ĐẠI HỌC DUY TÂN

TRƯỜNG KHOA HỌC MÁY TÍNH

KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN CÁ NHÂN

ĐỀ TÀI: PHẦN MỀM QUẢN LÝ BÁN HÀNG

Môn Học **:** LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

GIẢNG VIÊN : NGUYỄN VĂN VĨ

LỚP : CS 311 S

SVTH : TÔN THẤT PHONG – 30211122181

Đà Nẵng, 11/2025

Mục Lục

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI & MỤC ĐÍCH ĐỀ TÀI 4
1.1. Giới thiệu đề tài 4
1.2. Mục tiêu 4
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT.....5
2.1. Ngôn ngữ Java và lập trình hướng đối tượng (OOP) 5
2.2. Java Swing – Xây dựng giao diện 8
2.3. Cơ sở dữ liệu Microsoft Access và kết nối Java với Access.....12
2.4. Mô hình kiến trúc MVC 13
2.5. Xử lý sự kiện (Event Handling) 13
CHƯƠNG 3. THIẾT KẾ GIAO DIỆN 14
3.1. Giao diện đăng nhập hệ thống (LoginView). 14
3.2. Giao diện chính.....15
3.4. Giao diện hóa đơn.....19
CHƯƠNG 4. THIẾT KẾ THUẬT TOÁN 20
4.1. Thuật toán đăng nhập 20
4.2. Thuật toán lấy danh sách sản phẩm.....20
4.3. Thuật toán thêm sản phẩm vào giỏ hàng. 21
4.4. Thuật toán tính tổng tiền.....22
4.5. Thuật toán tạo mã hóa đơn tự động. 22
4.6. Thuật toán tạo hóa đơn và lưu chi tiết hóa đơn. 22
CHƯƠNG 5. DEMO ĐỒ ÁN 24
5.1. Khởi động ứng dụng.....24
5.2. Đăng nhập thành công và chuyển sang giao diện chính.....25
5.3. Thao tác thêm sản phẩm vào giỏ hàng. 25
5.4. Thao tác thanh toán hóa đơn.....26
5.5. Lưu dữ liệu vào CSDL Access. 27
5.6. Đăng xuất.....28

CHƯƠNG 6: KẾT LUẬN 29
6.1. Kết quả đạt được. 29
Tài liệu tham khảo 30

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI & MỤC ĐÍCH ĐỀ TÀI

1.1. Giới thiệu đề tài

- Hiện nay, có rất nhiều chuỗi cửa hàng đồ uống như Mixue, Katinat,....có nhu cầu quản lý bán hàng nhanh chóng, chính xác, thân thiện với người sử dụng rất quan trọng.
- Hệ thống bán hàng được xây dựng bằng Java Swing kết hợp Cơ sở dữ liệu Microsoft Access (.accdb) nhằm hỗ trợ: quản lý nhân viên, quản lý sản phẩm, tạo hóa đơn và chi tiết hóa đơn, tính tiền tự động, hiển thị giao diện.

1.2. Mục tiêu

- Xây dựng giao diện bán hàng trực quan
- Quản lý sản phẩm hiệu quả
- Hỗ trợ tạo hóa đơn và chi tiết hóa đơn
- Xác thực đăng nhập và quản lý nhân viên
- Kết nối cơ sở dữ liệu Access bằng UcanAccess
- Áp dụng mô hình MVC
- Xây dựng nền tảng để phát triển hệ thống lớn hơn

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Ngôn ngữ Java và lập trình hướng đối tượng (OOP)

- Java được chọn làm ngôn ngữ chính nhờ các ưu điểm:

  - Hỗ trợ lập trình hướng đối tượng (OOP)
  - Mạnh mẽ, ổn định, dễ bảo trì
  - Thư viện GUI phong phú (Swing)
  - Có khả năng kết nối tốt với nhiều loại cơ sở dữ liệu

- Trong đồ án này, mã nguồn sử dụng JDK 21 (Adoptium Temurin) và được tổ chức theo mô hình MVC.

- Lập trình hướng đối tượng (OOP) trong dự án:

  - Tính đóng gói (Encapsulation):

Lớp **SanPham, NhanVien, HoaDon, ChiTietHoaDon** khai báo thuộc tính **private** và sử dụng **getter/setter** :

```java
package model;
public class NhanVien {
    private String maNV;
    private String tenNV;
    private String username;
    private String password;
    public NhanVien() {}
    public NhanVien(String maNV, String tenNV,
            String username, String password) {
        this.maNV = maNV;
        this.tenNV = tenNV;
        this.username = username;
        this.password = password;
    }
    public String getMaNV() {
        return maNV;
    }
    public void setMaNV(String maNV) {
        this.maNV = maNV;
    }
    public String getTenNV() {
        return tenNV;
    }
    public void setTenNV(String tenNV) {
        this.tenNV = tenNV;
    }
    public String getUsername() {
        return username;
    }
    public void setUsername(String username) {
        this.username = username;
    }
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
}
```

- Tính kế thừa (Inheritance)

Trong dự án không sử dụng kế thừa phức tạp, nhưng áp dụng cấu trúc DAO (dùng chung kết nối DB):

```java
package dao;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class KetNoiAccess {
    private static final String DB_URL = "jdbc:ucanaccess://D:\\BaitapJava\\QuanLyBanHang\\mixue.accdb";
    private static Connection conn = null;
    public static Connection getConnection() {
        try {
            if (conn == null || conn.isClosed()) {
                conn = DriverManager.getConnection(DB_URL);
                System.out.println("Ket Noi Aceess thanh cong!");
            }
        } catch (SQLException e) {
            System.out.println("Loi ket noi CSDL: " + e.getMessage());
        }
        return conn;
    }

    public static void closeConnection() {
        try {
            if (conn != null && !conn.isClosed()) {
                conn.close();
                System.out.println("Da dong ket noi.");
            }
        } catch (SQLException e) {
            System.out.println("Loi khi dong ket noi: " + e.getMessage());
        }
    }

    public static void main(String[] args) {
        Connection testConn = KetNoiAccess.getConnection();
        KetNoiAccess.closeConnection();
    }
}
```

- Tính đa hình (Polymorphism)

Các DAO sử dụng chung cách truy vấn nhưng xử lý dữ liệu khác nhau:

```java
public class SanPhamDAO {
    public static List<SanPham> getAllSanPham() {
        List<SanPham> list = new ArrayList<>();
        try (Connection conn = KetNoiAccess.getConnection()) {
            String sql = "SELECT * FROM sanpham";
            PreparedStatement ps = conn.prepareStatement(sql);
            ResultSet rs = ps.executeQuery();
            while (rs.next()) {
                String ma = rs.getString("maSp");
                String ten = rs.getString("tenSp");
                String loai = rs.getString("loai");
                double gia = rs.getDouble("gia");

                list.add(new SanPham(ma, ten, loai, gia));
            }
        } catch (Exception e) {
            System.out.println("Loi doc du lieu san pham: " + e.getMessage());
        }
        return list;
    }
```

- Tính trừu tượng (Abstraction)

Ẩn đi cách kết nối Access, chỉ trả dữ liệu cho lớp View.

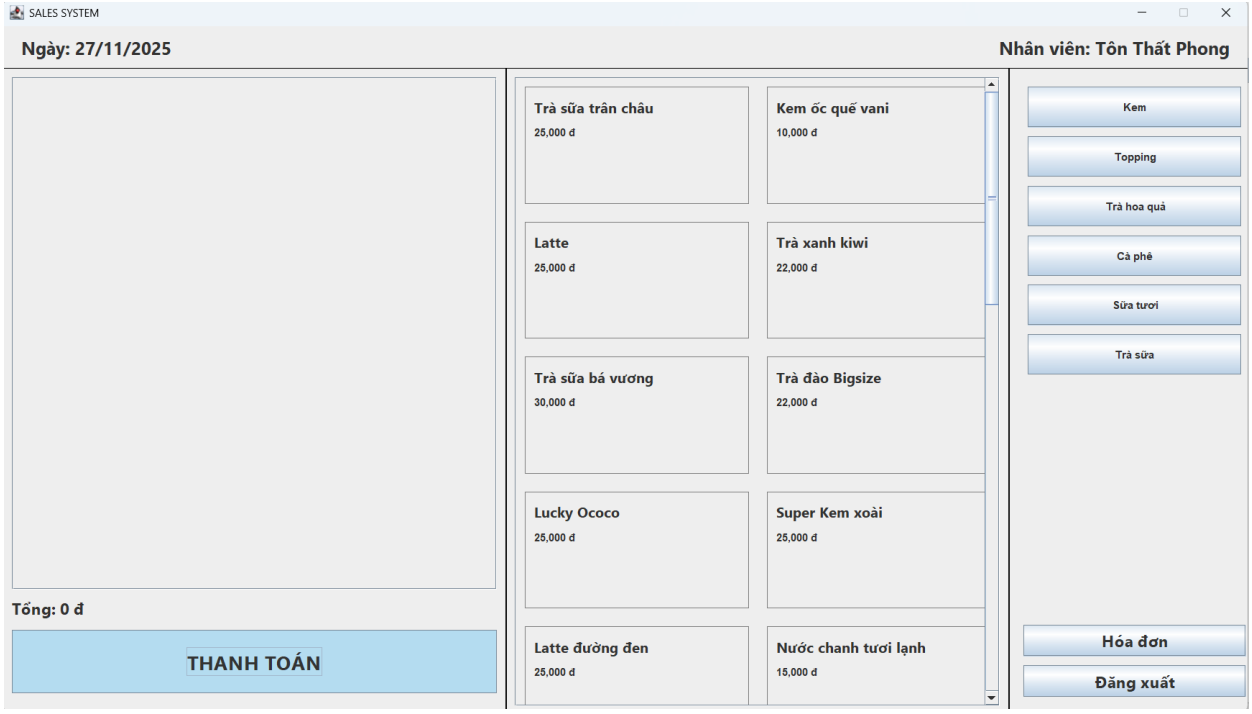
2.2. Java Swing – Xây dựng giao diện

- Java Swing là thư viện GUI chính, hỗ trợ tạo ứng dụng desktop.

- Đồ án sử dụng các thành phần:

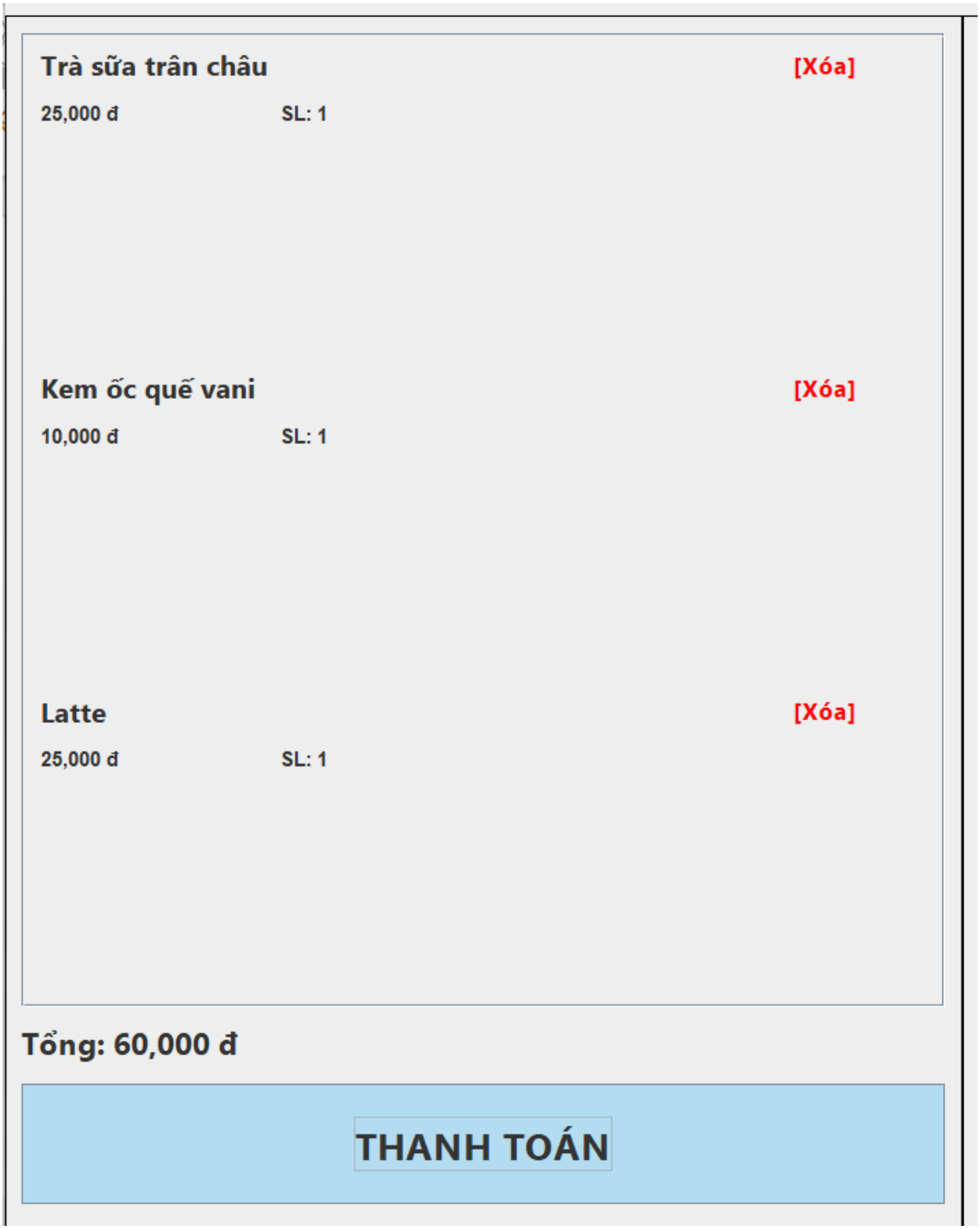
  - JFrame: Tạo cửa sổ chính như **LoginView, View**

ĐĂNG NHẬP
SALES SYSTEM
Tên đăng nhập:
Mật khẩu:
Hiện mật khẩu
Đăng nhập
Thoát

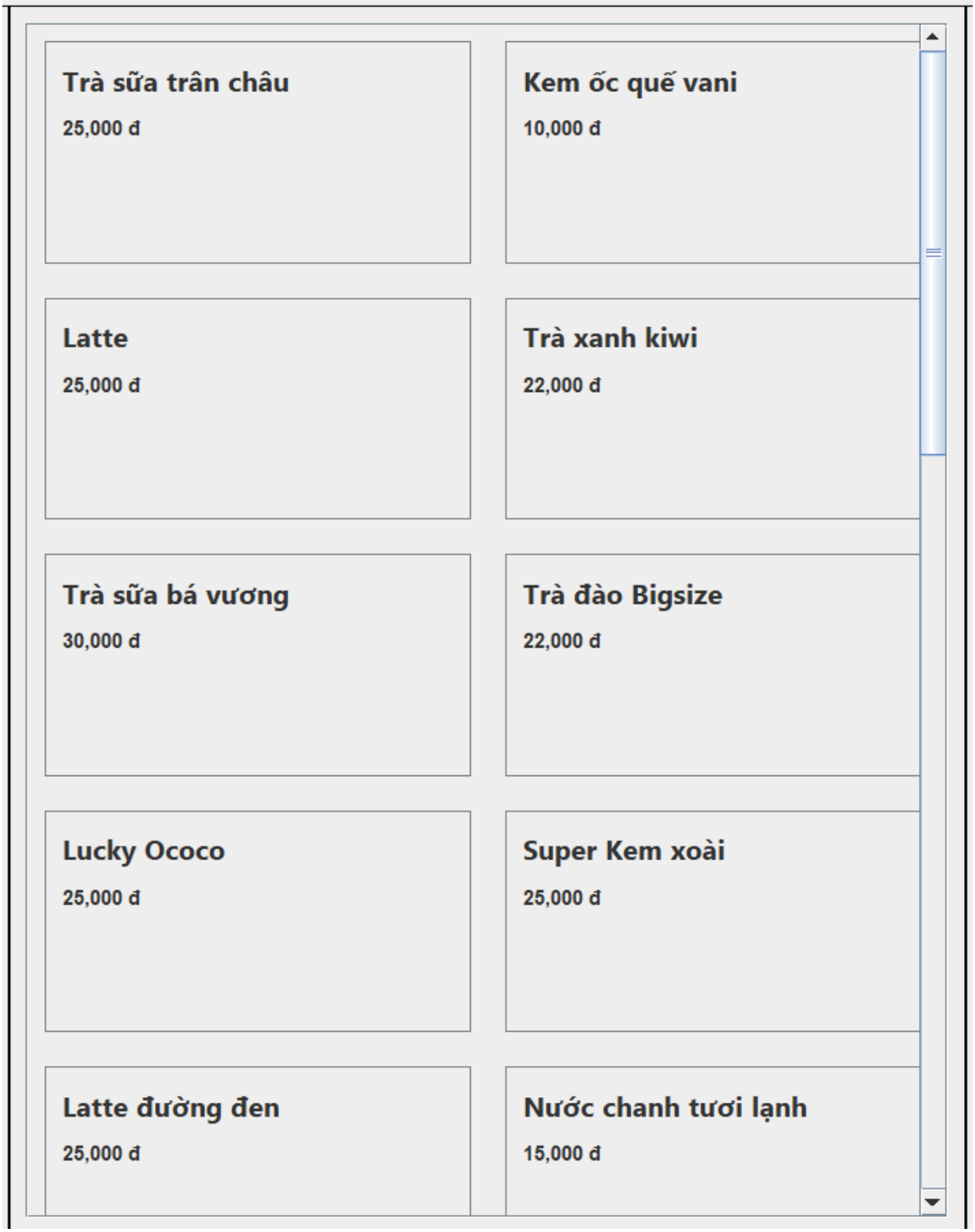

SALES SYSTEM
Ngày: 27/11/2025
Nhân viên: Tôn Thất Phong
Trà sữa trân châu
25,000 đ
Kem ốc quế vani
10,000 đ
Latte
25,000 đ
Trà xanh kiwi
22,000 đ
Trà sữa bá vương
30,000 đ
Trà đào Bigsize
22,000 đ
Lucky Ococo
25,000 đ
Super Kem xoài
25,000 đ
Latte đường đen
25,000 đ
Nước chanh tươi lạnh
15,000 đ
Kem
Topping
Trà hoa quả
Cà phê
Sữa tươi
Trà sữa
Tổng: 0 đ
THANH TOÁN
Hóa đơn
Đăng xuất

- JPanel: Chia giao diện thành các khu vực:

- Giỏ hàng:

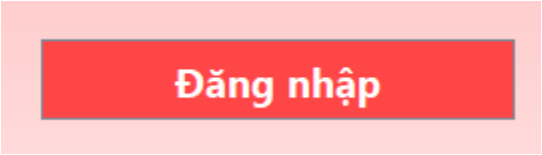


- Sản phẩm

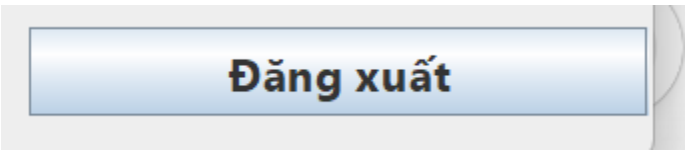

Trà sữa trân châu
25,000 đ
Kem ốc quế vani
10,000 đ
Latte
25,000 đ
Trà xanh kiwi
22,000 đ
Trà sữa bá vương
30,000 đ
Trà đào Bigsize
22,000 đ
Lucky Ococo
25,000 đ
Super Kem xoài
25,000 đ
Latte đường đen
25,000 đ
Nước chanh tươi lạnh
15,000 đ

- JLabel, JButton: Hiển thị thông tin và tạo các nút thao tác

- Đăng nhập



- Đăng xuất



- JTextField, JPasswordField:

Nhập thông tin đăng nhập và nhập liệu khác:

➔ Toàn bộ giao diện trong đồ án đều đucợ xử lý trong lớp **LoginView.java và LoginView.java**

2.3. Cơ sở dữ liệu Microsoft Access và kết nối Java với Access

- Hệ thống sử dụng file **data.accdb** để lưu trữ dữ liệu, file gồm các bảng sau:

- Bảng **SanPham** dùng đề lưu thông tin sản phẩm.
- Bảng **NhanVien** dùng để lưu tài khoản và thông tin nhân viên
- Bảng **HoaDon** dùng để lưu hóa đơn bán hàng
- Bảng **ChiTietHoaDon** dùng để lưu chi tiết từng món trong hóa đơn

- Kết nối cơ sở dữ liệu thông qua thư viện **UCanAccess**.

```java
public class KetNoiAccess {
    private static final String DB_URL = "jdbc:ucanaccess://D:\\BaitapJava\\QuanLyBanHang\\data.accdb";
    private static Connection conn = null;
    public static Connection getConnection() {
        try {
            if (conn == null || conn.isClosed()) {
                conn = DriverManager.getConnection(DB_URL);
                System.out.println("Ket Noi Aceess thanh cong!");
            }
        } catch (SQLException e) {
            System.out.println("Loi ket noi CSDL: " + e.getMessage());
        }
        return conn;
    }
```

2.4. Mô hình kiến trúc MVC

- MVC (Model – View – Controller) là mô hình kiến trúc được dùng để xây dựng ứng dụng theo hướng tách biệt giữa giao diện, dữ liệu, và xử lý logic.

- Model : Lớp dữ liệu (Data Layer)

  Gồm các lớp: **SanPham.java, NhanVien.java, HoaDon.java và ChiTietHoaDon.java**, các lớp này có nhiệm vụ chứa dữ liệu.

- View

  Giao diện người dùng (UI Layer), gồm 2 lớp:**LoginView.java và View.java**, hai lớp này được dùng để hiển thị UI, nhận thao tác người dùng.

- Contronller/Dao

  Xử lý logic và dữ liệu, trong dự án này DAO đóng vai trò như bộ điều khiển thông qua 4 lớp: **SanPhamDAO.java, NhanVienDAO.java, HoaDonDao.java và ChiTietHoaDonDAO.java**, trong 4 lớp này có nhiệm vụ thực thi SQL, kết nối database, lấy dữ liệu về và trả cho View.

2.5. Xử lý sự kiện (Event Handling)

- Ứng dụng sử dụng các sự kiện:

- ActionListener cho nút (Button).
- MouseListener cho click sản phẩm.
- KeyListener cho nhập ký tự.

CHƯƠNG 3. THIẾT KẾ GIAO DIỆN

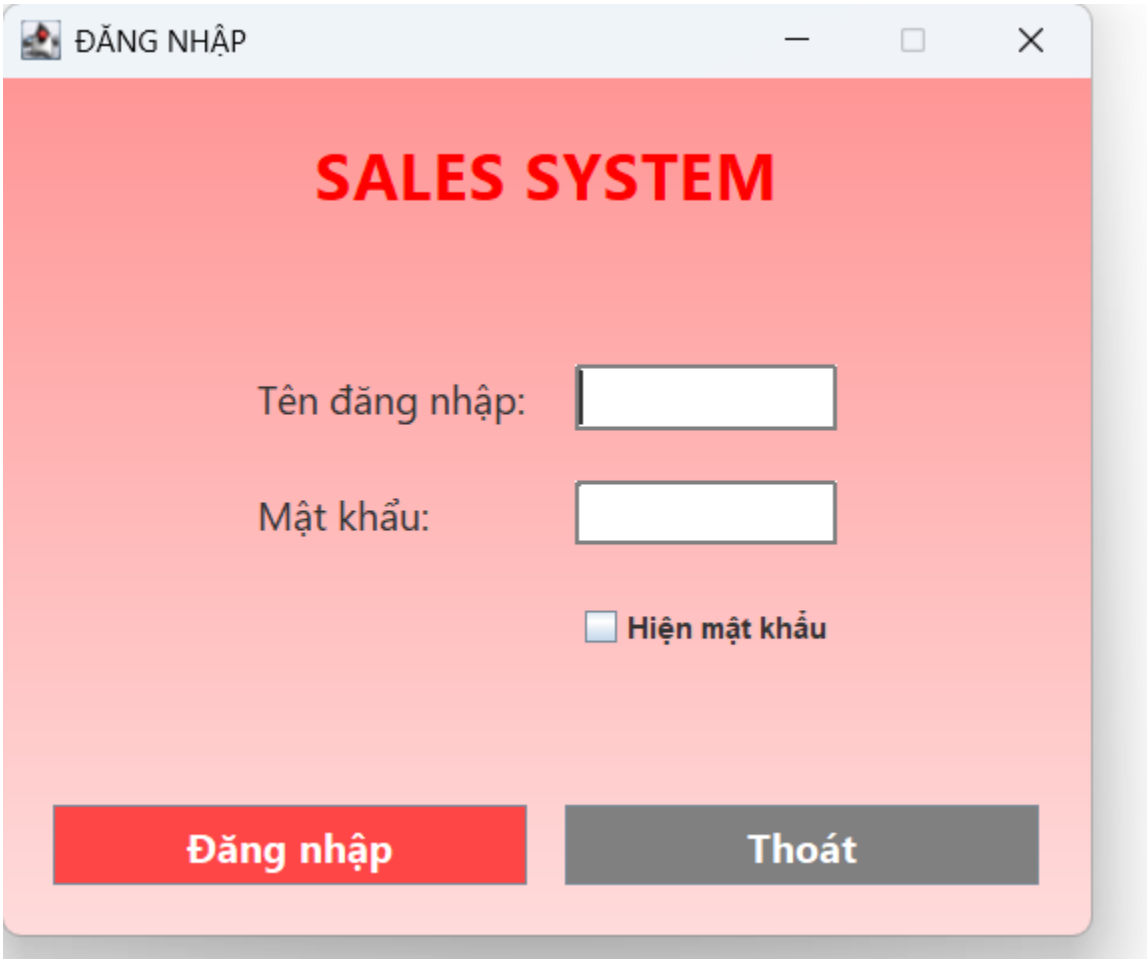
3.1. Giao diện đăng nhập hệ thống (LoginView).

- Giao diện đầu tiên của chương trình là màn hình đăng nhập, được dùng để xác thực nhân viên, kiểm soát truy cập, ngăn người lạ sử dụng hệ thống.

- Thành phần chính của giao diện gồm:

- Hai ô nhập : Tên đăng nhập và Mật khẩu.
- Nút đăng nhập
- Thông báo lỗi nếu nhập sai

→ Màn hình Login



3.2. Giao diện chính

- Sau khi đăng nhập thành công, hệ thống chuyển sang giao diện bán hàng chính. Đây là giao diện quan trọng nhất của ứng dụng.

- Thanh tiêu đề chung
  - Tên hệ thống
  - Ngày giờ
  - Tên nhân viên

➔**Header**

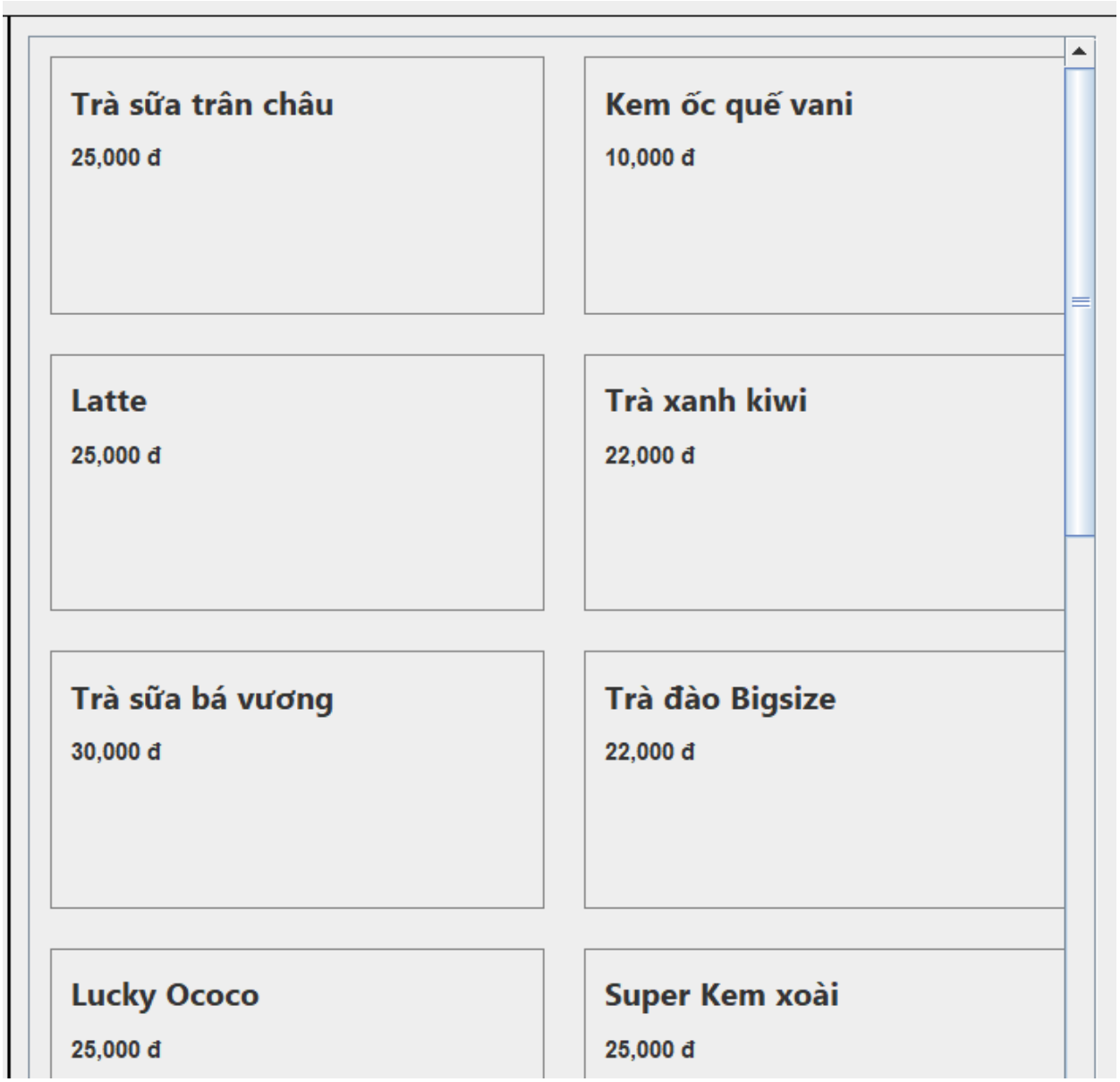
- Cột trái (Giỏ hàng):
  - Danh sách sản phẩm đã chọn.
  - Hiển thị tên sản phẩm, số lượng và giá.
  - Nút xóa sản phẩm
  - Hiển thị tổng tiền
  - Nút thanh toán

➔ Giỏ hàng

Trà sữa trân châu
[Xóa]
25,000 đ
SL: 1
Kem ốc quế vani
[Xóa]
10,000 đ
SL: 1
Trà xanh kiwi
[Xóa]
22,000 đ
SL: 1
Trà sữa bá vương
[Xóa]
30,000 đ
SL: 1
Trà đào Bigsize
[Xóa]
22,000 đ
SL: 1
Lucky Ococo
[Xóa]
25,000 đ
SL: 1
Tổng: 134,000 đ
THANH TOÁN

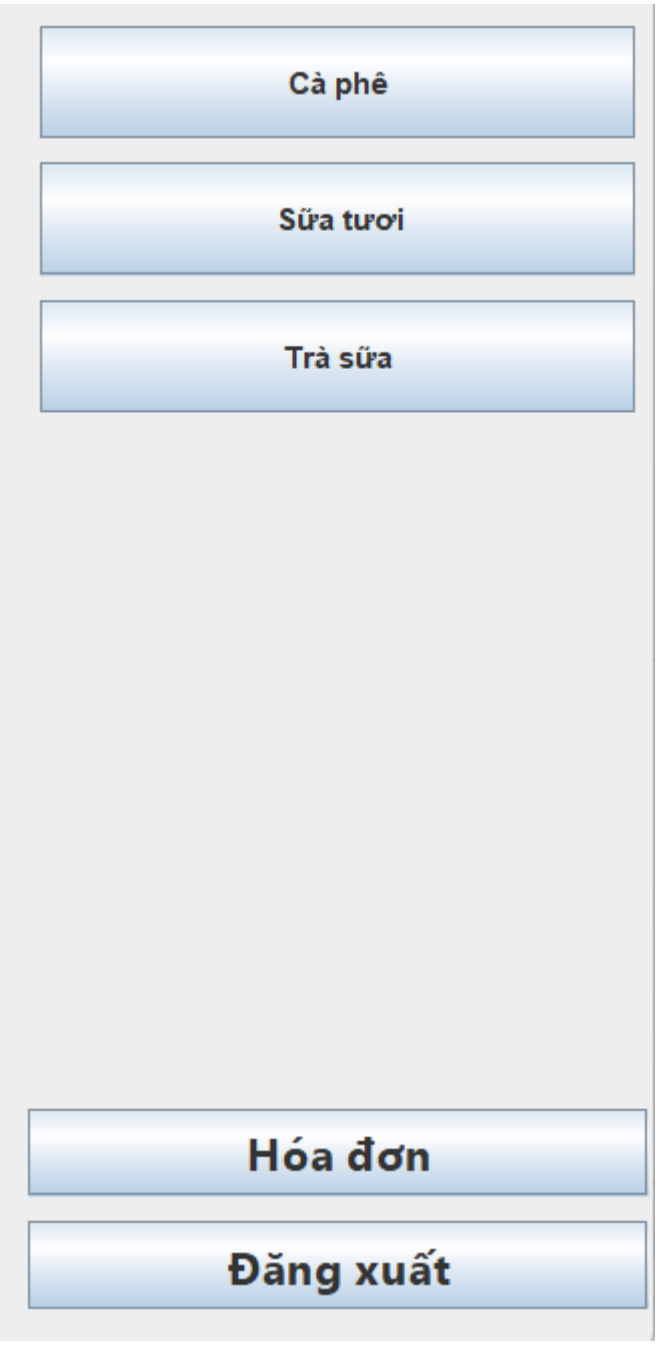
- Cột giữa (Danh sách sản phẩm):
    - Hiển thị sản phẩm được đọc từ Database Access.
    - Mỗi sản phẩm hiển thị dạng Button/Panel: tên, giá.

➔Hình ảnh Danh sách sản phẩm



- Cột phải:
    - Loại sản phẩm
    - Nút đăng xuất
    - Nút tiện ích

➔Hình ảnh giao diện cột phải



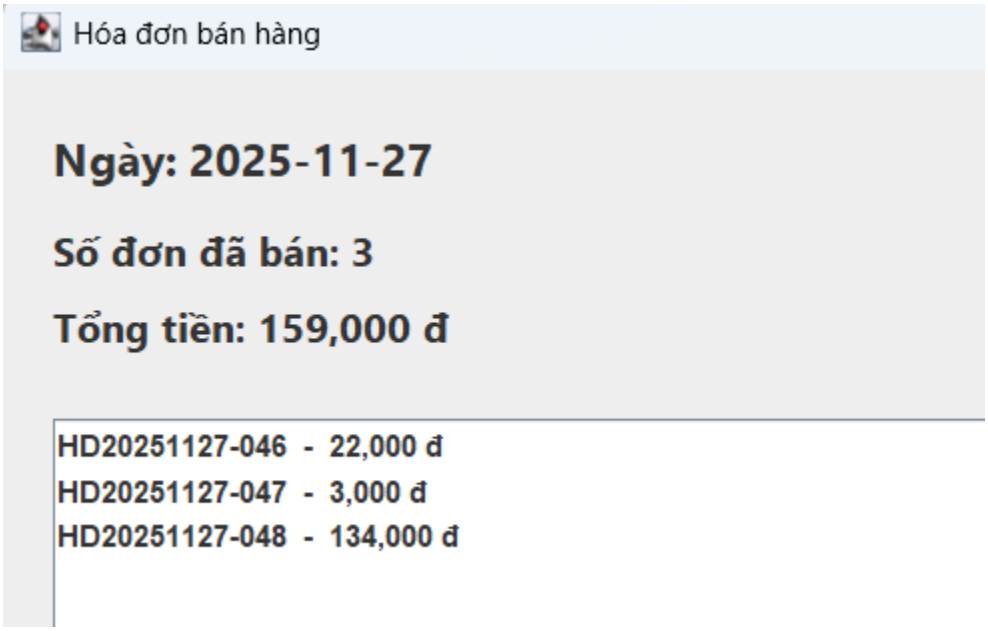
3.4. Giao diện hóa đơn

- Sau khi bấm “Thanh toán”, hệ thống:

- Tự tạo mã hóa đơn
- Lưu dữ liệu hóa đơn
- Reset giỏ hàng
- Hiển thị thông báo thành công

➔Thanh toán

➔Hóa đơn



CHƯƠNG 4. THIẾT KẾ THUẬT TOÁN

4.1. Thuật toán đăng nhập

- Kiểm tra thông tin người dùng nhập và xác thực tài khoản trong cơ sở dữ liệu.

- Quy trình thuật toán:

B1: Nhận username và password từ giao diện.

B2: Kiểm tra input có rỗng không.

B3:Gọi DAO để kiểm tra thông tin trong bảng NhanVien.

B4: Nếu tồn tại → chuyển sang giao diện chính.

B5: Nếu sai → báo lỗi.

```
private void xuLyDangNhap() {
String user = txtUser.getText().trim();
String pass = new String(txtPass.getPassword());
if (user.isEmpty() || pass.isEmpty()) {
    JOptionPane.showMessageDialog(this, "Vui lòng nhập đầy đủ thông tin!");
    return;
}
NhanVien nv = nvDAO.checkLogin(user, pass);
if (nv == null) {
    JOptionPane.showMessageDialog(this, "Sai tên đăng nhập hoặc mật khẩu!");
    return;
}
View ql = new View(nv);
ql.setVisible(true);
dispose();
```

4.2. Thuật toán lấy danh sách sản phẩm

- Đọc dữ liệu từ bảng **SanPham** và hiển thị lên giao diện bán hàng.

- Quy trình thuật toán:

B1: Kết nối database qua UcanAccess.

B2: SELECT toàn bộ sản phẩm.

B3: Chuyển từng dòng thành đối tượng SanPham.

B4: Trả về danh sách để hiển thị.

```java
public class SanPhamDAO {
    public static List<SanPham> getAllSanPham() {
        List<SanPham> list = new ArrayList<>();
        try (Connection conn = KetNoiAccess.getConnection()) {
            String sql = "SELECT * FROM sanpham";
            PreparedStatement ps = conn.prepareStatement(sql);
            ResultSet rs = ps.executeQuery();
            while (rs.next()) {
                String ma = rs.getString("maSp");
                String ten = rs.getString("tenSp");
                String loai = rs.getString("loai");
                double gia = rs.getDouble("gia");

                list.add(new SanPham(ma, ten, loai, gia));
            }
        } catch (Exception e) {
            System.out.println("Loi doc du lieu san pham: " + e.getMessage());
        }
        return list;
    }
```

4.3. Thuật toán thêm sản phẩm vào giỏ hàng.

- Khi nhân viên bấm vào một sản phẩm, sản phẩm được thêm vào giỏ hàng (hoặc tăng số lượng nếu đã tồn tại).

- Quy trình thuật toán:

B1: Nhận mã sản phẩm được click.

B2: Kiểm tra xem sản phẩm đã có trong giỏ chưa.

B3: Nếu có → tăng số lượng.

B4: Nếu chưa có → thêm mới.

B5: Cập nhật tổng tiền ngay.

```java
private void themVaoGio(SanPham sp) {

    gioHang.put(sp.getMaSP(), gioHang.getOrDefault(sp.getMaSP(), 0) + 1);

    veGioHang();
    capNhatTong();
}
```

4.4. Thuật toán tính tổng tiền.

- Tính tổng tiền của các sản phẩm trong giỏ dựa trên số lượng và giá.

- Quy trình thuật toán:

B1: Lặp qua từng sản phẩm trong giỏ.

B2: Lấy giá từ database / map sản phẩm.

B3: Nhân giá với số lượng.

B4: Cộng tất cả vào tổng tiền.

B5: Hiển thị kết quả lên giao diện.

```
private void capNhatTong() {
    double tong = 0;
    for (SanPham sp : dsSanPham) {
        if (gioHang.containsKey(sp.getMaSP())) {
            tong += sp.getGia() * gioHang.get(sp.getMaSP());
        }
    }
    lblTongTien.setText("Tổng: " + String.format("%,.0f", tong) + " đ");
}
```

4.5. Thuật toán tạo mã hóa đơn tự động.

- Sinh ra mã hóa đơn duy nhất theo ngày + số thứ tự.

- Quy trình thuật toán:

B1: Lấy ngày hiện tại dạng yyyyMMdd.

B2: Đếm số hóa đơn đã có.

B3: Tạo mã theo dạng: HD + yyyyMMdd + - + số thứ tự (3 số).

B4: Trả về chuỗi mã hóa đơn.

```
private String taoMaHD() {
    String date = LocalDate.now().format(DateTimeFormatter.ofPattern("yyyyMMdd"));
    int so = HoaDonDAO.getAll().size() + 1;
    return "HD" + date + "-" + String.format("%03d", so);
}
```

4.6. Thuật toán tạo hóa đơn và lưu chi tiết hóa đơn.

- Lưu thông tin hóa đơn vào DB khi nhân viên nhấn “Thanh toán”.

- Quy trình thuật toán:

B1: Tạo mã hóa đơn.

B2: Lưu hóa đơn vào bảng HoaDon.

B3: Lặp qua từng sản phẩm trong giỏ.

B4: Lưu từng sản phẩm vào ChiTietHoaDon.

B5: Trả về kết quả và reset giỏ hàng.

```
private void xuLyThanhToan() {

    if (gioHang.isEmpty()) {
        JOptionPane.showMessageDialog(this, "Giỏ hàng trống!");
        return;
    }
    String maHD = taoMaHD();
    double tongTien = 0;
    for (SanPham sp : dsSanPham) {
        if (gioHang.containsKey(sp.getMaSP())) {
            tongTien += sp.getGia() * gioHang.get(sp.getMaSP());
        }
    }
    HoaDon hd = new HoaDon(maHD, new java.util.Date(), tongTien);
    HoaDonDAO.luuHoaDon(hd);
    java.util.List<ChiTietHoaDon> cts = new ArrayList<>();
    for (SanPham sp : dsSanPham) {
        if (gioHang.containsKey(sp.getMaSP())) {
            int sl = gioHang.get(sp.getMaSP());
            cts.add(new ChiTietHoaDon(maHD, sp.getMaSP(), sl, sp.getGia() * sl));
        }
    }
    ChiTietHoaDonDAO.luuChiTietHoaDon(cts);
    JOptionPane.showMessageDialog(this, "Thanh toán thành công!");
    gioHang.clear();
    veGioHang();
    capNhatTong();
}
```

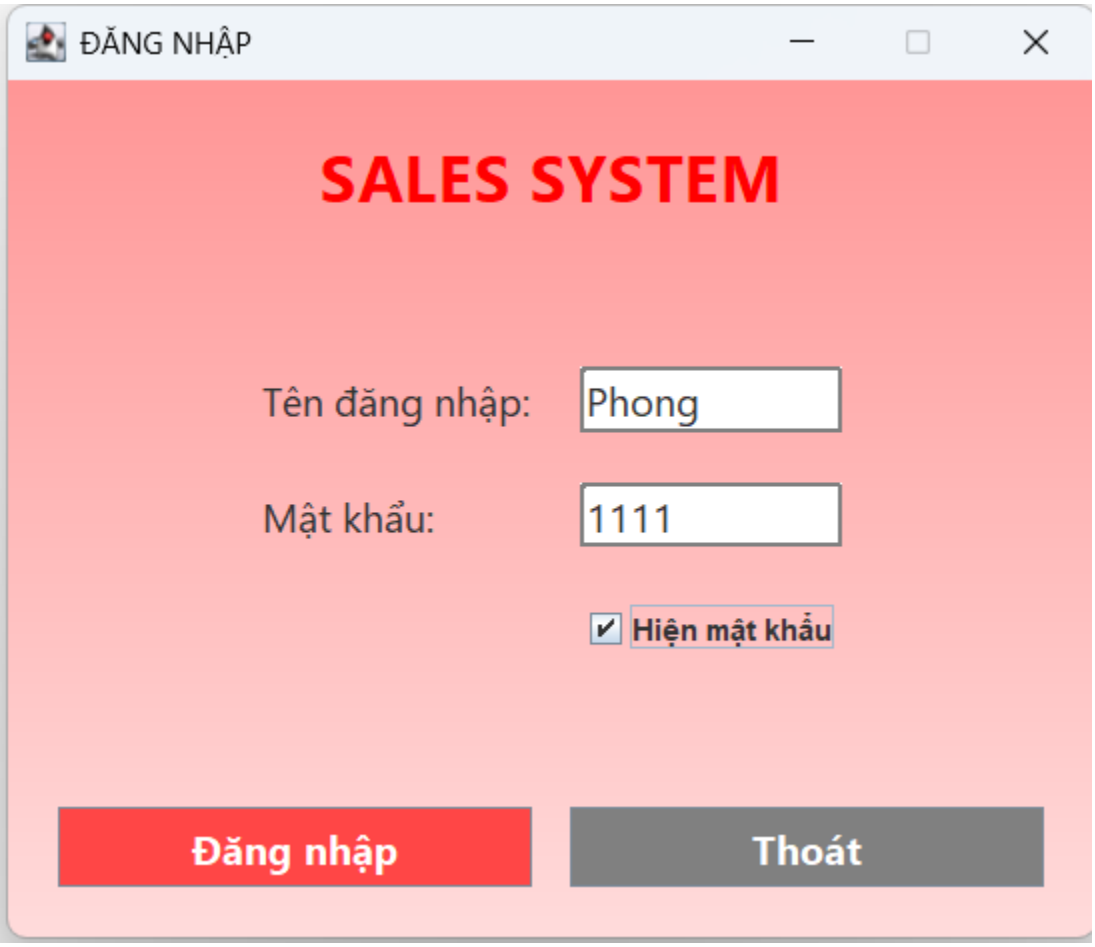
CHƯƠNG 5. DEMO ĐỒ ÁN

5.1. Khởi động ứng dụng

- Khi chạy chương trình Netbeans, giao diện đầu tiên xuất hiện là màn hình đăng nhập.

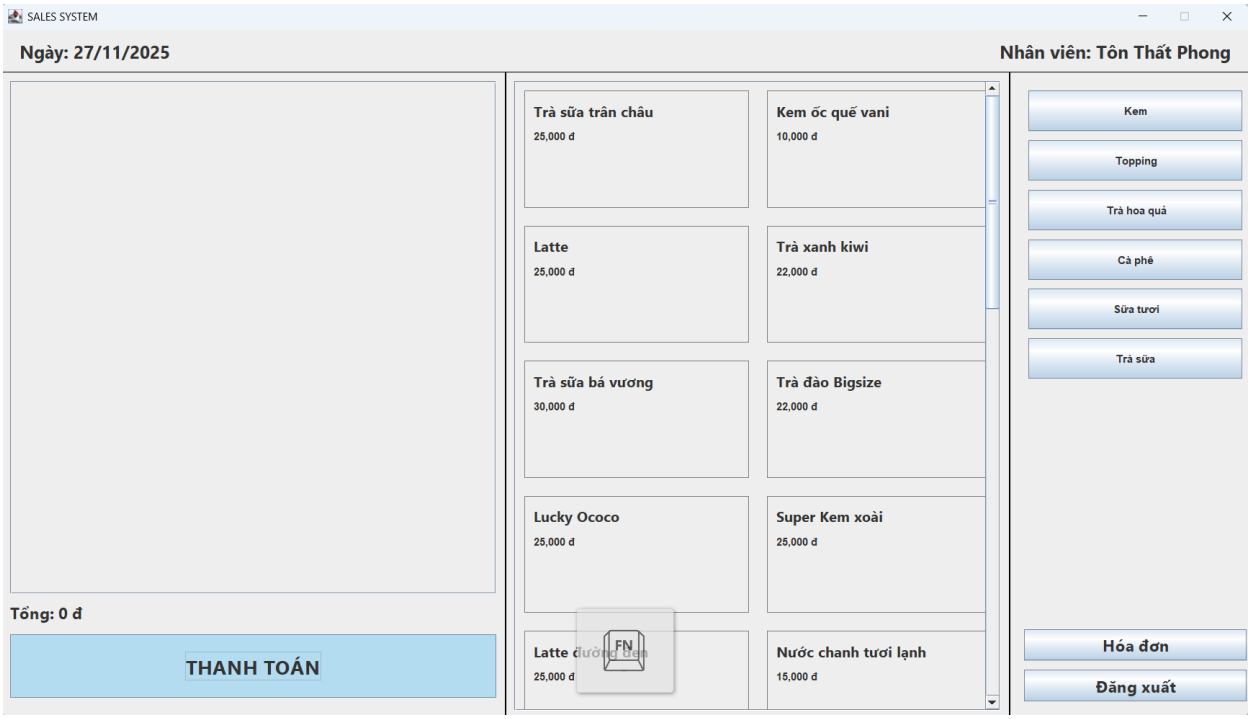
- Chức năng:

- Nhân viên nhập tên đăng nhập và mật khẩu.
- Chương trình kiểm tra thông tin qua **NhanVienDAO.**
- Nếu sai → hiển thị cảnh báo.
- Nếu đúng → chuyển sang giao diện POS.



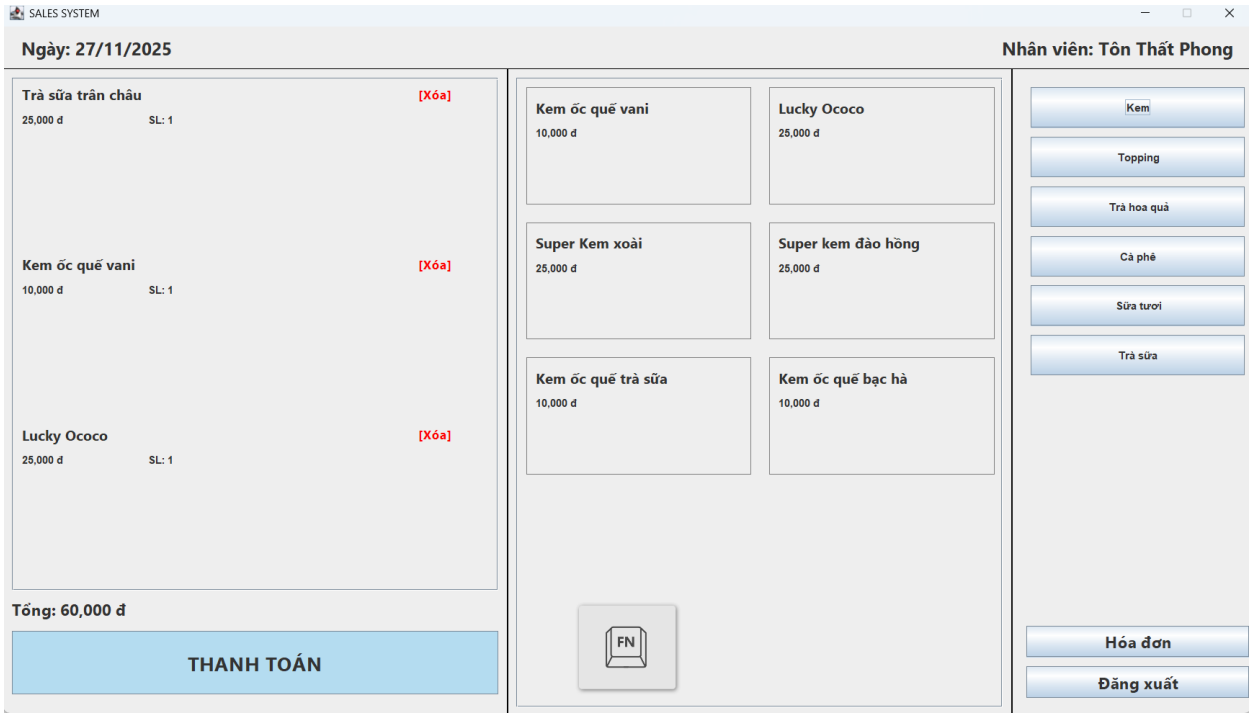
5.2. Đăng nhập thành công và chuyển sang giao diện chính

- Sau khi đăng nhập hợp lệ, hệ thống tự động mở giao diện chính.
- Giao diện chính được chia làm 4 khu vực:
  - Header – hiển thị thời gian và nhân viên đăng nhập.
  - Danh sách sản phẩm – ở giữa.
  - Giỏ hàng – phía trái.
  - Chức năng – tiện ích – phía phải.



5.3. Thao tác thêm sản phẩm vào giỏ hàng.

- Khi nhân viên **bấm chọn một sản phẩm**:
  - Sản phẩm được thêm vào giỏ hàng.
  - Nếu sản phẩm đã có từ trước → số lượng tăng lên.
  - Tổng tiền được cập nhật ngay.



5.4. Thao tác thanh toán hóa đơn.

Bước 1: Tạo mã hóa đơn tự động.

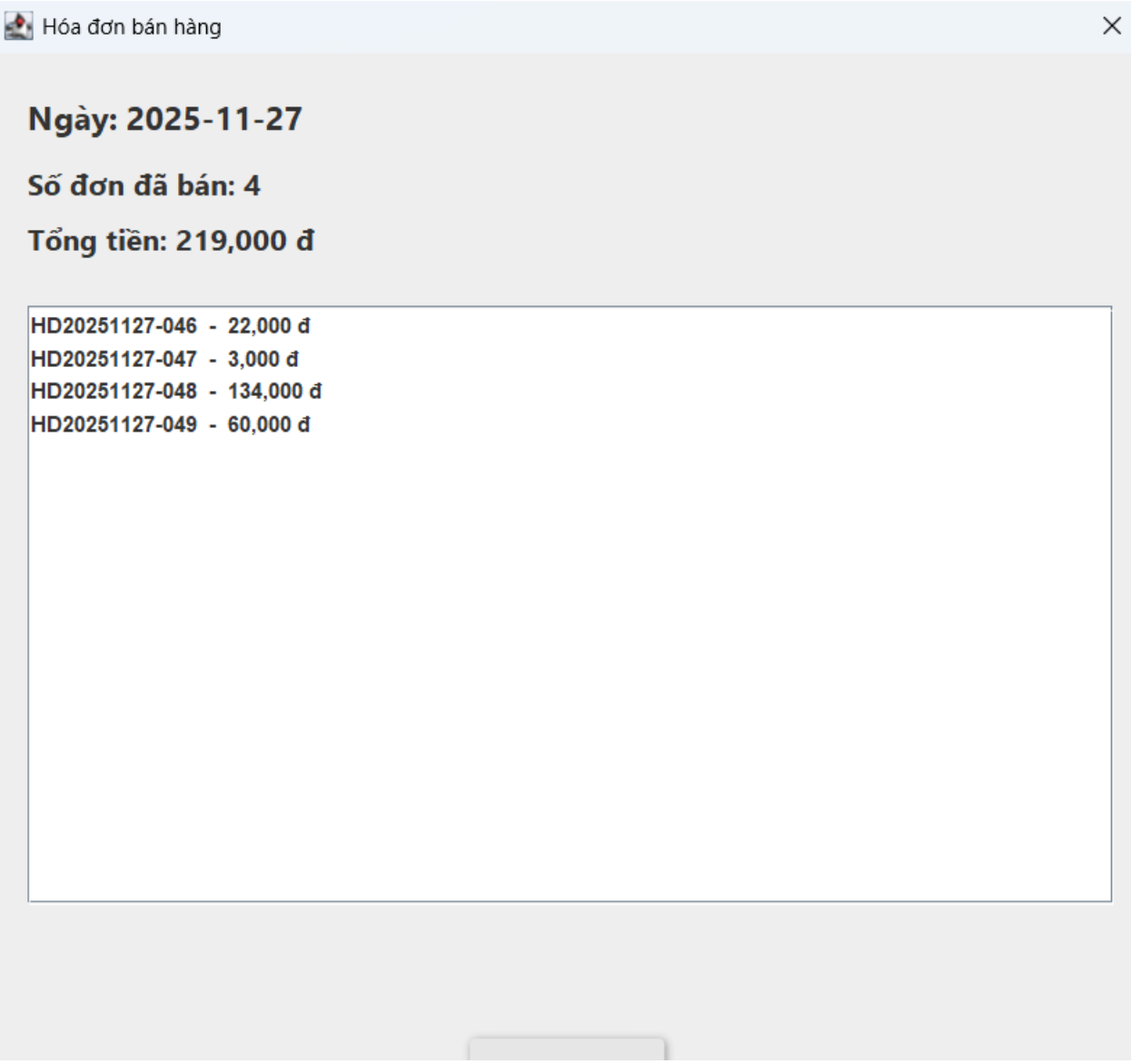
Ví dụ: HD20250112-003

Bước 2: Lưu hóa đơn vào bảng HoaDon.

Bước 3: Lưu từng sản phẩm của giỏ vào bảng ChiTietHoaDon.

Bước 4: Thông báo thanh toán thành công.

Bước 5: Làm trống giỏ hàng.



5.5. Lưu dữ liệu vào CSDL Access.

- Sau khi thanh toán, đơn hàng được lưu vào bảng HoaDon và bảng ChiTietHoaDon trong Access.

-Bảng HoaDon

ChiTietHoaDon | hoadon

maHD	ngayLap	tongTien	Click to Add
HD1762848069618	11/11/2025 3:01:09 PM	87000	
HD1762848125225	11/11/2025 3:02:05 PM	47000	
HD1762848189953	11/11/2025 3:03:09 PM	22000	
HD1762848281506	11/11/2025 3:04:41 PM	57000	
HD1762848506977	11/11/2025 3:08:26 PM	50000	
HD1762848621689	11/11/2025 3:10:21 PM	32000	
HD1762848755976	11/11/2025 3:12:35 PM	10000	
HD1762848947954	11/11/2025 3:15:47 PM	37000	
HD1762849485568	11/11/2025 3:24:45 PM	35000	

- Bảng ChiTietHoaDon

ChiTietHoaDon | hoadon

maHD	maSp	soLuong	thanhTien
HD1762849919065	SP04	1	$22,000.00
HD1762849975249	SP07	1	$25,000.00
HD1762850285409	SP02	1	$10,000.00
HD1762850285409	SP03	1	$25,000.00
HD1762850285409	SP04	1	$22,000.00
HD1762850331120	SP01	1	$25,000.00
HD1762850331120	SP02	1	$10,000.00
HD1762850331120	SP03	1	$25,000.00

5.6. Đăng xuất.

- Nhấn nút Đăng xuất:

- Quay về màn hình đăng nhập
- Đảm bảo an toàn dữ liệu
- Hạn chế người không có quyền thao tác

CHƯƠNG 6: KẾT LUẬN

6.1. Kết quả đạt được.

- Đề tài đã hoàn thành các chức năng chính sau:

• Giao diện người dùng trực quan (Java Swing)

- Đăng nhập → giao diện chính → thanh toán → thoát

• Quản lý sản phẩm

- Hiển thị danh sách sản phẩm
- Chọn nhanh bằng thao tác click
- Tự động thêm vào giỏ hàng

• Quản lý giỏ hàng

- Tăng/giảm số lượng
- Xóa sản phẩm
- Tính tổng tiền theo thời gian thực

• Quản lý hóa đơn

- Tạo mã hóa đơn tự động theo ngày
- Lưu hóa đơn và chi tiết hóa đơn vào Access
- Đảm bảo tính toàn vẹn dữ liệu

• Áp dụng mô hình MVC

- Tách biệt giao diện – xử lý – dữ liệu
- Giúp chương trình dễ bảo trì và mở rộng

Tài liệu tham khảo

https://github.com/ThatPhong28/QuanLyBanHang.git

https://ucanaccess.sourceforge.net/site.html