

BÀI TẬP LECTURE 6.

I. CÂU HỎI TRẮC NGHIỆM

Câu 1: C. Pha lấy yêu cầu

Câu 2: C. Mô hình Agile

Câu 3: C. Quan sát

Câu 4: C. 3NF

Câu 5: B. Sơ đồ tuần tự

Câu 6: B. Sơ đồ tuần tự

Câu 7: C. Cải tiến quy trình

Câu 8: A. Linh hoạt và thay đổi nhanh chóng

Câu 9: A. Hạn chế viết code lặp lại

Câu 10: B. Đảm bảo tính đóng gói và tái sử dụng

Câu 11: C. Kiểm thử có thể thực hiện song song với phát triển phần mềm

II. CÂU HỎI NGẮN

Câu 1: Phần mềm là tập hợp các chương trình và dữ liệu giúp máy tính thực hiện nhiệm vụ cụ thể. Công nghệ phần mềm là tập hợp các phương pháp, công cụ và quy trình dùng để phát triển, bảo trì phần mềm hiệu quả.

Câu 2: Các mô hình vòng đời phổ biến gồm:

- **Thác nước:** thực hiện tuần tự theo pha.
- **Xoắn ốc:** kết hợp giữa phát triển và đánh giá rủi ro.
- **Agile:** phát triển linh hoạt, chia nhỏ thành nhiều vòng lặp ngắn.

Câu 3: Các loại yêu cầu chính:

- **Yêu cầu chức năng:** mô tả các chức năng phần mềm cần thực hiện.
- **Yêu cầu phi chức năng:** liên quan đến hiệu suất, bảo mật, khả năng mở rộng.

Câu 4: Sơ đồ lớp UML mô tả các lớp, thuộc tính, phương thức và mối quan hệ giữa chúng, giúp định hình cấu trúc logic của hệ thống trong giai đoạn thiết kế.

Câu 5: Kiểm thử phần mềm giúp phát hiện lỗi, đảm bảo sản phẩm đáp ứng yêu cầu và nâng cao chất lượng trước khi bàn giao.

Câu 6: Nguyên lý **SOLID** gồm:

- S: Single Responsibility
- O: Open/Closed
- L: Liskov Substitution
- I: Interface Segregation
- D: Dependency Inversion
→ Giúp mã dễ bảo trì, mở rộng và tái sử dụng.

Câu 7:

- **Kiểm thử hộp đen:** tập trung vào đầu vào và đầu ra, không cần biết mã nguồn.
- **Kiểm thử hộp trắng:** dựa trên cấu trúc bên trong và mã chương trình.

Câu 8: Thiết kế cơ sở dữ liệu từ sơ đồ lớp UML gồm các bước:

1. Xác định các lớp và biến chúng thành bảng.
2. Xác định khóa chính, khóa ngoại.
3. Thiết lập mối quan hệ giữa các bảng.
4. Chuẩn hóa dữ liệu.

Câu 9: Ba ưu điểm của Agile:

1. Linh hoạt khi thay đổi yêu cầu.
2. Tăng tốc độ phản hồi và cải tiến liên tục.
3. Cải thiện giao tiếp giữa nhóm và khách hàng.

Câu 10: Các giai đoạn chuẩn hóa cơ sở dữ liệu:

1. 1NF – loại bỏ nhóm lặp.
2. 2NF – loại bỏ phụ thuộc bộ phận.
3. 3NF – loại bỏ phụ thuộc bắc cầu.

III. CÂU HỎI THẢO LUẬN NHÓM

Câu 1: Chúng em thấy mô hình Agile linh hoạt, chia nhỏ thành nhiều vòng lặp ngắn, phù hợp với dự án có yêu cầu thay đổi thường xuyên. Trong khi đó, Waterfall làm việc theo từng pha tuần tự, phù hợp với dự án có yêu cầu rõ ràng ngay từ đầu.

Câu 2: Sử dụng UML giúp chúng em hình dung rõ cấu trúc hệ thống, giảm sai sót khi thiết kế và tạo tài liệu chuẩn để trao đổi giữa các thành viên.

Câu 3: Để phần mềm dễ bảo trì, chúng em tuân thủ các nguyên tắc lập trình tốt như SOLID, viết tài liệu rõ ràng và tổ chức mã nguồn có cấu trúc.

Câu 4: Các công ty thường chọn mô hình phát triển lặp vì có thể kiểm tra, điều chỉnh sớm và liên tục cải tiến sản phẩm qua từng vòng lặp.

Câu 5: Kiến trúc phần mềm giúp định hình cấu trúc tổng thể, đảm bảo khả năng mở rộng, hiệu suất và tính ổn định cho hệ thống phức tạp.

Câu 6: Chúng em đảm bảo an toàn bằng cách áp dụng cơ chế mã hóa, phân quyền truy cập, kiểm thử bảo mật và tuân thủ các tiêu chuẩn an ninh.

Câu 7: Kiểm thử đơn vị tập trung vào kiểm tra từng hàm hoặc module nhỏ, còn kiểm thử tích hợp kiểm tra sự tương tác giữa các module với nhau.

Câu 8: Thách thức lớn khi thu thập yêu cầu là khách hàng mô tả không rõ ràng, thay đổi yêu cầu liên tục và khó xác định phạm vi hệ thống.

Câu 9: Khi áp dụng Scrum, chúng em chia dự án thành các sprint ngắn, họp hằng ngày, review định kỳ và điều chỉnh kế hoạch linh hoạt.

Câu 10: Việc tối ưu mã nguồn giúp tăng hiệu suất, giảm lỗi, tiết kiệm tài nguyên và dễ bảo trì, đảm bảo phần mềm hoạt động ổn định lâu dài.

IV. CÂU HỎI TÌNH HUỐNG

Câu 1: Em sẽ trao đổi lại với khách hàng để xác định yêu cầu cốt lõi, thống nhất phạm vi thay đổi và điều chỉnh kế hoạch phát triển hợp lý.

Câu 2: Em sẽ báo ngay cho trưởng nhóm và ưu tiên khắc phục lỗi trước khi bàn giao để tránh ảnh hưởng đến chất lượng sản phẩm.

Câu 3: Em sẽ đề xuất áp dụng mô hình Agile để linh hoạt điều chỉnh, chia nhỏ công việc theo sprint giúp dễ kiểm soát tiến độ.

Câu 4: Em sẽ kiểm tra lại mô hình dữ liệu, loại bỏ phần trùng lặp, chuẩn hóa bảng và rà soát quan hệ giữa các bảng để tối ưu cơ sở dữ liệu.

Câu 5: Em sẽ chọn mô hình Agile vì phù hợp với startup – giúp phản hồi nhanh, linh hoạt khi thay đổi và tiết kiệm thời gian phát triển.

Câu 6: Em sẽ tối ưu thuật toán, sử dụng cache, giảm truy vấn dư thừa và kiểm tra hiệu suất định kỳ để nâng cao tốc độ phần mềm.

Câu 7: Em sẽ cập nhật bản vá bảo mật, mã hóa dữ liệu nhạy cảm và thiết lập quyền truy cập hợp lý mà không ảnh hưởng đến người dùng.

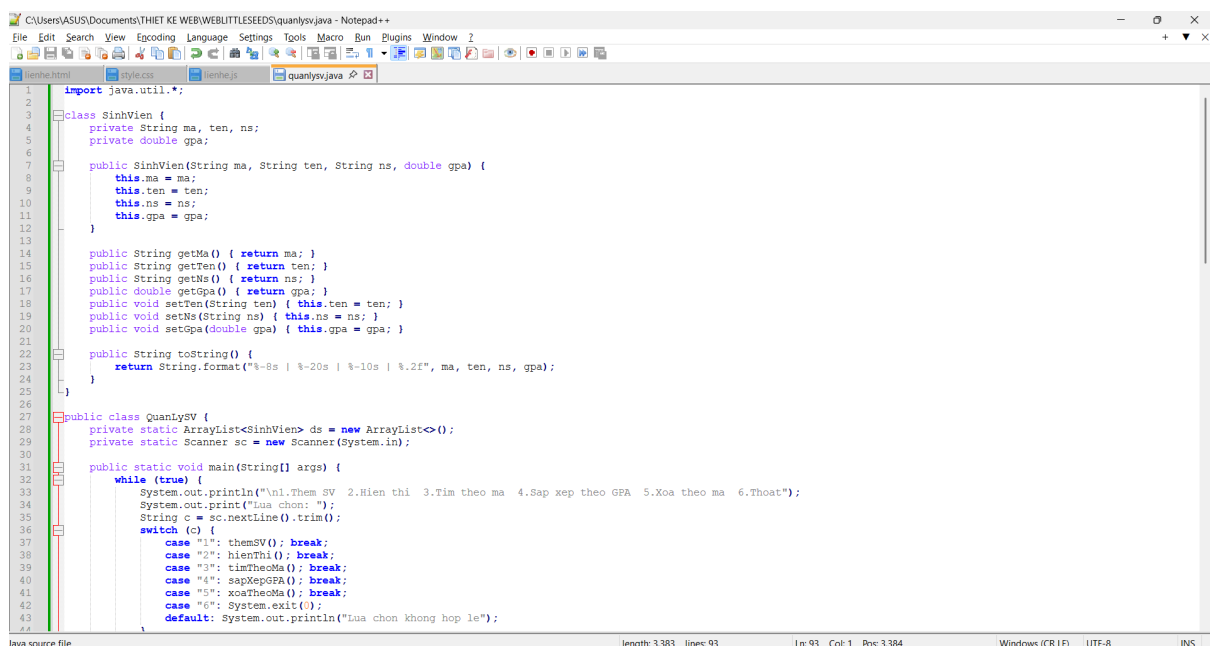
Câu 8: Em sẽ tổ chức họp nhóm, lắng nghe các ý kiến, phân tích ưu nhược điểm từng hướng và chọn giải pháp tối ưu nhất dựa trên mục tiêu chung.

Câu 9: Em sẽ nghiên cứu hành vi người dùng, thử nghiệm giao diện mẫu (prototype) và thu nhận phản hồi để điều chỉnh phù hợp, dễ sử dụng.

Câu 10: Em sẽ đọc mã nguồn, thử nghiệm các chức năng, trao đổi với đội cũ nếu có và lập lại tài liệu hướng dẫn để dễ tiếp tục phát triển.

V. THỰC HÀNH

Câu 1 — chương trình Java quản lý thông tin sinh viên (OOP) kèm menu cơ bản: thêm, hiển thị, tìm, sắp xếp theo GPA, xóa, thoát.



```
1 import java.util.*;
2
3 class SinhVien {
4     private String ma, ten, ns;
5     private double gpa;
6
7     public SinhVien(String ma, String ten, String ns, double gpa) {
8         this.ma = ma;
9         this.ten = ten;
10        this.ns = ns;
11        this.gpa = gpa;
12    }
13
14    public String getMa() { return ma; }
15    public String getTen() { return ten; }
16    public String getNs() { return ns; }
17    public double getGpa() { return gpa; }
18    public void setTen(String ten) { this.ten = ten; }
19    public void setNs(String ns) { this.ns = ns; }
20    public void setGpa(double gpa) { this.gpa = gpa; }
21
22    public String toString() {
23        return String.format("%-8s | %-20s | %-10s | %.2f", ma, ten, ns, gpa);
24    }
25
26
27    public class QuanLySV {
28        private static ArrayList<SinhVien> ds = new ArrayList<>();
29        private static Scanner sc = new Scanner(System.in);
30
31        public static void main(String[] args) {
32            while (true) {
33                System.out.println("\n1.Them SV 2.Hien thi 3.Tim theo ma 4.Sap xep theo GPA 5.Xoa theo ma 6.Thoat");
34                System.out.print("Lua chon: ");
35                String c = sc.nextLine().trim();
36                switch (c) {
37                    case "1": themSV(); break;
38                    case "2": hienThi(); break;
39                    case "3": timThecMa(); break;
40                    case "4": sapXepGPA(); break;
41                    case "5": xoaThecMa(); break;
42                    case "6": System.exit(0);
43                    default: System.out.println("Lua chon khong hop le");
44                }
45            }
46        }
47    }
48 }
```

Câu 2. Xây dựng sơ đồ lớp UML cho hệ thống quản lý thư viện.

1. Các lớp chính

Lớp Sach

- Thuộc tính:
 - maSach: String
 - tenSach: String
 - tacGia: String
 - nhaXuatBan: String
 - namXuatBan: int
 - soLuong: int
 - Phương thức:
 - nhapThongTin()
 - hienThiThongTin()
-

Lớp DocGia

- Thuộc tính:
 - maDocGia: String
 - tenDocGia: String
 - ngaySinh: String
 - diaChi: String
 - soDienThoai: String
 - Phương thức:
 - dangKy()
 - xemThongTin()
-

Lớp ThuThu

- Thuộc tính:

- maThuThu: String
 - tenThuThu: String
 - caLam: String
 - Phương thức:
 - themSach()
 - xoaSach()
 - quanLyMuonTra()
-

Lớp PhieuMuon

- Thuộc tính:
 - maPhieu: String
 - ngayMuon: String
 - ngayTra: String
 - maDocGia: String
 - maSach: String
 - Phương thức:
 - lapPhieu()
 - tinhSoNgayMuon()
-

Lớp QuanLyThuVien

- Thuộc tính:
 - dsSach: List<Sach>
 - dsDocGia: List<DocGia>
 - dsPhieu: List<PhieuMuon>

- Phương thức:
 - themDocGia()
 - muonSach()
 - traSach()
 - thongKeSach()
-

2. Mối quan hệ giữa các lớp

- DocGia 1 - n PhieuMuon
→ Một độc giả có thể có nhiều phiếu mượn.
- Sach 1 - n PhieuMuon
→ Một quyển sách có thể được mượn nhiều lần.
- QuanLyThuVien chứa các đối tượng Sach, DocGia, và PhieuMuon.
- ThuThu quản lý hoạt động mượn trả thông qua QuanLyThuVien.

Câu 3. Viết test case cho tính năng đăng ký tài khoản trong một hệ thống website.

Mục tiêu:

Kiểm tra xem chức năng **đăng ký tài khoản** hoạt động đúng yêu cầu hay không.

Các trường đầu vào:

- Họ tên
 - Email
 - Mật khẩu
 - Xác nhận mật khẩu
-

Bảng Test Case

STT	Mô tả trường hợp kiểm thử	Dữ liệu đầu vào	Kết quả mong đợi	Kết quả thực tế
1	Nhập đầy đủ thông tin hợp lệ	Họ tên: Nguyễn Văn A Email: a@gmail.com Mật khẩu: 123456 Xác nhận: 123456	Hệ thống thông báo “Đăng ký thành công” và chuyển đến trang đăng nhập	Đạt
2	Bỏ trống tất cả các trường	(Trống)	Hiện thị thông báo lỗi yêu cầu nhập thông tin	Đạt
3	Email không đúng định dạng	Email: abc@xyz	Hiện thị thông báo “Email không hợp lệ”	Đạt
4	Mật khẩu ít hơn 6 ký tự	Mật khẩu: 123 Xác nhận: 123	Hiện thị thông báo “Mật khẩu quá ngắn”	Đạt
5	Mật khẩu và xác nhận không trùng khớp	Mật khẩu: 123456 Xác nhận: 654321	Hiện thị thông báo “Mật khẩu không khớp”	Đạt
6	Email đã được đăng ký trước đó	Email: a@gmail.com	Hiện thị thông báo “Email đã tồn tại”	Đạt

7	Nhập ký tự đặc biệt trong họ tên	Họ tên: @@@	Hiện thị thông báo “Họ tên không hợp lệ”	Đạt
8	Nhập khoảng trắng đầu và cuối trong email	Email: "a@gmail.com "	Hệ thống tự loại bỏ khoảng trắng và đăng ký thành công	Đạt

Câu 4. Thiết kế cơ sở dữ liệu cho hệ thống bán hàng trực tuyến và chuẩn hóa đến 3NF.

1. Mục tiêu:

Xây dựng mô hình cơ sở dữ liệu giúp quản lý thông tin khách hàng, sản phẩm, đơn hàng và chi tiết đặt hàng một cách hiệu quả, tránh dư thừa dữ liệu.

2. Các bảng chính:

Bảng KHACHHANG

Tên cột	Kiểu dữ liệu	Ghi chú
MaKH	INT	Khóa chính
HoTen	VARCHAR(100)	Họ tên khách hàng
Email	VARCHAR(100)	Duy nhất
DiaChi	VARCHAR(255)	

SoDienThoai	VARCHAR(15)	
-------------	-------------	--

Bảng SANPHAM

Tên cột	Kiểu dữ liệu	Ghi chú
MaSP	INT	Khóa chính
TenSP	VARCHAR(100)	Tên sản phẩm
DonGia	DECIMAL(10,2)	Giá bán
SoLuongTon	INT	Số lượng còn lại
MoTa	TEXT	Mô tả sản phẩm

Bảng DONDATHANG

Tên cột	Kiểu dữ liệu	Ghi chú
MaDH	INT	Khóa chính

MaKH	INT	Khóa ngoại → KHACHHANG(MaKH)
NgayDat	DATE	Ngày đặt hàng
TongTien	DECIMAL(12,2)	Tổng tiền đơn hàng
TrangThai	VARCHAR(50)	Ví dụ: “Đã giao”, “Đang xử lý”

Bảng CHITIETDONDH

Tên cột	Kiểu dữ liệu	Ghi chú
MaDH	INT	Khóa ngoại → DONDATHANG(MaDH)
MaSP	INT	Khóa ngoại → SANPHAM(MaSP)
SoLuong	INT	Số lượng mua
DonGia	DECIMAL(10,2)	Giá tại thời điểm đặt
ThànhTiền	DECIMAL(12,2)	= SoLuong * DonGia

Khóa chính: (MaDH, MaSP)

3. Chuẩn hóa đến 3NF

1NF (Dạng chuẩn 1):

- Mỗi cột chỉ chứa giá trị nguyên tử (không chứa danh sách hoặc giá trị lặp).
→ Đã đạt.

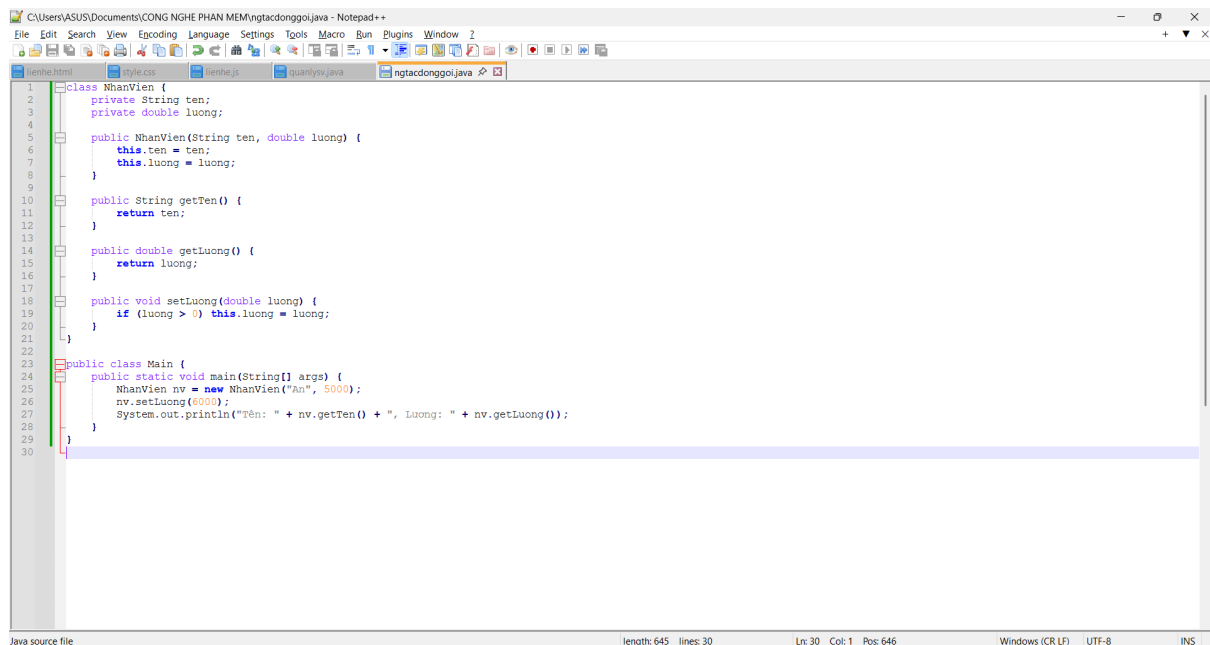
2NF (Dạng chuẩn 2):

- Tất cả thuộc tính không khóa phụ thuộc hoàn toàn vào khóa chính.
→ Trong bảng CHITIETDONDH, các cột SoLuong, DonGia phụ thuộc hoàn toàn vào (MaDH, MaSP).
→ Đã đạt.

3NF (Dạng chuẩn 3):

- Không có phụ thuộc bắc cầu (thuộc tính không khóa phụ thuộc vào thuộc tính không khóa khác).
→ Mọi bảng đều chỉ phụ thuộc vào khóa chính.
→ Đã đạt.

Câu 5. Viết một đoạn mã Java đơn giản để minh họa nguyên tắc đóng gói trong lập trình hướng đối tượng.



```
1 class NhanVien {
2     private String ten;
3     private double luong;
4
5     public NhanVien(String ten, double luong) {
6         this.ten = ten;
7         this.luong = luong;
8     }
9
10    public String getTen() {
11        return ten;
12    }
13
14    public double getLuong() {
15        return luong;
16    }
17
18    public void setLuong(double luong) {
19        if (luong > 0) this.luong = luong;
20    }
21 }
22
23 public class Main {
24     public static void main(String[] args) {
25         NhanVien nv = new NhanVien("An", 5000);
26         nv.setLuong(6000);
27         System.out.println("Tên: " + nv.getTen() + ", Luong: " + nv.getLuong());
28     }
29 }
30
```

Câu 6. Xây dựng sơ đồ tuần tự UML cho chức năng “Đăng nhập hệ thống”.

Các đối tượng tham gia:

- Người dùng
- Giao diện đăng nhập (UI)
- Bộ điều khiển (Controller)
- Cơ sở dữ liệu (Database)

Trình tự thực hiện:

1. Người dùng nhập tên đăng nhập và mật khẩu, gửi yêu cầu đăng nhập đến giao diện UI.
2. Giao diện UI chuyển thông tin đăng nhập đến Controller để xử lý.
3. Controller gửi yêu cầu kiểm tra thông tin đăng nhập đến Database.
4. Database thực hiện xác thực thông tin:
 - Nếu đúng, trả về kết quả hợp lệ.
 - Nếu sai, trả về thông báo không hợp lệ.
5. Controller nhận kết quả và gửi phản hồi lại cho UI.
6. UI hiển thị thông báo kết quả cho người dùng:
 - “Đăng nhập thành công” nếu thông tin hợp lệ.
 - “Sai tên đăng nhập hoặc mật khẩu” nếu không hợp lệ.

Kết luận:

Sơ đồ tuần tự thể hiện rõ quá trình tương tác giữa người dùng và các thành phần trong hệ thống khi thực hiện chức năng đăng nhập, giúp mô tả cụ thể luồng dữ liệu và thứ tự xử lý trong quy trình xác thực.

Câu 7. Viết chương trình kiểm thử đơn vị (Unit Test) cho một phương thức tính tổng hai số trong Java.

Mã nguồn:

```
// Lớp chính chứa phương thức cần kiểm thử
```

```

public class TinhToan {

    public int tinhTong(int a, int b) {

        return a + b;

    }

}

// Lớp kiểm thử sử dụng JUnit

import static org.junit.Assert.*;

import org.junit.Test;

public class TinhToanTest {

    @Test

    public void testTinhTong() {

        TinhToan tt = new TinhToan();

        assertEquals(5, tt.tinhTong(2, 3)); // Kiểm thử tổng dương

        assertEquals(0, tt.tinhTong(-2, 2)); // Kiểm thử tổng bằng 0

        assertEquals(-7, tt.tinhTong(-3, -4)); // Kiểm thử tổng âm

    }

}

```

Câu 8. Tạo kịch bản kiểm thử chức năng thanh toán trong hệ thống thương mại điện tử.

Bảng kịch bản kiểm thử:

STT	Mô tả kiểm thử	Dữ liệu đầu vào	Kết quả mong đợi
-----	----------------	-----------------	------------------

1	Người dùng thanh toán bằng thẻ hợp lệ	Thông tin thẻ: hợp lệ, đủ số dư	Thanh toán thành công, hiển thị thông báo “Đặt hàng thành công”
2	Người dùng thanh toán bằng thẻ hết hạn	Thông tin thẻ: hết hạn	Hiển thị thông báo lỗi “Thẻ không hợp lệ”
3	Người dùng không nhập thông tin thanh toán	Trường thanh toán để trống	Hệ thống yêu cầu nhập thông tin thanh toán
4	Người dùng chọn phương thức thanh toán COD (trả tiền khi nhận hàng)	Phương thức: COD	Đặt hàng thành công, trạng thái đơn hàng “Chờ giao hàng”
5	Người dùng thanh toán bằng ví điện tử không đủ số dư	Số dư < giá trị đơn hàng	Hiển thị thông báo lỗi “Số dư không đủ”
6	Kết nối mạng bị gián đoạn trong khi thanh toán	Mất kết nối Internet	Hiển thị thông báo lỗi “Lỗi kết nối, vui lòng thử lại”
7	Thanh toán thành công, kiểm tra email xác nhận đơn hàng	Email người dùng hợp lệ	Người dùng nhận được email xác nhận đơn hàng

Câu 9. Xây dựng mô hình thực thể (ERD) cho hệ thống quản lý bệnh viện.

Các thực thể và thuộc tính chính:

1. Bệnh nhân (BenhNhan)

- MaBN (PK)

- HoTen
- NgaySinh
- GioiTinh
- DiaChi
- SDT

2. Bác sĩ (BacSi)

- MaBS (PK)
- HoTen
- Khoa
- SDT
- ChucVu

3. Khoa (Khoa)

- MaKhoa (PK)
- TenKhoa
- ViTri

4. Hồ sơ khám bệnh (HoSoKham)

- MaHoSo (PK)
- NgayKham
- ChanDoan
- MaBN (FK)
- MaBS (FK)

5. Thuốc (Thuoc)

- MaThuoc (PK)
- TenThuoc
- DonViTinh
- Gia

6. Đơn thuốc (DonThuoc)

- MaDon (PK)
- MaHoSo (FK)
- NgayLap

7. Chi tiết đơn thuốc (ChiTietDonThuoc)

- MaDon (FK)
- MaThuoc (FK)
- SoLuong

Các mối quan hệ:

- **Bệnh nhân – Hồ sơ khám bệnh:** 1–N (một bệnh nhân có thể có nhiều hồ sơ khám)
- **Bác sĩ – Hồ sơ khám bệnh:** 1–N (một bác sĩ có thể khám cho nhiều bệnh nhân)
- **Khoa – Bác sĩ:** 1–N (mỗi khoa có nhiều bác sĩ)
- **Hồ sơ khám bệnh – Đơn thuốc:** 1–1 hoặc 1–N (mỗi hồ sơ có thể có một hoặc nhiều đơn thuốc)
- **Đơn thuốc – Chi tiết đơn thuốc – Thuốc:** N–N (thông qua bảng ChiTietDonThuoc)

Câu 10. Phân tích và mô tả một use case cụ thể trong hệ thống đặt vé máy bay.

Use Case được chọn: *Đặt vé máy bay trực tuyến*

1. Tên Use Case:

Đặt vé máy bay

2. Mục tiêu:

Cho phép khách hàng tìm kiếm, chọn chuyến bay và đặt vé thành công thông qua hệ thống.

3. Tác nhân (Actor):

- **Khách hàng:** người sử dụng hệ thống để đặt vé.
- **Hệ thống thanh toán:** xử lý giao dịch thanh toán.

4. Mô tả tóm tắt:

Khách hàng truy cập hệ thống, tìm chuyến bay theo ngày và điểm đến, chọn chuyến bay phù hợp, nhập thông tin cá nhân, sau đó thanh toán để hoàn tất việc đặt vé.

5. Luồng sự kiện chính:

1. Khách hàng truy cập vào hệ thống.
 2. Hệ thống hiển thị giao diện tìm kiếm chuyến bay.
 3. Khách hàng nhập thông tin: điểm đi, điểm đến, ngày khởi hành, số lượng vé.
 4. Hệ thống hiển thị danh sách các chuyến bay phù hợp.
 5. Khách hàng chọn chuyến bay mong muốn.
 6. Khách hàng nhập thông tin cá nhân và chọn phương thức thanh toán.
 7. Hệ thống kết nối với hệ thống thanh toán để xử lý giao dịch.
 8. Sau khi thanh toán thành công, hệ thống gửi vé điện tử qua email cho khách hàng.
-

6. Luồng thay thế (nếu có):

- Nếu thanh toán thất bại, hệ thống thông báo lỗi và yêu cầu khách hàng chọn phương thức thanh toán khác.
- Nếu không có chuyến bay phù hợp, hệ thống hiển thị thông báo “Không tìm thấy chuyến bay”.

7. Kết quả:

Khách hàng nhận được vé điện tử và thông tin đặt chỗ thành công qua email.

8. Điều kiện tiên quyết:

Khách hàng phải có tài khoản và đăng nhập vào hệ thống.

9. Điều kiện sau cùng:

Hệ thống cập nhật thông tin đặt vé và trừ số lượng ghế trống của chuyến bay tương ứng.