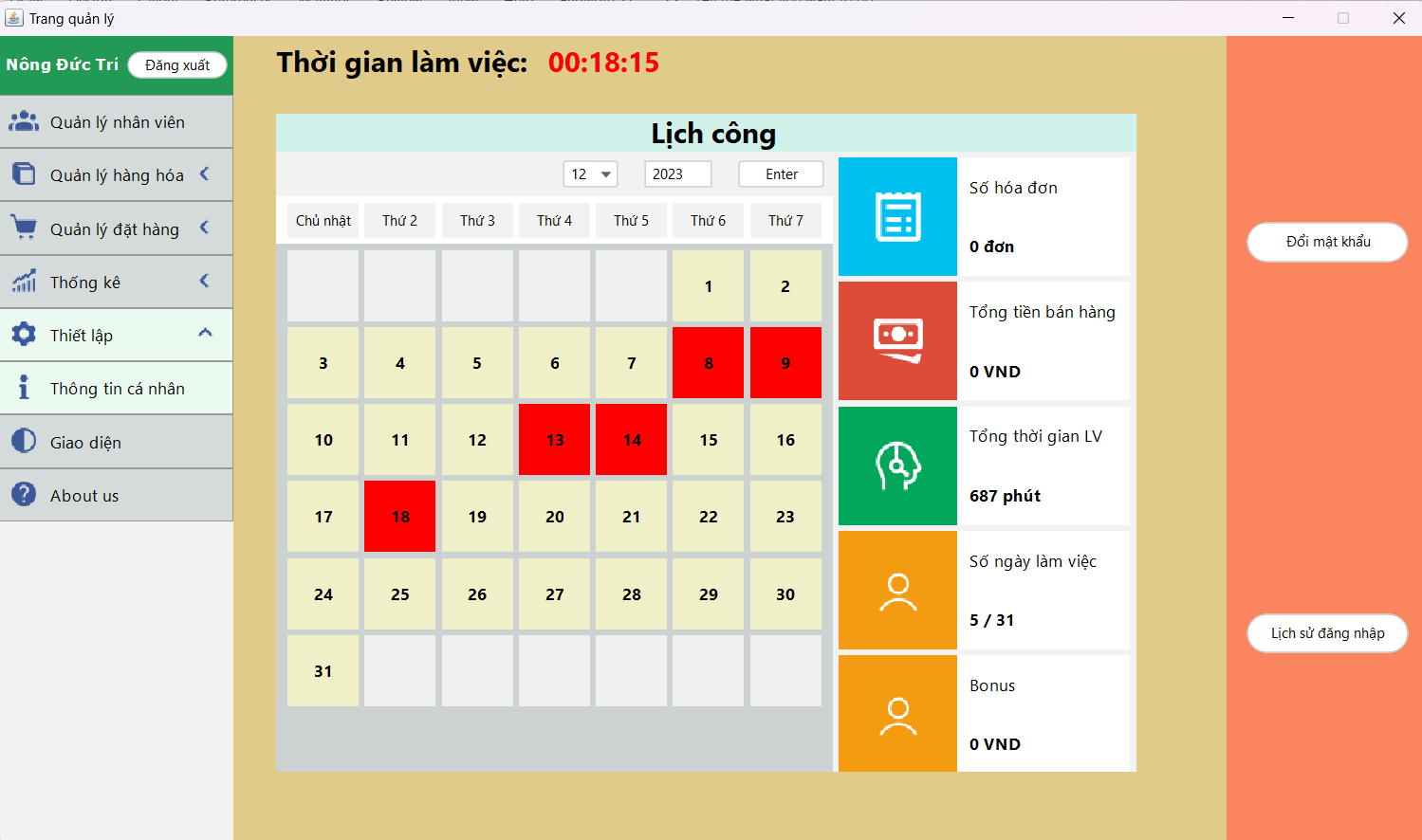
Giao diện tổng thống kê sẽ thể hiện được tổng số hóa đơn được tạo của cửa hàng, tổng số nhân viên, tổng số khách hàng tổng thu nhập và biểu đồ đường thể hiện tình trạng kinh doanh của cửa hàng (Sử dụng JfreeChart), và có thể xem các thông số trên dựa vào một mốc thời gian nhất định.



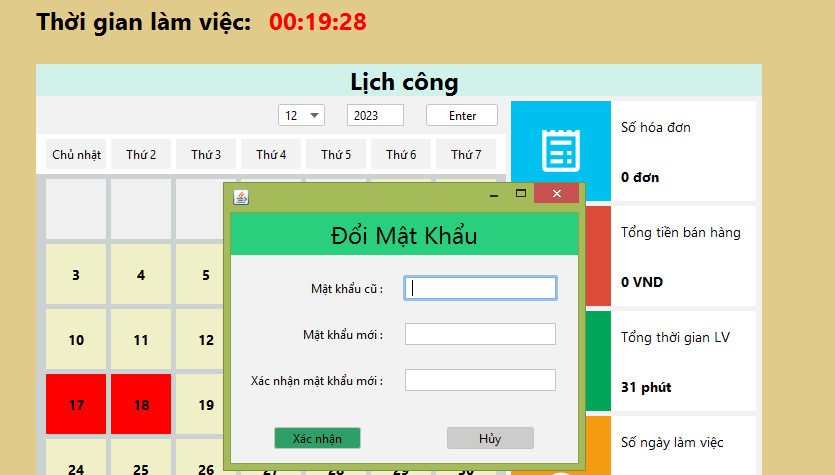
13. Hình 3.2.8.1. Giao diện tổng thống kê

3.2.9 Thiết lập

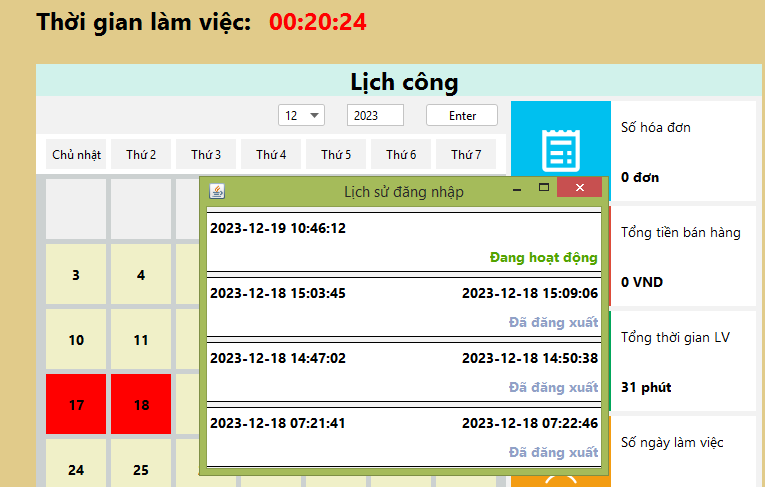
Trang thiết lập cho phép người dùng có thể đổi mật khẩu và xem lịch sử giao dịch. Trang còn hiển thị thời gian làm việc, ngày làm việc và số hoá đơn, tổng tiền ngày hôm đó,…



14. Hình 3.2.9. Giao diện thông tin cá nhân



15. Hình 3.2.9.1. Cửa số bật lên khi đổi mật khẩu

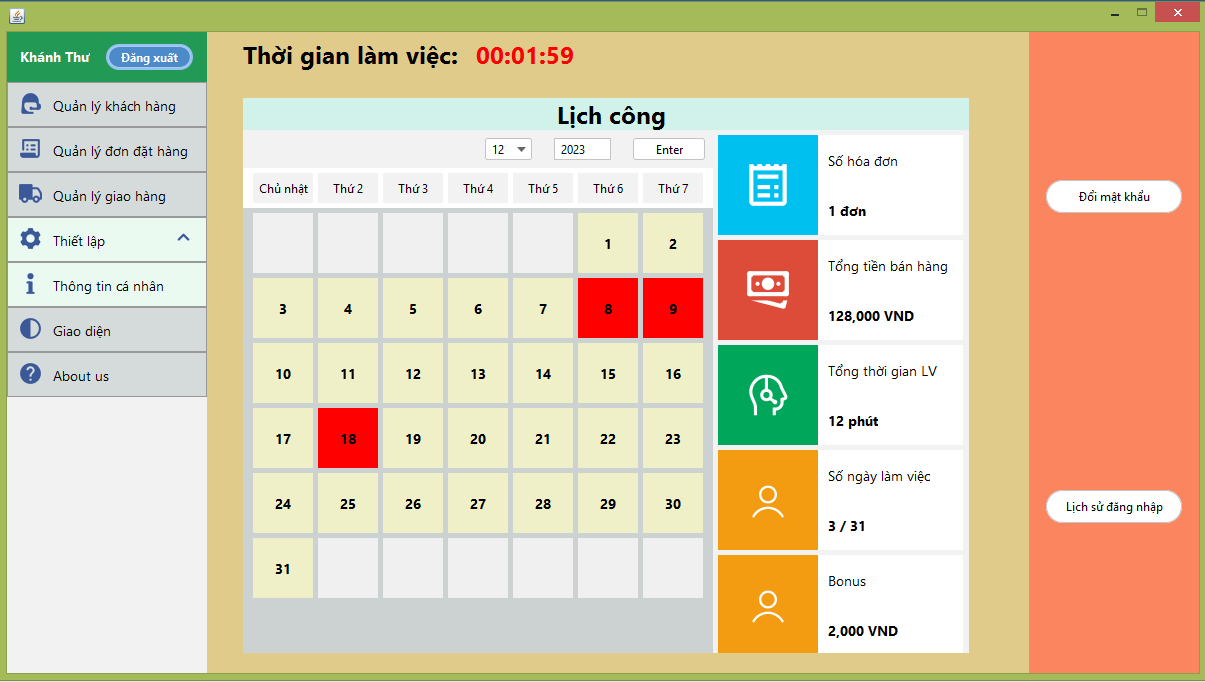


16. Hình 3.2.9.2. Cửa số bật lên khi xem lịch sử đăng nhập

**NHÂN VIÊN**

Các chức năng liệt kê ở trên thuộc về Quản lý, và nhân viên sẽ hạn chế quyền hơn quản lý, nhân viên chỉ có thể Quản lý khách hàng, quản lý hóa đơn mà mình tạo, quản lý giao hàng, xem thông tin cá nhân.





17. Hình 3.2.10. Tất cả các quyền của nhân viên

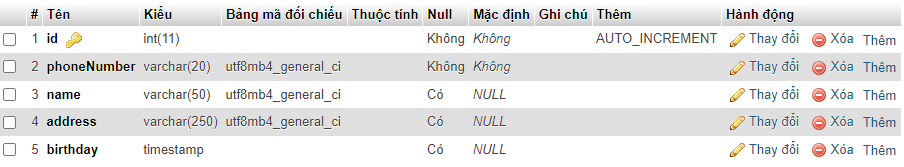
Chương 4:  
QUÁ TRÌNH THỰC HIỆN VÀ MÃ NGUỒN

4.1 Xây dựng dữ liệu

Bộ dữ liệu được xây dựng bằng MySQL và sử dụng PHPMyAdmin để quản lý bằng giao diện web.

4.1.2 Các bảng dữ liệu

**Bảng customer:** Dùng để lưu trữ khách hàng và có các thông tin khách hàng cơ bản.



**Bảng employee:** Dùng để lưu trữ thông tin của nhân viên và các trạng thái.



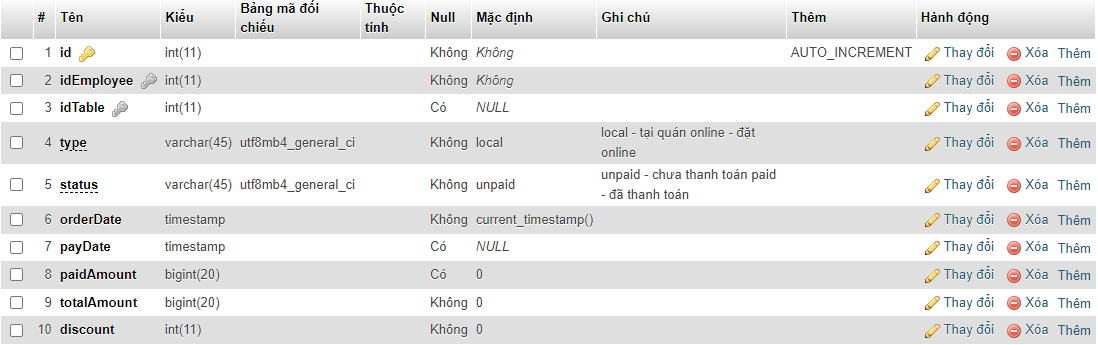
**Bảng food\_category:** Dùng để lưu trữ loại của món ăn.



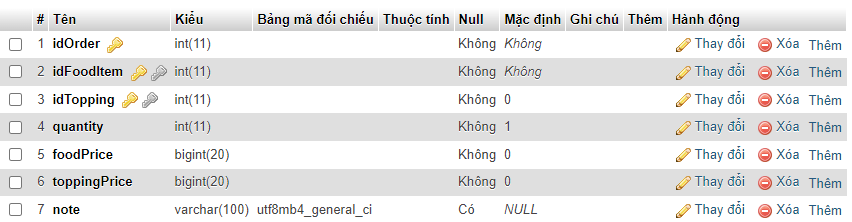
**Bảng food\_item:** Dùng để lưu trữ chi tiết của một món ăn, tham chiếu khóa ngoại đến loại món ăn.



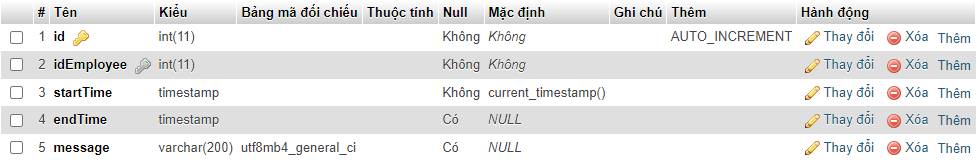
**Bảng order:** Dùng để lưu trữ thông tin của một hóa đơn, tham chiếu khóa ngoại để lấy tên của người lập hóa đơn, số bàn.



**Bảng order\_item:** Dùng để lưu trữ chi tiết của một hóa đơn, tham chiếu đến mã hóa đơn cần xem chi tiết, mã món ăn có trong hóa đơn, mã của topping.



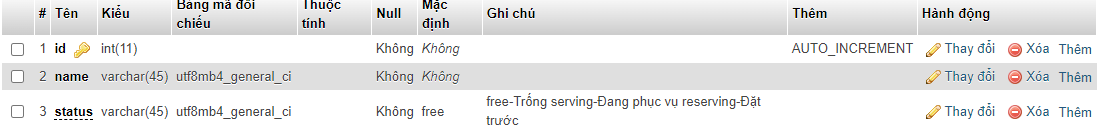
**Bảng session:** Dùng để lưu trữ phiên đăng nhập của một tài khoản mỗi khi đăng nhập thành công, tham chiếu khóa ngoại của id nhân viên.



**Bảng shipment:** Dùng để quản lý chức năng giao hàng, bảng này chứa những thông tin cơ bản như thời gian bắt đầu, giá, thời gian kết thúc, trạng thái,… và tham chiếu khóa ngoại để lấy id của nhân viên.



**Bảng table:** Dùng để quản lý các bàn.



18. Hình 4.1.2. Các bảng dữ liệu

4.1.3 Lược đồ quan hệ



19. Hình 4.1.3. Quan hệ giữa các bảng.

4.2 Xây dựng mã nguồn

Dưới đây là tổng hợp những đoạn mã xử lý những sự kiện cơ bản của chương trình, bắt đầu từ việc kết nối cơ sở dữ liệu và tiếp đến các truy vấn giúp thay đổi dữ liệu như thêm, xóa, sửa, và lấy danh sách trong dữ liệu để hiển thị lên và sự kiện tìm kiếm một phần tử trong dữ liệu dựa vào tiêu chí.

Đây là đoạn mã minh họa cho những sự kiện đã nêu trước đó, ví dụ này được lấy ở lớp Customer, tương tự cho các lớp xử lý khác, cũng sẽ có những sự kiện như thêm, xóa, sửa, hiển thị, tìm kiếm, và sẽ có những câu truy vấn phù hợp.

4.2.1 Kết nối cơ sở dữ liệu

**[code]**

*database.driver\_class=****com.mysql.cj.jdbc.Driver***

*database.jdbc=****mysql***

*database.host=****localhost***

*database.port=****3308***

*database.username=****root***

*database.password=*

*database.name=****restaurant***

*path.mysqldump=****C:\\xampp\\mysql\\bin\\mysqldump.exe***

*path.mysql=****C:\\xampp\\mysql\\bin\\mysql.exe***

*path.backup=****E:\\Data\\backup***

**[code]**

Đầu tiên cấu hình kết nối tới một cơ sở dữ liệu MySQL bằng cách sử dụng JDBC (Java Database Connectivity). Dưới đây là một phân tích chi tiết về từng thuộc tính:

* database.driver\_class=com.mysql.cj.jdbc.Driver:

Đây là tên của lớp driver JDBC sử dụng để kết nối đến cơ sở dữ liệu MySQL. Trong trường hợp này, đây là driver cho MySQL.

* database.jdbc=mysql:

Thuộc tính này xác định giao thức JDBC được sử dụng. Trong trường hợp này, giao thức là MySQL.

* database.host=localhost:

Xác định máy chủ cơ sở dữ liệu. Ở đây, cơ sở dữ liệu đang chạy trên máy chủ cục bộ, được đại diện bằng địa chỉ "localhost".

* database.port=3308:

Port mà cơ sở dữ liệu MySQL lắng nghe. Trong trường hợp này, port là 3308.

* database.username=root:

Tên người dùng sẽ được sử dụng để kết nối đến cơ sở dữ liệu MySQL. Ở đây, tên người dùng là "root".

* database.password=:

Mật khẩu cho người dùng kết nối đến cơ sở dữ liệu. Trong trường hợp này, mật khẩu là trống, có thể đồng nghĩa với việc không có mật khẩu.

* database.name=restaurant:

Tên của cơ sở dữ liệu muốn kết nối đến. Ở đây, tên cơ sở dữ liệu là "restaurant".

* path.mysqldump=C:\xampp\mysql\bin\mysqldump.exe:

Đường dẫn tới công cụ mysqldump, một công cụ dùng để sao lưu cơ sở dữ liệu MySQL.

* path.mysql=C:\xampp\mysql\bin\mysql.exe:

Đường dẫn tới công cụ mysql, một công cụ dùng để thực hiện các truy vấn và tương tác với cơ sở dữ liệu MySQL.

* path.backup=E:\Data\backup:

Đường dẫn nơi muốn lưu trữ các bản sao lưu của cơ sở dữ liệu. Trong trường hợp này, đường dẫn là "E:\Data\backup".

**[code]**

*public class Database {*

*private static final LoadConfig cfg = LoadConfig.getIntanse();*

*private static Database instance = null;*

*private Connection conn = null;*

*private Database() {*

*try {*

*String connect Property=* ***"useUnicode=true&useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&serverTimezone=UTC"****;*

*String host = cfg.getProperty("****database.host****"),*

*port = cfg.getProperty("****database.port****"),*

*user = cfg.getProperty("****database.username****"),*

*password = cfg.getProperty("****database.password****"),*

*name = cfg.getProperty("****database.name****");*

*Class.forName(cfg.getProperty("****database.driver\_class****"));*

*String url = String.format("jdbc:%s://%s:%s/%s?%s", cfg.getProperty("database.jdbc"), host, port, name, connectProperty);*

*this.conn = DriverManager.getConnection(url, user, password);*

*System.out.println("****Kết nối cơ sở dữ liệu thành công****!");*

*} catch (ClassNotFoundException e) {*

*System.out.println("****Chưa cài driver mysql****!");*

*System.out.println(e.toString());*

*System.exit(0);*

*} catch (SQLException e) {*

*System.out.println("****Kết nối cơ sở dữ liệu thất bại****:");*

*System.out.println(e.toString());*

*System.exit(0);*

*}*

*}*

**[code]**

Hàm tạo Database:

Hàm tạo này được sử dụng để thiết lập kết nối đến cơ sở dữ liệu MySQL. Trong quá trình này, nó sử dụng thông tin cấu hình đọc từ LoadConfig. Các bước bao gồm:

* Tạo chuỗi kết nối (url) bằng cách sử dụng các thông số như host, port, tên cơ sở dữ liệu từ cấu hình.
* Sử dụng Class.forName để tải lớp driver JDBC cho MySQL.
* Sử dụng DriverManager.getConnection để thiết lập kết nối với cơ sở dữ liệu sử dụng các thông tin về URL, tên người dùng và mật khẩu từ cấu hình.

4.2.2 Các truy vấn

**a. Hiển thị danh sách**

**[code]**

*@Override*

*public ArrayList<Customer>* ***getAll****() throws SQLException {*

*ArrayList<Customer> customers = new ArrayList<>();*

*Statement statement = conn.createStatement();*

*String query =* ***"SELECT \* FROM `customer`";***

*ResultSet rs = statement.executeQuery(query);*

*while (rs.next()) {*

*Customer customer = Customer.getFromResultSet(rs);*

*customers.add(customer);*

*}*

*return customers;*

*}*

**[code]**

Tạo danh sách ArrayList:

* Tạo một đối tượng ArrayList để chứa các đối tượng Customer kết quả từ truy vấn.

Tạo đối tượng Statement:

* Sử dụng đối tượng Statement để tạo một câu lệnh SQL không tham số.

Tạo câu truy vấn SQL:

* Sử dụng câu truy vấn SQL đơn giản để lấy tất cả các bản ghi từ bảng customer.

Thực hiện truy vấn SELECT:

* Sử dụng statement.executeQuery(query) để thực hiện truy vấn SELECT và nhận một đối tượng ResultSet chứa kết quả.

Lặp qua kết quả và tạo đối tượng Customer:

* Sử dụng một vòng lặp while để lặp qua mỗi bản ghi trong ResultSet. Dùng phương thức Customer.getFromResultSet(rs) để tạo đối tượng Customer từ dữ liệu của bản ghi và thêm vào danh sách customers.

Trả về danh sách kết quả:

* Trả về danh sách customers chứa tất cả các đối tượng Customer từ bảng.

**b. Sự kiện thêm**

**[code]**

*@Override*

*public void* ***save****(Customer t) throws SQLException {*

*if (t == null) {*

*throw new SQLException("****Customer rỗng****");*

*}*

*String query =* ***"INSERT INTO `customer` (`phoneNumber`, `name`, `address`, `birthday`) VALUES (?, ?, ?, ?)"****;*

*PreparedStatement stmt = conn.prepareStatement(query);*

*stmt.setNString(1, t.getPhoneNumber());*

*stmt.setNString(2, t.getName());*

*stmt.setNString(3, t.getAddress());*

*stmt.setTimestamp(4, t.getBirthday());*

*int row = stmt.executeUpdate();*

*}*

**[code]**

Kiểm tra đối tượng rỗng:

* Trước khi thực hiện thêm vào cơ sở dữ liệu, hàm kiểm tra xem đối tượng Customer (t) có giá trị là null hay không. Nếu là null, nó sẽ ném một ngoại lệ SQLException với thông báo "Customer rỗng".

Câu truy vấn SQL:

* Câu truy vấn SQL được sử dụng để thêm một bản ghi mới vào bảng customer. Câu truy vấn này sử dụng các tham số thay thế (?) để tránh tình trạng SQL injection.

PreparedStatement:

* Sử dụng PreparedStatement để thực hiện câu truy vấn SQL. Điều này giúp tối ưu hóa hiệu suất và tránh các vấn đề an ninh liên quan đến SQL injection.

Thiết lập giá trị tham số:

* Sử dụng các phương thức setNString và setTimestamp của PreparedStatement để thiết lập giá trị cho các tham số trong câu truy vấn. Dữ liệu được lấy từ các trường của đối tượng Customer (t).

Thực hiện truy vấn và nhận kết quả:

* Sử dụng executeUpdate để thực hiện truy vấn SQL và trả về số dòng bị ảnh hưởng. Biến row nhận giá trị này.

**c. Sự kiện xóa**

**[code]**

@Override

public void **delete**(Customer t) throws SQLException {

PreparedStatement stmt = conn.prepareStatement**("DELETE FROM `customer` WHERE `id` = ?"**);

stmt.setInt(1, t.getId());

stmt.executeUpdate();

}

**[code]**

Câu truy vấn SQL:

* Sử dụng câu truy vấn SQL để xóa dữ liệu khỏi bảng customer. Câu truy vấn này sử dụng một tham số (?) để chỉ định điều kiện xóa, trong trường hợp này là id.

PreparedStatement:

* Sử dụng PreparedStatement để tạo câu truy vấn có tham số và tránh các vấn đề an ninh liên quan đến SQL injection.

Thiết lập giá trị tham số:

* Sử dụng stmt.setInt(1, t.getId()) để thiết lập giá trị cho tham số trong câu truy vấn. Giá trị này lấy từ trường id của đối tượng Customer được chuyển vào phương thức.

Thực hiện truy vấn xóa:

* Sử dụng stmt.executeUpdate() để thực hiện truy vấn xóa.

**d. Sự kiện sửa**

**[code]**

*@Override*

*public void* ***update****(Customer t) throws SQLException {*

*if (t == null) {*

*throw new SQLException("Customer rỗng");*

*}*

*String query = "UPDATE `customer` SET `phoneNumber` = ?, `name` = ?, `address` = ?, `birthday` = ? WHERE `id` = ?";*

*PreparedStatement stmt = conn.prepareStatement(query);*

*stmt.setNString(1, t.getPhoneNumber());*

*stmt.setNString(2, t.getName());*

*stmt.setNString(3, t.getAddress());*

*stmt.setTimestamp(4, t.getBirthday());*

*stmt.setInt(5, t.getId());*

*int row = stmt.executeUpdate();*

*}*

**[code]**

Kiểm tra đối tượng rỗng:

* Trước khi thực hiện cập nhật, hàm kiểm tra xem đối tượng Customer (t) có giá trị là null hay không. Nếu là null, nó sẽ ném một ngoại lệ SQLException với thông báo "Customer rỗng".

Câu truy vấn SQL:

* Sử dụng câu truy vấn SQL để cập nhật dữ liệu trong bảng customer. Câu truy vấn này sử dụng các tham số thay thế (?) để tránh tình trạng SQL injection.

PreparedStatement:

* Sử dụng PreparedStatement để tạo câu truy vấn có tham số và tránh các vấn đề an ninh liên quan đến SQL injection.

Thiết lập giá trị tham số:

* Sử dụng các phương thức setNString và setTimestamp của PreparedStatement để thiết lập giá trị cho các tham số trong câu truy vấn. Dữ liệu được lấy từ các trường của đối tượng Customer (t).

Thực hiện truy vấn cập nhật:

* Sử dụng stmt.executeUpdate() để thực hiện truy vấn cập nhật và trả về số dòng bị ảnh hưởng. Biến row nhận giá trị này.

**e. Sự kiện tìm kiếm**

**[code]**

*public ArrayList<Customer>* ***searchByKey****(String key, String word) throws SQLException {*

*ArrayList<Customer> customers = new ArrayList<>();*

*Statement statement = conn.createStatement();*

*String query =* ***"SELECT \* FROM `customer` WHERE " + key + " LIKE '%" + word + "%';"****;*

*ResultSet rs = statement.executeQuery(query);*

*while (rs.next()) {*

*Customer customer = Customer.getFromResultSet(rs);*

*customers.add(customer);*

*}*

*return customers;*

*}*

**[code]**

Tạo danh sách ArrayList:

* Tạo một đối tượng ArrayList để chứa các đối tượng Customer kết quả từ truy vấn.

Tạo đối tượng Statement:

* Sử dụng đối tượng Statement để tạo một câu lệnh SQL không tham số.

Tạo câu truy vấn SQL:

* Sử dụng câu truy vấn SQL để tìm kiếm các bản ghi trong bảng customer. Câu truy vấn này sử dụng các tham số động để tìm kiếm dựa trên một cột cụ thể và từ khóa được nhập vào.

Thực hiện truy vấn SELECT:

* Sử dụng statement.executeQuery(query) để thực hiện truy vấn SELECT và nhận một đối tượng ResultSet chứa kết quả.

Lặp qua kết quả và tạo đối tượng Customer:

* Sử dụng một vòng lặp while để lặp qua mỗi bản ghi trong ResultSet. Dùng phương thức Customer.getFromResultSet(rs) để tạo đối tượng Customer từ dữ liệu của bản ghi và thêm vào danh sách customers.

Trả về danh sách kết quả:

* Trả về danh sách customers chứa tất cả các đối tượng Customer tìm thấy dựa trên từ khóa và cột xác định.

Chương 5:  
KIỂM THỬ VÀ KẾT QUẢ

5.1 Kết quả đạt được

Trong quá trình phát triển ứng dụng quản lý bán hàng sử dụng thư viện Java – Swing, dự án đã đạt được một số kết quả quan trọng sau.

* Giao diện người dùng đáng tin cậy: Giao diện người dùng trực quan và dễ sử dụng cho ứng dụng quản lý bán hàng. Giao diện người dùng cho phép người dùng thêm, cập nhật, và xóa và tìm kiếm thông tin về các hóa đơn, bàn, nhân viên,… cũng như thông tin về khách hàng.
* Quản lý thông tin của hóa: Ứng dụng cho phép quản lý thông tin của các hóa đơn, nhân viên có thể thêm mới hóa đơn, cập nhật, xóa và hiển thị được thông tin chi tiết của hóa đơn
* Quản lý thông tin của tất cả khía cạnh trong kinh doanh: Trong kinh doanh, ngoài việc quản lý hóa đơn, ứng dụng cần phải đáp ứng đủ được các khía cạnh khác trong kinh doanh như: Nhân viên, bàn, khách hàng, giao hàng, thống kê nhân viên, thống kê doanh thu nhân viên, thống kê tổng doanh thu của cửa hàng, theo dõi phiên làm việc của nhân viên.
* Liên kết với cơ sở dữ liệu: Ứng dụng kết nối với cơ sở dữ liệu MySQL để lưu trữ và truy xuất thông tin về kinh doanh của cửa hàng. Điều này giúp bảo vệ và duy trì dữ liệu một cách hiệu quả.
* Thử nghiệm và xác minh: Thực hiện kiểm thử ứng dụng để đảm bảo tính ổn định và hiệu quả. Ứng dụng đã được kiểm tra trước khi triển khai để đảm bảo nó hoạt động đúng như dự kiến.

5.2 Hạn chế

Mặc dù đã đạt được nhiều kết quả tích cực, ứng dụng quản lý bán hàng vẫn còn một số hạn chế:

* Giao diện hạn chế: Mặc dù mạnh mẽ, nhưng có thể có giới hạn trong việc tạo các giao diện người dùng phức tạp và đẹp mắt. Đôi lúc sẽ không thể khởi tạo trực tiếp khi ứng dụng xảy ra lỗi hoặc lỗi một giao diện không chứa bất kỳ phương thức hoặc biến nào.
* Khả năng mở rộng: Ứng dụng hiện tại chỉ tập trung vào quản lý bán hàng. Để phát triển ứng dụng hoàn chỉnh, có thể cần phải mở rộng tính năng và tích hợp nhiều khía cạnh hơn của quản lý bán hàng. Tham khảo một số doanh nghiệp đang có xu hướng phát triển và thế mạnh để biết thêm một số tính năng và sáng tạo mới trong ứng dụng quản lý bán hàng.
* Hiệu suất: Ứng dụng có thể cần tối ưu hóa để cải thiện hiệu suất và thời gian đáp ứng, đặc biệt khi số lượng khách hàng và đặt hàng lớn
* Hạn chế về tích hợp: Thư viện JFreeChart có thể có hạn chế trong việc khả năng truy cập với API và  Có thể có lỗi khi cố gắng nhập các lớp từ thư viện JFreeChart. [Ví dụ, khi cố gắng nhập org.jfree.chart.ChartPanel, bạn có thể nhận được lỗi "Loại org.jfreechart.ChartPanel không thể truy cập"](https://stackoverflow.com/questions/55386075/the-type-org-jfree-chart-jfreechart-is-not-accessible-after-adding-jfreechart-an).
* Hạn chế về đa nhiệm: Thư viện JFreeChart không được thiết kế để hỗ trợ đa luồng (multithreading) một cách tự nhiên. [Điều này có nghĩa là, nếu bạn cố gắng cập nhật dữ liệu biểu đồ từ nhiều luồng, bạn có thể gặp phải các vấn đề về đồng bộ hóa](https://github.com/jfree/jfreechart). Đây là một hạn chế quan trọng, đặc biệt là khi bạn muốn cập nhật biểu đồ theo thời gian thực hoặc xử lý dữ liệu lớn từ nhiều nguồn. [Trong những trường hợp như vậy, bạn cần phải tự quản lý việc đồng bộ hóa dữ liệu và cập nhật biểu đồ](https://github.com/jfree/jfreechart).

Chương 6:  
KẾT LUẬN

Trong dự án này, ứng dụng quản bán hàng cho phép quản lý thông tin về khách hàng và mặt hàng, quản lý bán hàng thông qua giao diện người dùng đáng tin cậy và kết nối với cơ sở dữ liệu MySQL.

Dự án đã đạt được một số kết quả quan trọng như sau:

* Xây dựng giao diện người dùng trực quan và dễ sử dụng cho việc quản lý bán hàng.
* Kết nối với cơ sở dữ liệu MySQL để lưu trữ và truy xuất thông tin.
* Thực hiện các chức năng quản lý cơ bản bao gồm thêm mới, cập nhật, xóa và hiển thị thông tin chi tiết.
* Kiểm tra ứng dụng để đảm bảo tính ổn định và hiệu quả.

Tuy nhiên, dự án cũng có một số hạn chế:

* Giao diện người dùng có thể hạn chế trong việc tạo các giao diện phức tạp và đẹp mắt.
* Cần cải thiện về tính bảo mật, đặc biệt trong việc quản lý đăng nhập và quản lý người dùng.
* Yêu cầu tối ưu hóa để cải thiện hiệu suất và thời gian đáp ứng, đặc biệt khi ứng dụng đối diện với một số lượng lớn dữ liệu.
* Mở rộng tính năng để đáp ứng các yêu cầu phức tạp hơn của quản lý bán hàng.

Như vậy, mặc dù còn nhiều hạn chế nhưng dự án đã cung cấp một cơ sở để phát triển ứng dụng quản lý bán hàng cơ bản. Dự án có tiềm năng để tiếp tục phát triển và mở rộng để đáp ứng nhu cầu của ngành dịch vụ. Cá nhân sinh viên đã học được rất nhiều trong quá trình phát triển dự án này, từ quá trình xây dựng một ứng dụng đến kỹ năng xây dựng ứng dụng, và hy vọng rằng nó có thể là một khởi đầu cho các ứng dụng quản lý phức tạp hơn trong tương lai.

TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | Stefik, Mark and Bobrow, Daniel G, "Object-oriented programming: Themes and variations," *AI magazine,* vol. 6, no. 4, pp. 40-40, 1975. |
| [2] | Timo Hynninen, Jussi Kasurinen, Ossi Taipale, "Framework for Observing the Maintenance Needs, Runtime Metrics and the Overall Quality-in-Use," *Journal of Software Engineering and Applications,* vol. 11, no. 4, pp. 50 - 62, 2018. |
| [3] | Mulla, Qauseen and Karajgi, Aslam and Quadri, S., "Stress Detection Using Social Media Interactions," pp. 306-310, 10 09 2023. |
| [4] | Joshi, Kapil and Kumar, Rajiv and Kumar, Anil and Reshi, Jagdeep and Sharma, Aditi and Dumka, Ankur, "A Framework Optimization in Social Media using Xampp: A Systematic Approach," in *2022 International Conference on Fourth Industrial Revolution Based Technology and Practices (ICFIRTP)*, 2022, pp. 1-4. |