**ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

🙤🙧🟍🙥🙦



**XÂY DỰNG PHẦN MỀM  
QUẢN LÝ BÁN HÀNG  
SỬ DỤNG CÔNG NGHỆ JAVA SWING**

|  |  |  |
| --- | --- | --- |
| **Sinh viên thực hiện:** | | |
| **STT** | **Họ tên** | **MSSV** |
| 1 | Đào Đăng Khoa | 20520017 |
| 2 | Vũ Tuấn Kiệt | 20520018 |
| 3 | Hoàng Hữu My | 20521029 |

**TP. HỒ CHÍ MINH – 02/2022**

**DANH MỤC BẢNG, HÌNH ẢNH**

[Hình 2‑1: Giao diện tab Khách hàng 2](#_Toc95893639)

[Hình 2‑2: Thông số kích thước các Panels 3](#_Toc95893640)

[Hình 2‑3: Thông số GridLayout của cụm JButtons chức năng 3](#_Toc95893641)

**PHÂN CÔNG NHIỆM VỤ**

| **STT** | **Thành viên** | **Nhiệm vụ** |
| --- | --- | --- |
| 1 | Đào Đăng Khoa | V1:   * Xây dựng Class ConnectDB. * Viết các chức năng thêm, xóa, sửa, load dữ liệu Sản phẩm. * Soạn thảo word   V2:   * Bổ sung các Methods kiểm tra rằng buộc định dạng SĐT, ngày ở các tab Khách Hàng, Nhân Viên. * Xây dựng frame login |
| 2 | Vũ Tuấn Kiệt | V1:   * Xây dựng frame chương trình. * Viết các chức năng thêm, xóa, sửa, load dữ liệu Khách hàng, Nhân Viên. * Chỉnh sửa, hoàn thiện câu query Hóa đơn và Chi Tiết hóa đơn * Thêm tính năng double click hóa đơn sẽ popup bảng Chi tiết hóa đơn   V2:   * Bổ sung tính năng Datepicker ở các trường có dữ liệu dạng ngày. * Chỉnh lại giao diện Guess, chỉnh sửa code phần login. * Chỉnh sửa, định dạng word. |
| 3 | Hoàng Hữu My | V1   * Test các chức năng, phát hiện, chỉnh sửa bugs * Viết các chức năng thêm, xóa, sửa, load dữ liệu Hóa đơn, Chi tiết hóa đơn * Soạn thảo word.   V2   * Bổ sung các Methods kiểm tra rằng buộc định dạng SĐT, ngày ở các tab Hóa Đơn. * Chỉnh sửa word |

**Mục lục**

[1. GIỚI THIỆU 1](#_Toc97039623)

[2. NỘI DUNG 1](#_Toc97039624)

[2.1. Giao diện 1](#_Toc97039625)

[2.2. Xây dựng class ConnectDB 4](#_Toc97039626)

[2.3. Xây dựng các chức năng chính 6](#_Toc97039627)

[2.3.1. Chức năng load dữ liệu 6](#_Toc97039628)

[2.3.2. Chức năng thêm dữ liệu 7](#_Toc97039629)

[2.3.3. Chức năng xóa dữ liệu 10](#_Toc97039630)

[2.3.4. Chức năng sửa dữ liệu 11](#_Toc97039631)

[2.3.5. Chức năng tìm kiếm dữ liệu 13](#_Toc97039632)

[2.4. Xây dựng chức năng Hiển thị Frame của Chi tiết Hóa Đơn 14](#_Toc97039633)

[3. KẾT LUẬN 16](#_Toc97039634)

[3.1. Ưu điểm 16](#_Toc97039635)

[3.2. Nhược điểm 16](#_Toc97039636)

[3.3. Hướng phát triển 16](#_Toc97039637)

# GIỚI THIỆU

Quản lý bán hàng là một phần quan trọng trong hoạt động tổ chức của doanh nghiệp kinh doanh. Cho dù buôn bán một dịch vụ hay một sản phẩm, việc lập kế hoạch, xây dựng quy trình và theo dõi kết quả kinh doanh là điều quan trọng hàng đầu để có thể kiểm soát được quá trình vận hành doanh nghiệp thương mại.

Trong thời đại phát triển nhanh chóng của công nghệ thông tin. Việc sử dụng phần mềm quản lý bán hàng để kiểm soát hoạt động kinh doanh là điều không thể thiếu đối với các doanh nghiệp. Một phần mềm quản lý bán hàng được xây dựng tốt giúp tiết kiệm chi phí đáng kể cho doanh nghiệp trong việc theo dõi, quản lý thông tin về sản phẩm, khách hàng và tình hình kinh doanh.

Bên cạnh đó, kết quả chính xác hơn từ phần mềm bán hàng giúp doanh nghiệp có thêm dữ liệu phục vụ việc dự đoán doanh thu, lên kế hoạch kinh doanh từ đó phát triển, mở rộng quy mô doanh nghiệp.

Đứng trước tiềm năng này, nhóm đã thực hiện xây dựng phần mềm quản lý bán hàng với mục đích xây dựng được một chương trình quản lý bán hàng dơn giản. Gồm các chức năng cơ bản: thêm, xóa, sửa, tìm kiếm, lưu trữ và hiển thị các thông tin khách hàng, nhân viên, sản phẩm, hóa đơn và chi tiết hóa đơn.

Tuy vậy, phần mềm do nhóm xây dựng hoàn toàn có thể sử dụng để phục vụ quá trình quản lý doanh nghiệp kinh doanh nhỏ. Bên cạnh đó, sản phẩm phần mềm có thể được nâng cấp cơ sở dữ liệu, bổ xung các tính năng và danh mục đối tượng cần quản lý, thay thế công nghệ phù hợp. Từ đó có thể được sử dụng ở doanh nghiệp quy mô lớn và hoạt động kinh doanh với nhiều thành phần cần quản lý hơn.

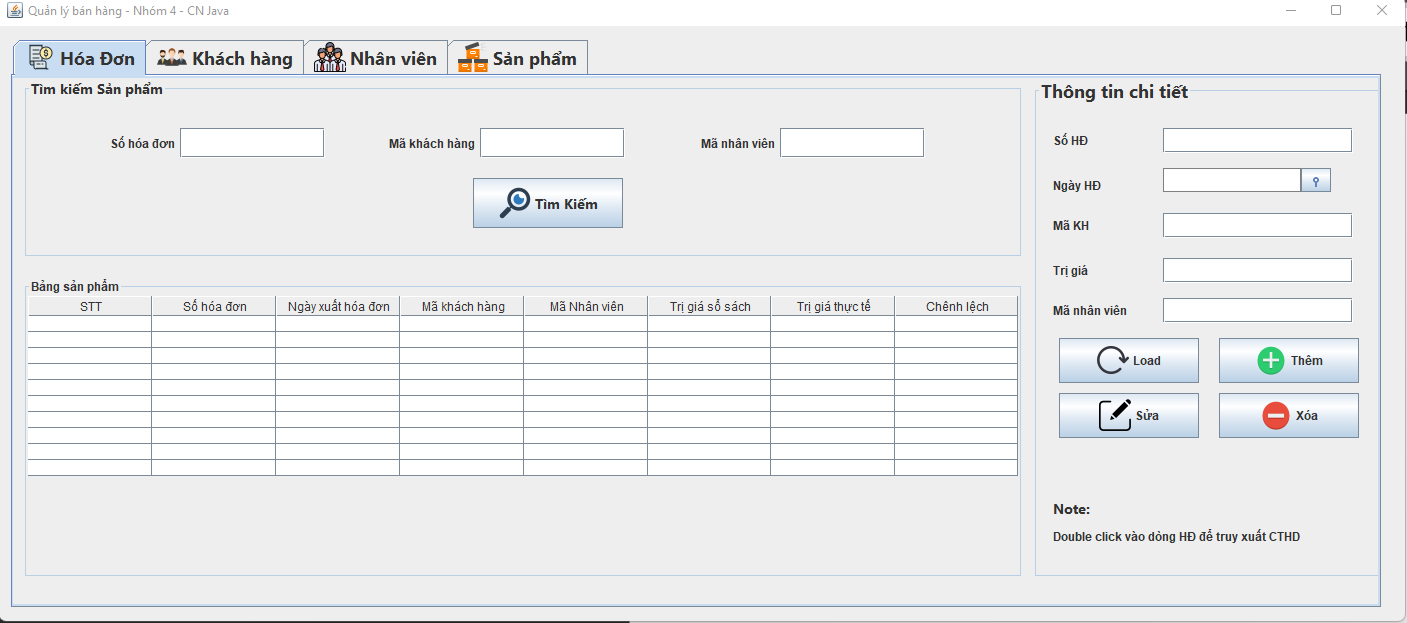
# NỘI DUNG

## Giao diện

Giao diện app được xây dựng theo dạng tab panel. Mỗi mục Hóa Đơn, Khách hàng, Nhân viên, Sản phẩm sẽ là một tab.

Ở các tab sẽ có bố cục tương tự nhau chia làm 3 khu vực chính bao gồm:

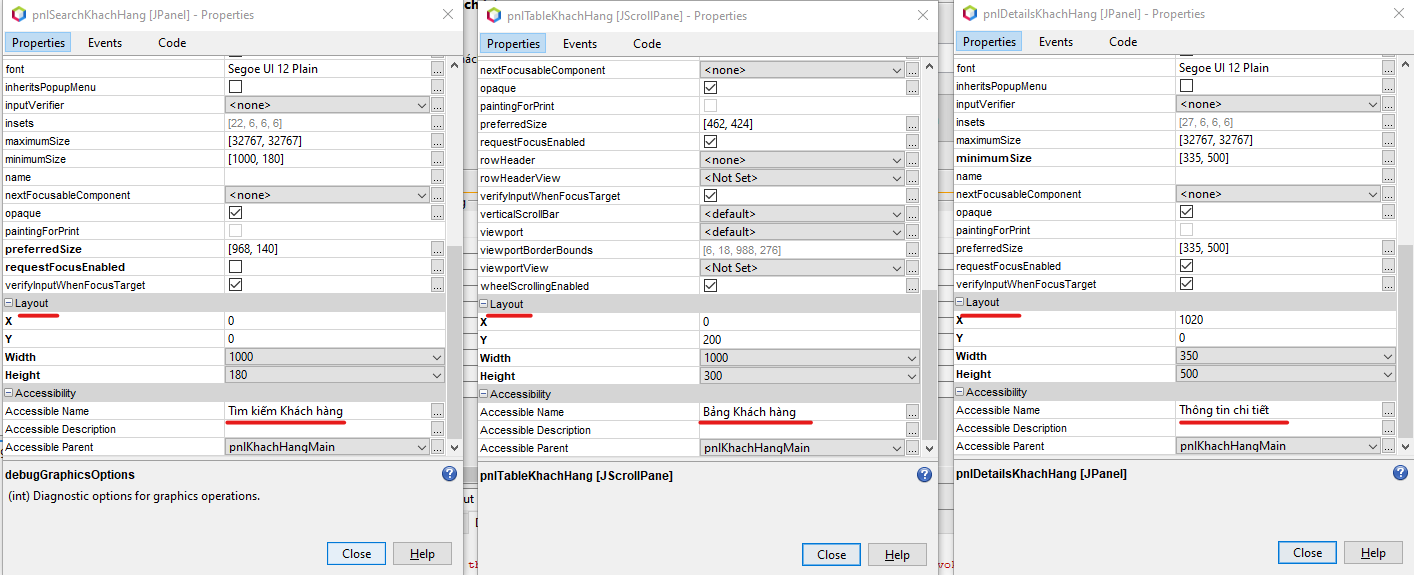
* Khu vực tìm kiếm: Bao gồm các JTextField để nhập thông tin tìm kiếm và JButton Tìm kiếm để thực hiện lệnh tìm kiếm.
* Khu vực Bảng thông tin: gồm JTable để trả về bảng thông tin.
* Khu vực Thông tin chi tiết: trả về các trường thông tin và cụm JButton giữ chức năng thao tác Load/Thêm/Xóa/Sửa dữ liệu.

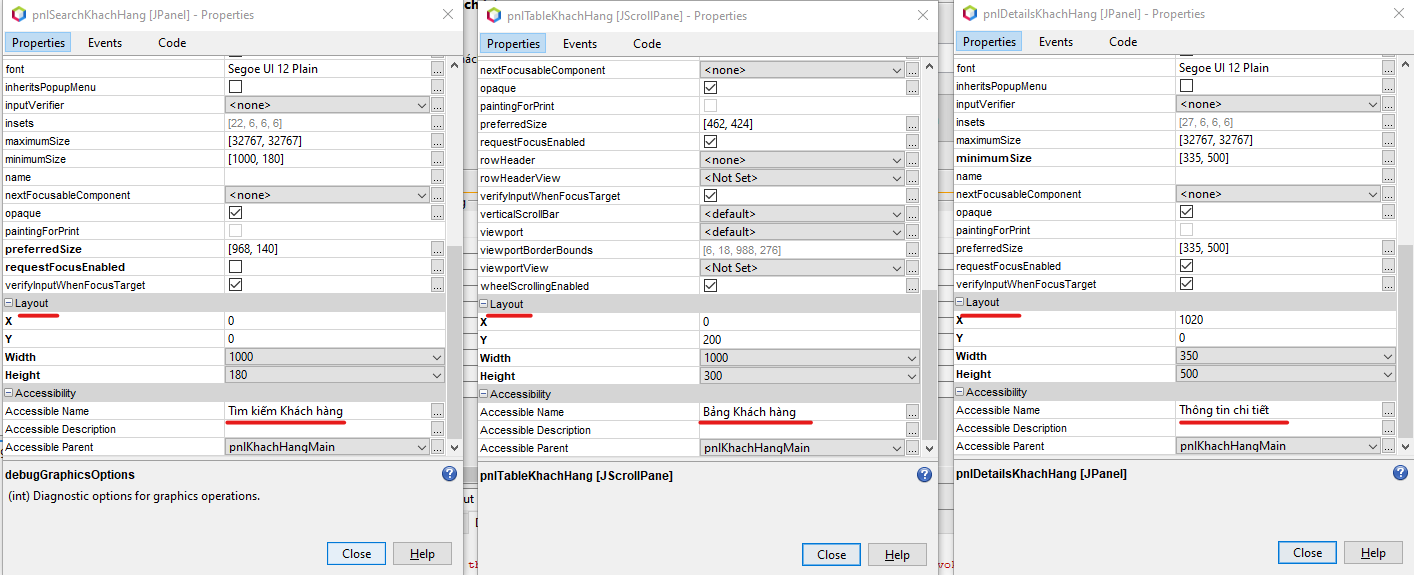


Hình ‑: Giao diện chương trình

Để tạo sự đồng nhất giữa các tab, các Panels chính sẽ được thiết lập layout là Null. Các thông số vị trí kích thước Panel con như sau:

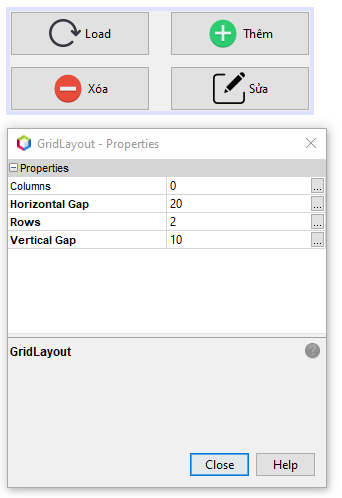
* Panel Tìm kiếm: X: 0; Y: 0; Width: 1000; Height: 180;
* Panel Bảng thông tin: X: 0; Y: 200; Width: 1000; Height: 300;
* Panel Thông tin chi tiết: X: 1020; Y: 0; Width: 350; Height: 500;





Hình ‑: Thông số kích thước các Panels

Cụm JButtons chức năng được thiết lập layout là Grid Layout để đảm bảo được kích thước, vị trí, khoảng cách của 4 nút trong Panel Button là đều nhau.



Hình ‑: Thông số GridLayout của cụm JButtons chức năng

Giao diện của phân quyền Guess sẽ tương tự giao diện chính như chỉ có tab Hóa đơn và tab Sản phẩm, các chức năng Thêm, Xóa, Sửa sẽ bị khóa.

Graphical user interface

Description automatically generated

Hình ‑: Giao diện phân quyền Guess

## Xây dựng class ConnectDB

1. Code:
2. public class ConnectDB {
3. //Attribute
4. //Thay đổi thông số để connect
5. String conUser = "sa";
6. String conPassword = "sa123";
7. String conServerName = "ACER-KIETVU";
8. String conPortNumber = "1433";
9. String conDatabaseName = "QLYBANHANG";
10. Connection con;
11. String error;
13. //Contruction
14. public ConnectDB() {
15. }
17. public void connectToDB() {
18. // TODO code application logic here
19. SQLServerDataSource ds = new SQLServerDataSource();
20. ds.setUser(this.conUser);
21. ds.setPassword(this.conPassword);
22. ds.setServerName(this.conServerName);
23. ds.setPortNumber(Integer.parseInt(this.conPortNumber));
24. ds.setDatabaseName(this.conDatabaseName);
26. try {
27. // Execute a stored procedure that returns some data.
28. Connection con = ds.getConnection();
29. // Iterate through the data in the result set and display it.
30. if (con != null) {
31. this.con = con;
32. System.out.println("Thanh cong");
33. }
34. }
35. // Handle any errors that may have occurred.
36. catch (SQLException ex) {
37. this.setError(ex.toString());
38. System.out.println(this.getError());
39. }
40. }
42. //Methods
43. //Thực thi câu lệnh SELECT
44. public ResultSet excuteQueryGetTable(String cauTruyVanSQL){
45. try {
46. Statement stmt = this.con.createStatement();
47. ResultSet rs = stmt.executeQuery(cauTruyVanSQL);
48. return rs;
49. } catch (SQLException ex) {
50. this.setError(ex.toString());
51. System.out.println(this.getError());
52. }
53. return null;
54. }
55. //Thực thi INSERT, DELETE, UPDATE
56. public void excuteQueryUpdateDB(String cauTruyVanSQL){
58. try {
59. Statement stmt = this.con.createStatement();
60. stmt.executeUpdate(cauTruyVanSQL);
62. } catch (SQLException ex) {
63. this.setError(ex.toString());
64. System.out.println(this.getError());
65. }
66. }
67. public String getError() {
68. return error;
69. }
70. public void setError(String error) {
71. this.error = error;
72. }
73. public Connection getCon() {
74. return con;
75. }
76. public void closeConnect(){
77. try {
78. if(this.con != null)
79. this.setError(null);
80. this.con.close();
81. }
82. catch (SQLException ex) {
83. this.setError(ex.toString());
84. System.out.println(this.getError());
85. }
86. }
87. }
88. Mô tả:

* Các Attributes của Class gồm:
* Các thông số để kết nối với Database.
* Connection con để lưu connection
* String error để lưu lỗi của truy vấn
* Methods gồm:
* connectToDB: để kết nối đến database
* excuteQueryGetTable: để thực hiện câu truy vấn Select
* excuteQueryUpdateDB: để thực hiện câu truy vấn Insert, Delete, Update
* getError/setError:
* closeConnect: để đóng kết nối.

## Xây dựng các chức năng chính

Chương trình gồm các chức năng chính gồm: load dữ liệu, thêm dữ liệu, xóa dữ liệu, sửa dữ liệu và tìm kiếm. Các chức năng này tuy truy vấn các Dữ liệu khác nhau nhưng cấu trúc câu lệnh tương tự nhau. Vì vậy, trong bài báo cáo, nhóm lập trình lấy Code thao tác với dữ liệu Khách hàng để báo cáo.

### Chức năng load dữ liệu

1. Code:
2. private void loadKHData (){
3. try {
4. //Khởi tạo Class connectDB
5. con.connectToDB();
6. if (con.getCon()!=null) {
7. System.out.println("Ket noi thanh cong");
9. //Dùng method excuteQueryGetTable của class connectDB
10. ResultSet rs = con.excuteQueryGetTable("Select \* from KHACHHANG");
11. Object[] obj = new Object[]{"STT","Mã KH", "Họ tên", "Địa chỉ ", "SĐT", "Ngày sinh", "Doanh số", "Ngày ĐK"};
12. DefaultTableModel tableModel = new DefaultTableModel(obj, 0);
13. tblKhachHang.setModel(tableModel);
14. int stt = 0;
15. while (rs.next()) {
16. Object[] item = new Object[8];
17. stt++;
18. item[0] = stt;
19. item[1] = rs.getString("MAKH");
20. item[2] = rs.getString("HOTEN");
21. item[3] = rs.getString("DCHI");
22. item[4] = rs.getString("SODT");
23. item[5] = rs.getString("NGSINH").substring(0, 10);
24. item[6] = rs.getString("DOANHSO");
25. item[7] = rs.getString("NGDK").substring(0, 10);
26. tableModel.addRow(item);
27. }
28. }
29. else {
30. System.out.println("Ket noi false");
31. }
32. con.closeConnect();
34. } catch (SQLException ex) {
35. }
36. }
37. Mô tả:

* Bước 1: kết nối database.
* Bước 2: sau khi kết nối thành công thì chuẩn bị và thực hiện câu truy vấn.
* Bước 3: lưu kết quả trả về từ câu truy vấn vào biến rs.
* Bước 4: đưa dữ liệu vào model DefaultTableModel để hiển thị lên giao diện.
* Bước 5: đóng kết nối.

1. Hướng dẫn sử dụng:

* Bước 1: Mở phần mềm chọn vào tab cần thao tác.
* Bước 2: Nhấn vào nút Load bên phải giao diện để tải dữ liệu.

### Chức năng thêm dữ liệu

1. Code:

Để thêm dữ liệu, app sẽ kiểm tra các trường thông tin đã được cung cấp dữ liệu đầy đủ không. Vì vậy, app sẽ có thêm method kiemTraNhap.

1. private boolean kiemTraNhapKhachHang() {
2. String newMaKH, newHoTen, newDiaChi, newSDT, newNgaySinh, newNgayDK, newDoanhSo, thongBao;
3. boolean kiemtra = false;
5. newMaKH= txtMaKH.getText();
6. newHoTen= txtHoTenKH.getText();
7. newDiaChi= txtDiaChiKH.getText();
8. newSDT= txtSDTKH.getText();
9. newNgaySinh= dateFormat.format(dpNgaySinhKH.getDate());
10. newNgayDK= dateFormat.format(dpNgayDKKH.getDate());
11. newDoanhSo= txtDoanhSoKH.getText();
12. thongBao ="";
14. if (newMaKH.equals("")) {
15. thongBao += "Bạn chưa nhập mã khách hàng\n";
16. txtMaKH.setForeground(Color.red);
17. }
18. if (newHoTen.equals("")) {
19. thongBao += "Bạn chưa Nhập Tên Khách hàng \n";
20. txtHoTenKH.setForeground(Color.red);
21. }
22. if (newDiaChi.equals("")) {
23. txtDiaChiKH.setForeground(Color.red);
24. thongBao += "Bạn chưa Nhập Địa Chỉ\n";
25. }
26. if (newSDT.equals("")) {
27. txtSDTKH.setForeground(Color.red);
28. thongBao += "Bạn chưa Nhập Số ĐT \n";
29. }
30. if (newNgaySinh.equals("")) {
31. dpNgaySinhKH.setForeground(Color.red);
32. thongBao += "Bạn chưa Nhập ngày sinh \n";
33. }
34. if (newNgayDK.equals("")) {
35. dpNgayDKKH.setForeground(Color.red);
36. thongBao += "Bạn chưa Nhập ngày đăng ký\n";
37. }
38. if (newDoanhSo.equals("")) {
39. txtDoanhSoKH.setForeground(Color.red);
40. thongBao += "Bạn chưa Nhập doanh số\n";
41. }
42. if (thongBao.equals("")) {
43. kiemtra = true;
44. txtMaKH.setForeground(Color.black);
45. txtHoTenKH.setForeground(Color.black);
46. txtDiaChiKH.setForeground(Color.black);
47. txtSDTKH.setForeground(Color.black);
48. dpNgaySinhKH.setForeground(Color.black);
49. dpNgayDKKH.setForeground(Color.black);
50. txtDoanhSoKH.setForeground(Color.black);
51. } else {
52. kiemtra = false;
53. JOptionPane.showMessageDialog(null,thongBao);
54. }
55. return kiemtra;
56. }

1. private void themKH(){
2. String newMaKH, newHoTen, newDiaChi, newSDT, newNgaySinh, newNgayDK, newDoanhSo;
3. newMaKH= txtMaKH.getText();
4. newHoTen= txtHoTenKH.getText();
5. newDiaChi= txtDiaChiKH.getText();
6. newSDT= txtSDTKH.getText();
7. newNgaySinh= dateFormat.format(dpNgaySinhKH.getDate());
8. newNgayDK= dateFormat.format(dpNgayDKKH.getDate());
9. newDoanhSo= txtDoanhSoKH.getText();
11. if(this.kiemTraNhapNgay(newNgaySinh) == false){
12. JOptionPane.showMessageDialog(null, con.getError() + "\nĐịnh dạng ngày không hợp lệ !\nVui lòng nhập theo định dạng yyyy-mm-dd");
13. return;
14. }
16. if(this.kiemTraNhapNgay(newNgayDK) == false){
17. JOptionPane.showMessageDialog(null, con.getError() + "\nĐịnh dạng ngày không hợp lệ !\nVui lòng nhập theo định dạng yyyy-mm-dd");
18. return;
19. }
21. if(this.kiemTraSo(newSDT) == false){
22. JOptionPane.showMessageDialog(null, con.getError() + "\nĐịnh dạng SDT không hợp lệ !\nVui lòng nhập lại");
23. return;
24. }
26. String cauTruyVan = "INSERT INTO KHACHHANG (MAKH,HOTEN,DCHI,SODT,NGSINH,NGDK,DOANHSO) VALUES ('" + newMaKH + "','" + newHoTen + "','" + newDiaChi + "','" + newSDT + "','" + newNgaySinh + "','" + newNgayDK + "','" + newDoanhSo + "')";
27. System.out.println(cauTruyVan);
28. boolean kiemtra = kiemTraNhapKhachHang();
29. if (kiemtra) {
30. con.connectToDB();
31. con.excuteQueryUpdateDB(cauTruyVan);
32. if (con.getError()!=null) {
33. JOptionPane.showMessageDialog(null, con.getError() + "\nThêm dữ liệu không thành công");
34. } else {
35. JOptionPane.showMessageDialog(null,"Thêm dữ liệu thành công");
36. }
37. }
38. loadKHData();
39. con.closeConnect();
40. }
41. Mô tả:

* Bước 1: lấy dữ liệu từ ô input mà người dùng nhập vào để chuẩn bị câu truy vấn. Nếu chưa nhập dữ liệu, chương trình sẽ thông báo.
* Bước 2: kết nối database.
* Bước 3: thực hiện câu truy vấn.
* Bước 4: tải lại dữ liệu.
* Bước 5: đóng kết nối.

1. Hướng dẫn sử dụng:

* Bước 1: mở phần mềm chọn vào tab cần thao tác.
* Bước 2: nhập dữ liệu sản phẩm cần thêm vào các text box bên phải, dữ liệu ngày có thể nhập tay định dạng yyyy-MM-dd hoặc chọn từ date picker.
* Bước 3: nhấn vào nút Thêm bên phải giao diện để thêm dữ liệu.

Lưu ý: Mã Khách hàng, Mã Nhân Viên, Mã Sản phẩm, Mã hóa đơn là khóa chính, vì vậy khi thêm mới phải đảm bảo không trùng với các mã đã có sẵn trong cơ sở dữ liệu.

### Chức năng xóa dữ liệu

1. Code:
2. private void xoaKH(){
3. int[] clickedRows = tblKhachHang.getSelectedRows();
4. if (clickedRows.length>0) {
5. int dialogButton = JOptionPane.YES\_NO\_OPTION;
6. int dialogResult = JOptionPane.showConfirmDialog(null,"Bạn muốn xóa "+clickedRows.length+ " Khách hàng?", "Warning",dialogButton);
7. if (dialogResult == JOptionPane.YES\_OPTION) {
8. for (int i = 0; i < clickedRows.length; i++) {
9. String delMaKH = tblKhachHang.getValueAt(clickedRows[i], 1).toString();
10. String cauTruyVan = "delete KHACHHANG where MAKH= '" + delMaKH +"'";
11. con.connectToDB();
12. con.excuteQueryUpdateDB(cauTruyVan);
13. if (con.getError()!=null) {
14. JOptionPane.showMessageDialog(null,con.getError() + "\nXóa dữ liệu không thành công");
15. break;
16. }
17. if (i==clickedRows.length-1 && con.getError()==null) {
18. JOptionPane.showMessageDialog(null,"Xóa dữ liêu thành công.");
19. }
20. }
22. loadKHData();
23. con.closeConnect();
24. }
25. else{
26. JOptionPane.showMessageDialog(null,"Hủy lệnh.");
27. con.closeConnect();
28. }
29. } else {
30. JOptionPane.showMessageDialog(null,"Bạn chưa chọn Khách hàng");
31. con.closeConnect();
32. }
33. }
34. Mô tả:

* Bước 1: lấy các chỉ số dòng của sản phẩm cần xóa.
* Bước 2: hiển thị hộp thoại xác nhận xem người dùng có thực sự muốn xóa dữ liệu sản phẩm.
* Bước 3: chuẩn bị câu truy vấn.
* Bước 4: kết nối database.
* Bước 5: thực hiện câu truy vấn.
* Bước 6: load lại dữ liệu sản phẩm sau khi đã xóa.
* Bước 7: đóng kết nối.

1. Hướng dẫn sử dụng:

* Bước 1: mở phần mềm chọn vào tab cần thao tác.
* Bước 2: chọn các dòng cần xóa.
* Bước 3: nhấn vào nút Xóa bên phải giao diện để xóa dữ liệu.

### Chức năng sửa dữ liệu

1. Code:
2. private void suaKH(){
3. String suaHoTen= txtHoTenKH.getText();
4. String suaDiaChi= txtDiaChiKH.getText();
5. String suaSDT= txtSDTKH.getText();
6. String suaNgaySinh= dateFormat.format(dpNgaySinhKH.getDate());
7. String suaNgayDK= dateFormat.format(dpNgayDKKH.getDate());
8. String suaDoanhSo= txtDoanhSoKH.getText();
9. int[] clickedRows = tblKhachHang.getSelectedRows();
10. if (clickedRows.length<1) {
11. JOptionPane.showMessageDialog(null,"Bạn chưa chọn khách hàng.");
12. }
13. if (clickedRows.length==1) {
14. int dialogButton = JOptionPane.YES\_NO\_OPTION;
15. int dialogResult = JOptionPane.showConfirmDialog(null,"Bạn muốn sửa thông tin khách hàng?", "Warning",dialogButton);
16. if (dialogResult == JOptionPane.YES\_OPTION) {
17. if(this.kiemTraNhapNgay(suaNgaySinh) == false){
18. JOptionPane.showMessageDialog(null, con.getError() + "\nĐịnh dạng ngày không hợp lệ !\nVui lòng nhập theo định dạng yyyy-mm-dd");
19. return;
20. }
21. if(this.kiemTraNhapNgay(suaNgayDK) == false){
22. JOptionPane.showMessageDialog(null, con.getError() + "\nĐịnh dạng ngày không hợp lệ !\nVui lòng nhập theo định dạng yyyy-mm-dd");
23. return;
24. }
25. if(this.kiemTraSo(suaSDT) == false){
26. JOptionPane.showMessageDialog(null, con.getError() + "\nĐịnh dạng SÐT không hợp lệ !\n");
27. return;
28. }

31. String MaKH = tblKhachHang.getValueAt(clickedRows[0], 1).toString();
32. String cauTruyVan = "update KHACHHANG set "
33. + "HOTEN = '"+  suaHoTen+"'"
34. + ", DCHI = '"+ suaDiaChi+"'"
35. + ", SODT = '"+ suaSDT+"'"
36. + ", NGSINH = '"+ suaNgaySinh+"'"
37. + ", DOANHSO = '"+ suaDoanhSo+"'"
38. + ", NGDK = '" + suaNgayDK+"'"
39. + " where MAKH = '" + MaKH+"'";
40. con.connectToDB();
41. con.excuteQueryUpdateDB(cauTruyVan);
42. System.out.println("test cau truy vấn: "+cauTruyVan);
43. if (con.getError()!=null) {
44. JOptionPane.showMessageDialog(null,con.getError() + "\nSửa dữ liệu không thành công");
45. } else {
46. JOptionPane.showMessageDialog(null,"Sửa dữ liệu thành công");
47. }
48. loadKHData();
49. con.closeConnect();
50. }
51. else{
52. JOptionPane.showMessageDialog(null,"Hủy lệnh.");
53. con.closeConnect();
54. }
55. } else if(clickedRows.length>1) {
56. JOptionPane.showMessageDialog(null,"Chỉ chọn 1 Khách hàng");
57. con.closeConnect();
58. }
59. }
60. Mô tả:

* Bước 1: lấy chỉ số dòng của dữ liệu cần sửa và lấy dữ liệu từ ô input mà người dùng nhập vào.
* Bước 2: hiển thị hộp thoại xác nhận xem người dùng có thực sự muốn sửa dữ liệu sản phẩm.
* Bước 3: chuẩn bị câu truy vấn.
* Bước 4: kết nối database.
* Bước 5: thực hiện câu truy vấn.
* Bước 6: load lại dữ liệu sản phẩm sau khi sửa.
* Bước 7: đóng kết nối.

1. Hướng dẫn sử dụng:

* Bước 1: mở phần mềm chọn vào tab cần thao tác.
* Bước 2: chọn dòng dữ liệu cần sửa và nhập dữ liệu sản phẩm muốn cập nhật vào các text box bên phải, dữ liệu ngày có thể nhập tay định dạng yyyy-MM-dd hoặc chọn từ date picker.
* Bước 3: nhấn vào nút Sửa bên phải giao diện để cập nhật dữ liệu.

Lưu ý: Mã khách hàng là khóa chính do đó sẽ không sửa được.

### Chức năng tìm kiếm dữ liệu

1. Code:
2. private void timKH(){
3. String searchMaKH = txtSearchMaKH.getText();
4. String searchTenKH = txtSearchTenKH.getText();
5. String searchSDTKH = txtSearchSDTKH.getText();
6. if (searchMaKH.isEmpty() && searchTenKH.isEmpty() && searchSDTKH.isEmpty()) {
7. JOptionPane.showMessageDialog(null,"Chưa nhập dữ liệu tìm kiếm");
8. } else{
9. String cauTruyVan = "Select \* from KHACHHANG WHERE 1=1 ";
10. if (!searchMaKH.isEmpty()) {
11. cauTruyVan = cauTruyVan + "and MAKH = '"+ searchMaKH +"' ";
12. }
13. if (!searchTenKH.isEmpty()) {
14. cauTruyVan = cauTruyVan + " and HOTEN like '%"+ searchTenKH +"%'";
15. }
16. if (!searchSDTKH.isEmpty()) {
17. cauTruyVan = cauTruyVan + " and SODT like '%"+ searchSDTKH +"%'";
18. }
19. System.out.println(cauTruyVan);
20. try{
21. con.connectToDB();
22. ResultSet rs = con.excuteQueryGetTable(cauTruyVan);
23. Object[] obj = new Object[]{"STT","Mã KH", "Họ tên", "Địa chỉ ", "SĐT", "Ngày sinh", "Doanh số", "Ngày ĐK"};
24. DefaultTableModel tableModel = new DefaultTableModel(obj, 0);
25. tblKhachHang.setModel(tableModel);
26. int stt = 0;
27. while (rs.next()) {
28. Object[] item = new Object[8];
29. stt++;
30. item[0] = stt;
31. item[1] = rs.getString("MAKH");
32. item[2] = rs.getString("HOTEN");
33. item[3] = rs.getString("DCHI");
34. item[4] = rs.getString("SODT");
35. item[5] = rs.getString("NGSINH").substring(0, 10);
36. item[6] = rs.getString("DOANHSO");
37. item[7] = rs.getString("NGDK").substring(0, 10);
38. tableModel.addRow(item);
39. }
40. con.closeConnect();
41. } catch (SQLException ex) {
42. }
43. }
44. }
45. Mô tả:

* Bước 1: lấy dữ liệu tìm kiếm của người dùng từ text input.
* Bước 2: chuẩn bị câu truy vấn.
* Bước 3: kết nối database.
* Bước 4: thực hiện câu truy vấn và lưu kết quả vào biến rs.
* Bước 5: đưa dữ liệu từ biến rs vào DefaultTableModel để hiện thị lên giao diện.
* Bước 6: đóng kết nối.

1. Hướng dẫn sử dụng:

* Bước 1: mở phần mềm chọn vào tab cần thao tác.
* Bước 2: nhập dữ liệu tìm kiếm vào các text input.
* Bước 3: nhấn vào nút Tìm kiếm để tìm dữ liệu.

## Xây dựng chức năng Hiển thị Frame của Chi tiết Hóa Đơn

1. Code:
2. //Lấy số hóa đơn khi click ở bảng Hóa đơn
3. private String getSelectedSoHD(int clickedRow) {
4. String soHD = tblHoaDon.getValueAt(clickedRow, 1).toString();
5. return soHD;
6. }
7. private void loadCTHDSelectedHD(String soHD){
8. String cauTruyVan = """
9. select CT.SOHD, CT.MASP, SP.TENSP, CT.SL, (CT.SL\*SP.GIA) ThanhTien
10. from CTHD CT join SANPHAM SP on CT.MASP = SP.MASP
11. """
12. + "where CT.SOHD = '"
13. +soHD + "'"
14. ;
15. System.out.println(cauTruyVan);
16. try {
17. con.connectToDB();
18. ResultSet rs = con.excuteQueryGetTable(cauTruyVan);
19. DefaultTableModel tableModel = (DefaultTableModel) tblCTHD.getModel();
20. tableModel.setRowCount(0);
21. int stt = 0;
22. while (rs.next()) {
23. Object[] item = new Object[6];
24. stt++;
25. item[0] = stt;
26. item[1] = rs.getString("SOHD");
27. item[2] = rs.getString("MASP");
28. item[3] = rs.getString("TENSP");
29. item[4] = rs.getString("SL");
30. item[5] = rs.getString("ThanhTien");
31. tableModel.addRow(item);
32. }
33. con.closeConnect();
34. } catch (SQLException ex) {
35. }
36. }
37. //Hiển thị Panel CTHD khi double click bảng hóa đơn
38. private void tblHoaDonMouseClicked(java.awt.event.MouseEvent evt) {//GEN-FIRST:event\_tblHoaDonMouseClicked
39. // TODO add your handling code here:
40. if (evt.getClickCount()==1) {
41. selectHD();
42. }
43. if (evt.getClickCount()==2) {
44. frmCTHD.setSize(1400, 550);
45. frmCTHD.setLocation(50, 50);
46. frmCTHD.show(true);
47. String soHD = getSelectedSoHD(tblHoaDon.getSelectedRow());
48. loadCTHDSelectedHD(soHD);
49. txtSoHD\_CTHD.setText(soHD);
50. }
51. }//GEN-LAST:event\_tblHoaDonMouseClicked
52. Mô tả:

* Bước 1: Từ tab Hóa đơn, sau khi load bảng hóa đơn, double click để mở pop up Frame Chi tiết hóa đơn, đồng hời lưu biến soHD là số hóa đơn của dòng được double clicked.
* Bước 2: Cấu hình thông số về kích thước và bị trí của cửa sổ popup.
* Bước 3: Gọi hàm loadCTHDSelectedHD với tham số đầu vào là soHD. Hàm loadCTHDSelectedHD chạy tương tự như các hàm load khác.

1. Hướng dẫn sử dụng

* Bước 1: Ở tab Hóa đơn, load dữ liệu.
* Bước 2: Double click vào dòng hóa đơn muốn mở truy xuất Chi tiết hóa đơn

Lưu ý: Muốn load toàn bộ các chi tiết hóa đơn, người dùng xóa dòng Số HĐ và bấm load.

# KẾT LUẬN

## Ưu điểm

* Giao diện được phân chia theo tab để người dùng dễ thao tác.
* Chương trình đáp ứng đầy đủ tính năng thêm, xóa, sửa, tìm kiếm, load dữ liệu ở tất cả các tabs
* Ở các ô nhập liệu ngày, người dùng có thể nhập tay theo định dạng, hoặc sử dụng date picker.
* Chương trình có kiểm tra ràng buộc định dạng ngày tháng (yyyy-mm-dd) và định dạng số ở SĐT, hạn chế việc người dùng nhập sai kiểu dữ liệu.
* Chương trình hổ trợ 2 giao diện admin và guess.
* Admin sẽ có đầy đủ chức năng ở tất cả tabs.
* Guess chi được xem dữ liệu ở Hóa đơn, Chi tiết Hóa đơn, và Sản phẩm

## Nhược điểm

* Ở thao tác thêm Hóa đơn, Chương trình chưa hổ trợ ComboBox để người dùng có thể chọn Mã KH và Mã NV
* Phân quyền chỉ có 2 roles là Guess và Admin còn đơn giản.
* Sản phẩm chưa bổ sung hình ảnh minh họa

## Hướng phát triển

Từ các nhược điểm kể trên, hướng phát triển tiếp theo của nhóm sẽ tập trung vào hai trọng tâm sau:

* Phần phân quyền để cụ thể và phù hợp hơn với thực tế.
* Phát triển ComboBox ở các ô nhập liệu Mã KH và Mã NV để người dùng chọn trên ComboBox hạn chế nhập liệu sai.