

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC VIỆT NHẬT



**XÂY DỰNG PHẦN MỀM
QUẢN LÝ PHÒNG KHÁM**

**BÀI TẬP LỚN
MÔN HỌC: LẬP TRÌNH NÂNG CAO**

Sinh viên thực hiện: Trần Thành Đạt
Mã số sinh viên : 23110303
Giảng viên : Chu Thị Minh Huệ

HÀ NỘI – 2026

MỤC LỤC

I.	YÊU CẦU	2
1.	Yêu cầu người dùng	2
2.	Phân tích các chức năng	2
II.	MÔ HÌNH THỰC THẾ	3
1.	Tổng quan cơ sở dữ liệu	3
2.	Danh sách các thực thể (Entities)	3
3.	Cài đặt chi tiết (Implementation)	5
III.	THIẾT KẾ CÁC THÀNH PHẦN PHẦN MỀM	9
1.	Kiến trúc tổng thể	9
2.	Tổ chức Dự án (Solution Structure)	10
3.	Ví dụ minh họa: Mô tả quá trình thực hiện chức năng ‘Đăng nhập’	10
4.	Kỹ thuật Transaction	11
5.	Truy vấn dữ liệu	13
IV.	CHƯƠNG TRÌNH PHẦN MỀM	14
1.	Màn hình đăng nhập	14
2.	Màn hình chính	14
3.	Phân quyền Admin	15
3.1.	Màn hình quản lý nhân viên	15
3.2.	Màn hình quản lý thuốc	15
3.3.	Màn hình quản lý các dịch vụ khám	16
3.4.	Màn hình thống kê lịch sử khám và doanh thu	16
4.	Phân quyền Nhân viên (Staffs)	17
4.1.	Màn hình quản lý bệnh nhân & Đăng ký khám	17
4.2.	Màn hình thanh toán sau khi khám	18
5.	Phân quyền Bác sĩ – Màn hình khám bệnh	18

I. YÊU CẦU

1. Yêu cầu người dùng

Phần mềm được xây dựng nhằm phục vụ nhu cầu quản lý số liệu và quy trình khám chữa bệnh tại một phòng khám tư nhân. Phần mềm đảm nhận giải quyết các vấn đề sau cho người dùng:

- **Quản lý tập trung:** Giảm thiểu/thay thế việc sử dụng sổ sách, giấy tờ bằng các cơ sở dữ liệu số hóa. Hỗ trợ việc tra cứu, quản lý hồ sơ bệnh nhân nhanh chóng, chính xác.
- **Quy trình khép kín:** Hỗ trợ công việc tiếp đón bệnh nhân, phân bổ chỉ định bác sĩ kê đơn và trừ tồn kho thuốc tự động.
- **Bảo mật và phân quyền:** Phân bổ các quyền truy cập theo đúng vai trò chức năng chuyên môn. Đảm bảo việc quản lý hành chính, bệnh nhân, tồn kho,... Không gây nhầm lẫn, sai sót hay mất mát về dữ liệu.
- **Hiệu năng:** Hệ thống phải hoạt động mượt mà, trơn tru khi tìm kiếm dữ liệu lớn hoặc thống kê báo cáo doanh thu.

2. Phân tích các chức năng

Phần mềm được chia thành các nhóm chức năng chính kèm theo các thao tác với dữ liệu (CRUD):

- a. Quản trị hệ thống & Danh mục:
 - Đăng nhập (Login): Xác thực người dùng, quản lý bảo mật cũng như phân quyền tài khoản dựa trên vai trò (Admin/Doctor/Staff).
 - Quản lý kho thuốc: Thêm mới, cập nhật giá bán/giá nhập/số lượng thuốc, xóa thuốc trong kho.
 - Quản lý người dùng: Quản lý danh sách các bác sĩ và nhân viên (Chỉ dành cho vai trò Quản trị viên).
- b. Chức năng dành cho nhân viên (Staffs)
 - Quản lý bệnh nhân: Thêm mới hồ sơ, cập nhật các thông tin liên hệ.
 - Tìm kiếm bệnh nhân: Tìm kiếm theo thông tin tên, số điện thoại liên lạc của bệnh nhân.

- Đăng ký khám cho bệnh nhân: Tạo phiếu khám, chỉ định bác sĩ khám kèm dịch vụ khám cho bệnh nhân. (Sử dụng Transaction nhằm đảm bảo việc tạo phiếu khám, lấy thông tin về dịch vụ diễn ra đồng thời).
- c. Chức năng dành cho bác sĩ (Doctors)
 - Khám bệnh & Ké đơn thuốc: Bác sĩ có quyền xem danh sách bệnh nhân đang chờ khám, đưa ra việc chẩn đoán bệnh sau quá trình khám và kê thuốc dựa trên kết quả chẩn đoán. (Sử dụng Transaction để cập nhật trạng thái phiếu khám, lưu đơn thuốc cũng như trừ số lượng thuốc tồn kho nếu có kê đơn).
 - Xem lịch sử khám bệnh: Cho phép kiểm tra lại các ca khám của các bệnh nhân cũ.
- d. Báo cáo, thống kê
 - Thống kê doanh thu và số lượt khám theo ngày trong khoảng thời gian tùy chọn (Chỉ dành cho Admins).

II. MÔ HÌNH THỰC THẾ

1. Tổng quan cơ sở dữ liệu

Phần mềm sử dụng Hệ quản trị cơ sở dữ liệu SQL Server, khâu thiết kế, truy vấn và tương tác dữ liệu được thực hiện thông qua Entity Framework (Entity Framework Core) theo phương pháp Code First. Các bảng (Table) của Database được thiết kế/định nghĩa thông qua các lớp (Class) trong component “Entity” thay vì được thiết kế trực tiếp trong SQL Server. Cấu trúc Database được tự động sinh ra thông qua cơ chế ‘Create Migration’.

2. Danh sách các thực thể (Entities)

a. Nhóm quản trị hệ thống

- Role (tblRoles): Quản lý vai trò của các tài khoản tương ứng với người dùng (VD: Admin, Bác sĩ, Lễ tân).
- User (tblUsers): Lưu trữ thông tin của nhân viên, tài khoản đăng nhập và liên kết với bảng Role.

b. Nhóm quản lý Bệnh nhân

- Patient (tblPatients): Lưu trữ hồ sơ hành chính của bệnh nhân như Họ tên, Giới tính, Năm sinh, Địa chỉ, Số điện thoại,..

c. Nhóm kho dược & Dịch vụ

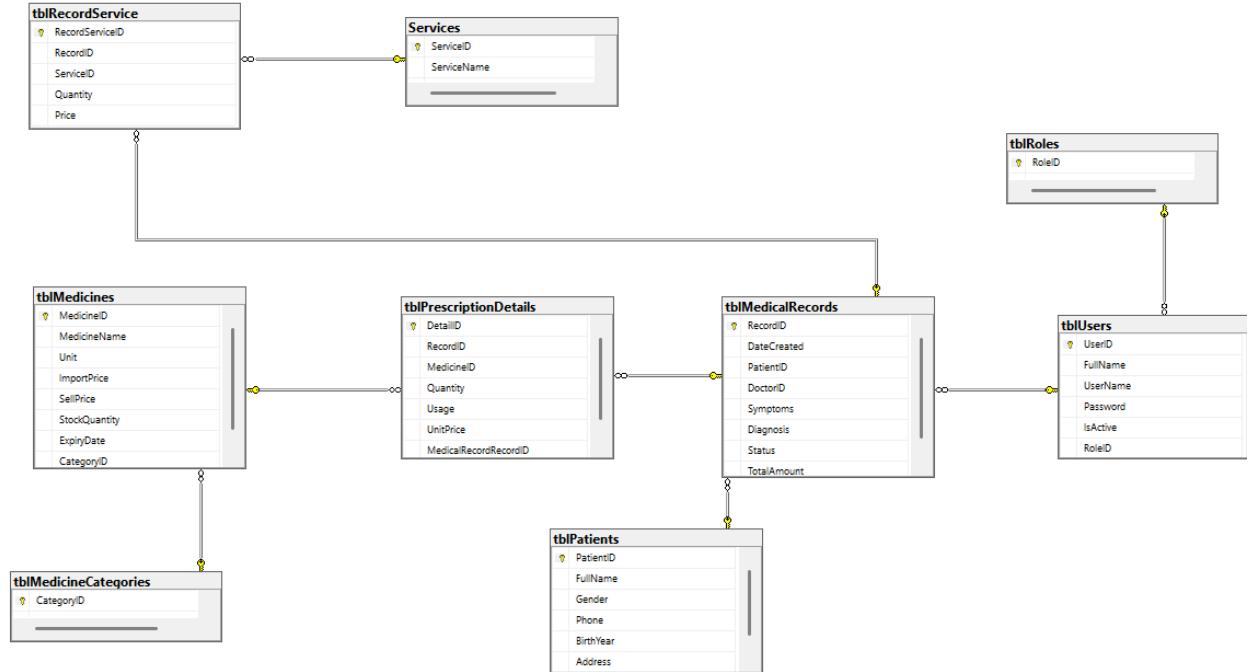
- MedicineCategory (tblMedicineCategories): Danh mục các loại thuốc.
- Medicine (tblMedicines): Lưu trữ thông tin chi tiết về các loại thuốc như Tên thuốc, Giá bán/Giá nhập, Số lượng tồn kho, Hạn sử dụng,..
- Service (tblServices): Danh sách các loại dịch vụ khám chữa bệnh (Siêu âm, Xét nghiệm,...) và bảng giá của chúng.

d. Nhóm khám chữa bệnh

- MedicalRecord (tblMedicalRecords): Phiếu khám bệnh bao gồm các thông tin của bệnh nhân: Họ tên, Địa chỉ, Triệu chứng,... và liên kết với Bác sĩ.
- RecordService (tblRecordService): Bảng lưu các dịch vụ đã sử dụng trong một lần khám.
- PrescriptionDetail (tblPrescriptionDetails): Bảng chi tiết lưu trữ đơn thuốc được kê cho bệnh nhân bởi bác sĩ trong lần khám trước đó.

e. Các lớp DTO:

- Các lớp như Statistics và History được xây dựng nhằm lưu trữ dữ liệu tổng hợp phục vụ cho việc hiển thị thống kê và lịch sử khám, không tác động trực tiếp vào bảng CSDL.



Hình 1. Sơ đồ quan hệ thực thể

3. Cài đặt chi tiết (Implementation)

a. Cấu trúc lớp thực thể (Entity Class).

Một vài ví dụ về cấu trúc các lớp thực thể:

- VD1: Lớp ‘Medicine’:

```
namespace Entities
{
    [Table("tblMedicines")]
    public class Medicine
    {
        [Key]
        public int MedicineID { get; set; }

        [Required]
        [StringLength(100)]
        public string MedicineName { get; set; }

        public string Unit { get; set; }
        public decimal ImportPrice { get; set; }
        public decimal SellPrice { get; set; }
        public int StockQuantity { get; set; }
        public DateTime? ExpiryDate { get; set; }

        public int CategoryID { get; set; }
        public virtual MedicineCategory Category { get; set; }

        public virtual ICollection<PrescriptionDetail> PrescriptionDetails { get; set; }
    }
}
```

- VD2: Lớp ‘PrescriptionDetail’. Thực thể đại diện cho bảng chi tiết về đơn thuốc, thể hiện mối quan hệ Một – Nhiều (One to Many) với bảng ‘MedicalRecord’ và bảng ‘Medicine’.

```
namespace Entities
{
    [Table("tblPrescriptionDetails")]
    10 references
    public class PrescriptionDetail
    {
        [Key]
        0 references
        public int DetailID { get; set; }

        4 references
        public int RecordID { get; set; }
        [ForeignKey("RecordID")]
        1 reference
        public virtual MedicalRecord MedicalRecord { get; set; }

        2 references
        public int MedicineID { get; set; }
        [ForeignKey("MedicineID")]
        5 references
        public virtual Medicine Medicine { get; set; }

        8 references
        public int Quantity { get; set; }

        [StringLength(200)]
        2 references
        public string Usage { get; set; }

        7 references
        public decimal UnitPrice { get; set; }
    }
}
```

- VD3: Lớp ‘User’ minh họa cho việc sử dụng các Data Annotations như [Required] và [StringLength] để ràng buộc dữ liệu.

```

namespace Entities
{
    [Table("tblUsers")]
    20 references
    public class User
    {
        [Key]
        6 references
        public int UserID { get; set; }

        [Required]
        [StringLength(100)]
        11 references
        public string FullName { get; set; }

        [Required]
        [StringLength(50)]
        12 references
        public string UserName { get; set; }

        [Required]
        11 references
        public string Password { get; set; }

        6 references
        public bool IsActive { get; set; } = true;

        11 references
        public int RoleID { get; set; }
        4 references
        public virtual Role Role { get; set; }
    }
}

```

b. Cấu hình quan hệ và ràng buộc (Fluent API)

```

namespace DAL
{
    29 references
    public class ClinicDbContext : DbContext
    {
        // --- 1. KHAI BÁO CÁC BẢNG (DBSETS) ---
        1 reference
        public DbSet<Role> Roles { get; set; }
        11 references
        public DbSet<User> Users { get; set; }

        7 references
        public DbSet<Patient> Patients { get; set; }
        9 references
        public DbSet<Service> Services { get; set; } // Mới thêm

        10 references
        public DbSet<Medicine> Medicines { get; set; }

        1 reference
        public DbSet<MedicineCategory> MedicineCategories { get; set; }
        7 references
        public DbSet<MedicalRecord> MedicalRecords { get; set; }
        3 references
        public DbSet<PrescriptionDetail> PrescriptionDetails { get; set; }
        3 references
        public DbSet<RecordService> RecordServices { get; set; }
    }
}

```

```
0 references
protected override void OnModelCreating(ModelBuilder modelBuilder)
{
    // --- A. CẤU HÌNH QUAN HỆ (FLUENT API) ---

    // 1. Cấu hình MedicalRecord - Doctor (User)
    modelBuilder.Entity<MedicalRecord>()
        .HasOne(m => m.Doctor)
        .WithMany()
        .HasForeignKey(m => m.DoctorID)
        .OnDelete(DeleteBehavior.Restrict);

    // 2. Cấu hình MedicalRecord - Patient
    // Mục đích: Một bệnh nhân có nhiều phiếu khám
    modelBuilder.Entity<MedicalRecord>()
        .HasOne(m => m.Patient)
        .WithMany(p => p.MedicalRecords)
        .HasForeignKey(m => m.PatientID);

    // 3. Cấu hình PrescriptionDetail - MedicalRecord
    modelBuilder.Entity<PrescriptionDetail>()
        .HasOne(pd => pd.MedicalRecord)
        .WithMany()
        .HasForeignKey(pd => pd.RecordID)
        .OnDelete(DeleteBehavior.Cascade);

    // 4. Cấu hình User - Role
    modelBuilder.Entity<User>()
        .HasOne(u => u.Role)
        .WithMany(r => r.Users)
        .HasForeignKey(u => u.RoleID);

    // 5. Cấu hình RecordService - MedicalRecord
    modelBuilder.Entity<MedicalRecord>()
        .Property(m => m.TotalAmount)
        .HasColumnType("decimal(18,2)");
}
```

```
// 6. Cấu hình PrescriptionDetail - Medicine
modelBuilder.Entity<PrescriptionDetail>()
    .Property(p => p.UnitPrice)
    .HasColumnType("decimal(18,2)");

// 7. Cấu hình RecordService - Service
modelBuilder.Entity<Service>()
    .Property(s => s.Price)
    .HasColumnType("decimal(18,2)");

// 8. Cấu hình Medicine - MedicineCategory
modelBuilder.Entity<Medicine>()
    .Property(m => m.ImportPrice)
    .HasColumnType("decimal(18,2)");

modelBuilder.Entity<Medicine>()
    .Property(m => m.SellPrice)
    .HasColumnType("decimal(18,2)");

modelBuilder.Entity<Medicine>()
    .HasOne(m => m.Category)
    .WithMany(c => c.Medicines)
    .HasForeignKey(m => m.CategoryID);

//Cấu hình RecordService - MedicalRecord
modelBuilder.Entity<RecordService>()
    .HasOne(rs => rs.MedicalRecord)
    .WithMany(m => m.RecordServices)
    .HasForeignKey(rs => rs.RecordID)
    .OnDelete(DeleteBehavior.Cascade);

modelBuilder.Entity<RecordService>()
    .Property(rs => rs.Price)
    .HasColumnType("decimal(18,2)");
```

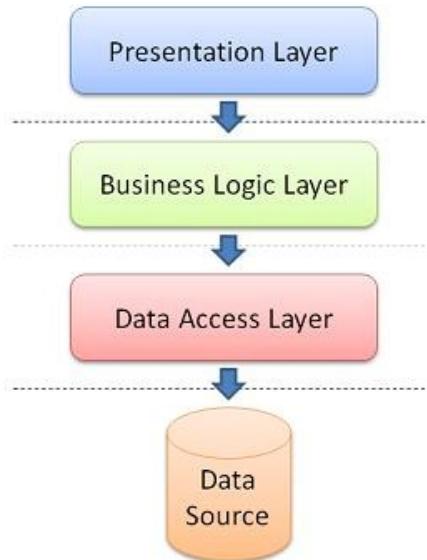
III. THIẾT KẾ CÁC THÀNH PHẦN PHẦN MỀM

1. Kiến trúc tổng thể.

Hệ thống được xây dựng theo mô hình Kiến trúc 3 lớp (3 – Layer Architechture) kết hợp với mô hình Entity Data Model và các tầng được tách thành các components riêng biệt. Việc thiết kế theo mô hình này giúp phân chia mã nguồn rõ ràng, dễ bảo trì, dễ mở rộng và tách biệt giữa giao diện người dùng và logic xử lý.

Mô hình phần mềm bao gồm các tầng:

- Presentation Layer (GUI): Tầng giao diện người dùng (WinForms).
- Business Logic Layer (BLL): Tầng xử lý nghiệp vụ.
- Data Access Layer (DAL): Tầng truy cập dữ liệu (Sử dụng công nghệ Entity Framework Core).
- Data Transfer Object (DTO/Entities): Tầng đối tượng vận chuyển dữ liệu giữa các tầng.

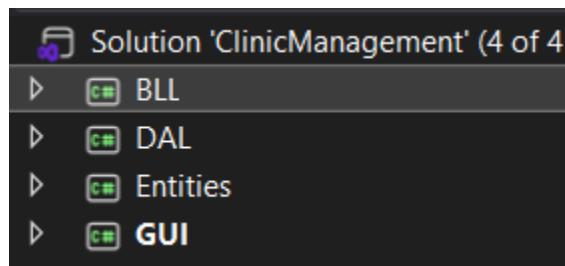


Hình 2. Mô hình kiến trúc 3 lớp

2. Tổ chức Dự án (Solution Structure)

Clinic Management Project trong Visual Studio được tổ chức thành 4 project thành phần riêng biệt trong cùng một solution nhằm đảm bảo tính đóng gói:

- GUI (ClinicManagement): Project chính để chạy chương trình, chứa các forms giao diện người dùng (frmLogin, frmMain,...).
- BLL (Business Logic): Chứa các file Services (UserService, PatientService,...) thực hiện các logic nghiệp vụ.
- DAL (Data Access): Chứa các file ‘ClinicDbContext’, ‘DbContextFactory’ và Migrations.
- Entities: Chứa các lớp thực thể (User, Patient, Medicine,...).



Hình 3. Cấu trúc Solution

3. Ví dụ minh họa: Mô tả quá trình thực hiện chức năng ‘Đăng nhập’

Quy trình luồng code đi từ tầng dữ liệu lên tầng giao diện:

- Bước 1: Xây dựng hàm xử lý tại tầng BLL (UserService) tại tầng Business Logic, lớp UserService sẽ khởi tạo DbContext và viết hàm kiểm tra thông tin đăng nhập từ cơ sở dữ liệu.

```
namespace BLL
{
    9 references
    public class UserService
    {
        private readonly ClinicDbContext _context;

        4 references
        public UserService()
        {
            _context = new ClinicDbContext();
        }

        1 reference
        public User Login(string username, string password)
        {
            return _context.Users.Include(u => u.Role).FirstOrDefault(u => u.UserName == username && u.Password == password);
        }
    }
}
```

- Bước 2: Gọi hàm xử lý tại tầng GUI (frmLogin). Tại Form Đăng nhập, ta khởi tạo đối tượng Service và gọi hàm Login khi người dùng bấm nút đăng nhập.

```

3 references
public partial class frmLogin : Form
{
    private UserService _service = new UserService();
    public frmLogin()
    {
        InitializeComponent();
    }

    //Button Login
    private void btnLogin_Click(object sender, EventArgs e)
    {
        if (string.IsNullOrEmpty(txtUsername.Text) || string.IsNullOrEmpty(txtPassword.Text))
        {
            MessageBox.Show("Vui lòng nhập tên đăng nhập và mật khẩu !");
            return;
        }

        User user = _service.Login(txtUsername.Text, txtPassword.Text);

        if (user != null)
        {
            UserSession.CurrentUser = user;
            MessageBox.Show($"Xin chào {user.FullName}!", "Đăng nhập thành công", MessageBoxButtons.OK, MessageBoxIcon.Information);

            MainForm f = new MainForm();
            f.Show();
            this.Hide();
        }
        else
        {
            MessageBox.Show("Tên tài khoản hoặc mật khẩu không đúng !", "Đăng nhập thất bại", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}

```

4. Kỹ thuật Transaction

Project có sử dụng kỹ thuật xử lý Giao dịch (Transaction) để đảm bảo tính toàn vẹn dữ liệu cho các nghiệp vụ phức tạp.

Áp dụng: Chức năng Đăng ký khám (RegisterExam) và Hoàn tất khám (FinishExamination).

- Chức năng Đăng ký khám (RegisterExam): Khi nhân viên tạo phiếu khám, hệ thống cần thực hiện các công việc:
 - o Tạo thêm dữ liệu mới trong bảng ‘MedicalRecord’ để thêm thông tin của bệnh nhân hay chỉ định bác sĩ khám.
 - o Thêm các dịch vụ, giá tiền dịch vụ vào bảng ‘RecordService’.

```

//Chức năng đăng ký khám
1 reference
public void RegisterExam(int patientId, int doctorId, string symptoms, int serviceId)
{
    using (var transaction = _context.Database.BeginTransaction())
    {
        try
        {
            var record = new MedicalRecord
            {
                PatientID = patientId,
                DoctorID = doctorId,
                Symptoms = symptoms,
                DateCreated = DateTime.Now,
                Status = "Waiting",
                TotalAmount = 0
            };
            _context.MedicalRecords.Add(record);
            _context.SaveChanges();

            if (serviceId > 0)
            {
                var service = _context.Services.Find(serviceId);
                if (service != null)
                {
                    var recordService = new RecordService
                    {
                        RecordID = record.RecordID,
                        ServiceID = serviceId,
                        Quantity = 1,
                        Price = service.Price
                    };
                    _context.RecordServices.Add(recordService);
                    _context.SaveChanges();
                }
            }
            transaction.Commit();
        }
        catch (Exception ex)
        {
            transaction.Rollback();
            throw new Exception("Đăng ký không thành công:" + ex.Message);
        }
    }
}

```

Hình 4. Xử lý Transaction cho hàm RegisterExam

- Chức năng Hoàn tất khám (FinishExamination): Khi bác sĩ đã hoàn tất việc thăm khám, chẩn đoán và kê đơn thuốc cho bệnh nhân, hệ thống cần thực hiện các công việc:
 - o Cập nhật trạng thái phiếu khám từ “Waiting” thành “Finished”.
 - o Lưu danh sách thuốc vào đơn thuốc (PrescriptionDetails).
 - o Thực hiện trừ số lượng tồn kho của các loại thuốc đã được kê đơn.

```

public void FinishExamination(int recordId, string diagnosis, List<PrescriptionDetail> listDonThuoc)
{
    using(var transaction = _context.Database.BeginTransaction())
    {
        try
        {
            var record = _context.MedicalRecords.FirstOrDefault(x => x.RecordID == recordId);
            if (record != null)
            {
                record.Diagnosis = diagnosis;
                record.Status = "Finished";
            }

            decimal totalDrugPrice = 0;
            foreach(var item in listDonThuoc)
            {
                item.RecordID = recordId;

                var medicine = _context.Medicines.Find(item.MedicineID);
                if(medicine != null)
                {
                    if(medicine.StockQuantity < item.Quantity)
                    {
                        throw new Exception($"Thuốc '{medicine.MedicineName}' không còn đủ tồn kho ! Số lượng còn lại: {medicine.StockQuantity}");
                    }
                    medicine.StockQuantity -= item.Quantity;
                    item.UnitPrice = medicine.SellPrice;
                }
                totalDrugPrice += (item.UnitPrice * item.Quantity);
                _context.PrescriptionDetails.Add(item);
            }
            record.TotalAmount = totalDrugPrice;
            _context.SaveChanges();
            transaction.Commit();
        }
        catch (Exception)
        {
            transaction.Rollback();
            throw;
        }
    }
}

```

Hình 5. Xử lý Transaction cho hàm FinishExamination

5. Truy vấn dữ liệu.

Hệ thống sử dụng công nghệ LINQ để xử lý các tác truy vấn và tương tác với dữ liệu (như tìm kiếm, sửa, xóa,...).

Ví dụ: Chức năng tìm kiếm bệnh nhân dựa trên từ khóa trên nhiều trường thông tin như Họ tên hoặc Số Điện Thoại.

```

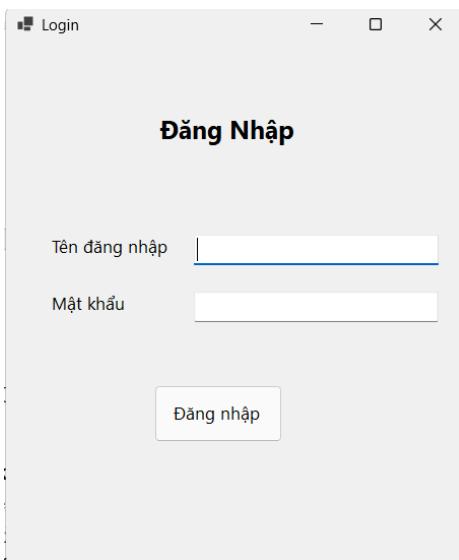
//Tìm kiếm bệnh nhân theo tên & SĐT
1 reference
public List<Patient> SearchPatient(string keyword)
{
    return _context.Patients
        .Where(p => p.FullName.Contains(keyword) || p.Phone.Contains(keyword))
        .ToList();
}

```

IV. CHƯƠNG TRÌNH PHẦN MỀM

1. Màn hình đăng nhập

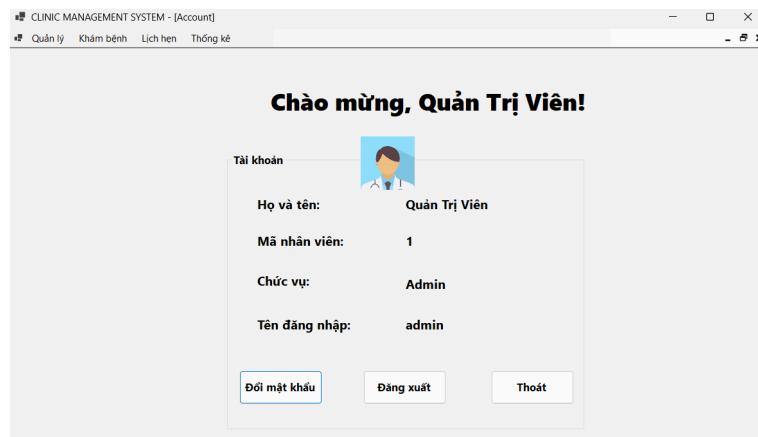
Giao diện khởi động sau khi chạy phần mềm, yêu cầu người dùng nhập các thông tin tài khoản (tên đăng nhập & mật khẩu). Sau khi đăng nhập, hệ thống sẽ tự động phân quyền (Admin, Bác sĩ, Nhân viên) dựa trên thông tin của tài khoản và hiển thị các chức năng tương ứng với từng vai trò.



Hình 6. Giao diện màn hình đăng nhập

2. Màn hình chính

Giao diện màn hình chính sau khi người dùng đăng nhập thành công. Menu các chức năng được bố trí ở phía bên trái (các menu strip).



Hình 7. Giao diện màn hình chính

3. Phân quyền Admin

3.1. Màn hình quản lý nhân viên

Giao diện cửa sổ dành cho quản trị viên giúp quản lý các thông tin tài khoản, nhân viên. Bao gồm các chức năng Thêm, Sửa, Phân vai trò, Cho thôi việc nhân viên,...

ID	User ID	Họ và tên	Tên đăng nhập	Chức vụ	Role
1	Quản Trị Viên	admin	ttt	1	Admin
3	Trần Thành Đạt			3	Nhân viên
4	Vũ Hoàng Vinh	vuvinh		2	Bác sĩ
5	Nguyễn Văn A	vana		2	Bác sĩ

Hình 8. Giao diện quản lý nhân viên

3.2. Màn hình quản lý thuốc

Giao diện cửa sổ tương tự với quản lý nhân viên, tại màn hình này quản trị viên có thể tương tác để quản lý thuốc trong kho được. Bao gồm các chức năng Thêm, Sửa, Xóa, Điều chỉnh thông tin chi tiết của các loại thuốc.

ID	Mã thuốc	Tên thuốc	Đơn vị	Giá nhập	Giá bán	Số lượng trong kho	Hạn sử dụng	Loại thuốc
2	Amoxicillin 500...	Vi	15.000	25.000	80	20/10/2025	Kháng sinh	
4	Berberin	Lo	10.000	18.000	193	01/01/2027	Kháng sinh	
1	Panadol Extra	Viên	1.000	2.500	500	31/12/2026	Giảm đau & H...	
3	Vitamin C 1000...	Hộp	45.000	60.000	50	15/05/2026	Vitamin & Kho...	

Hình 9. Giao diện quản lý thuốc

3.3. Màn hình quản lý các dịch vụ khám

Giao diện cửa sổ này cho phép quản trị viên quản lý các loại dịch vụ khám chưa bệnh và đơn giá của chúng trong phòng khám.

The screenshot shows a software interface titled "SERVICE MANGEMENT". At the top, there are four input fields: "Tim kiem" (Search), "Mã dịch vụ" (Service code), "Tên dịch vụ" (Service name), and "Giá tiền" (Price). Below these are five buttons: "Tim kiem" (Search), "Thêm" (Add), "Sửa" (Edit), "Xóa" (Delete), and "Reset". A table lists examination services with their names and prices:

Tên dịch vụ	Giá tiền (VND)
Khám thường	100.000
Khám VIP	300.000
Siêu âm	150.000

Hình 10. Giao diện quản lý dịch vụ

3.4. Màn hình thống kê lịch sử khám và doanh thu

Màn hình cho phép tra cứu toàn bộ hồ sơ bệnh án cũ của bệnh nhân, chi tiết các đơn thuốc đã kê và màn hình hiển thị bảng số liệu số ca khám cũng như doanh thu theo ngày trong khoảng thời gian lựa chọn xác định.

The screenshot shows a software interface titled "EXAMINATION HISTORY". At the top, there are three input fields: "Tim kiem" (Search), "Nhập tên hoặc SĐT" (Enter name or phone number), and "Tim kiem" (Search). Below these are two tables: "Danh sách khám bệnh" (List of examinations) and "Chi tiết đơn thuốc" (Prescription details).

Danh sách khám bệnh:

Ngày khám	Tên bệnh nhân	SDT	Bác sĩ chỉ định	Chẩn đoán	Tổng tiền	Trạng thái
27/12/2025 ...	Trần ...	11111...	Nguy...	tiêu c...	186.000	Paid
24/12/2025 ...	Trần ...	11111...	Vũ H...	Ngộ ...	600.000	Paid
20/12/2025 ...	Trần T...	01234...	Bác sĩ...	Viêm ...	0	Finish...
19/12/2025 ...	Lê Vă...	09998...	Bác sĩ...	Viêm ...	150.000	Paid

Chi tiết đơn thuốc:

Tên thuốc	SoLuong	DonVi	CachDung	DonGia	ThanhTier
Berberin	2	Lọ	s1t1	18000,00	36.000

Dịch vụ khám:

Tên dịch vụ	SoLuong	DonGia	ThanhTien
Siêu âm	1	150000,00	150.000

Hình 11. Giao diện màn hình lịch sử khám

THỐNG KÊ LƯỢT KHÁM THEO NGÀY			
Từ ngày:	16 Tháng Mười Hai 2025	Thống kê	
Đến ngày:	15 Tháng Giêng 2026		
Ngày	Số lượt khám	Doanh thu	
19/12/2025	1	150.000	
24/12/2025	1	600.000	
27/12/2025	1	186.000	

Tổng số lượt khám: 3
TỔNG DOANH THU: 936000,00

Hình 12. Giao diện màn hình thống kê

4. Phân quyền Nhân viên (Staffs)

4.1. Màn hình quản lý bệnh nhân & Đăng ký khám

Giao diện màn hình dành cho bộ phận tiếp đón bệnh nhân và đăng ký khám, bao gồm:

- Danh sách bệnh nhân: Hiển thị thông tin hành chính, hỗ trợ tìm kiếm theo tên hoặc SĐT.
- Đăng ký khám: Tạo phiếu khám và chỉ định dịch vụ khám kèm bác sĩ khám cho bệnh nhân.

PATIENT MANAGEMENT

Mã bệnh nhân	Họ và tên	Số điện thoại	Địa chỉ	Năm sinh	Giới tính
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Nam <input type="button" value="▼"/>
Triệu chứng <input type="text"/>					
<input type="button" value="Tim kiếm"/> <input type="button" value="Thêm"/> <input type="button" value="Sửa"/> <input type="button" value="Xóa"/> <input type="button" value="Reset"/>					
Chỉ định bác sĩ		Vũ Hoàng Vinh <input type="button" value="▼"/>	Chọn dịch vụ khám		Khám thường <input type="button" value="▼"/>
<input type="button" value="Đăng ký khám"/>					

Hình 13. Giao diện màn hình quản lý bệnh nhân

4.2. Màn hình thanh toán sau khi khám

Màn hình chứa chức năng hiển thị thông tin về các ca bệnh đã khám hoàn tất bởi bác sĩ, tính toán tổng chi phí thuốc, dịch vụ khám và cho phép thanh toán và ghi vào lịch sử thống kê/ lịch sử khám.

<p>Danh sách chờ thanh toán</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Họ và tên</th> <th>Số điện thoại</th> <th>Chẩn đoán</th> <th>Thời gian khám</th> </tr> </thead> <tbody> <tr> <td>Trần Thị B</td> <td>01234567...</td> <td>Viêm họng...</td> <td>20/12/202...</td> </tr> </tbody> </table>	Họ và tên	Số điện thoại	Chẩn đoán	Thời gian khám	Trần Thị B	01234567...	Viêm họng...	20/12/202...	<p>Tiền thuốc</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Tên thuốc</th> <th>Số lượng</th> <th>Đơn giá</th> <th>Thành tiền</th> </tr> </thead> <tbody> <tr> <td colspan="4" style="background-color: #cccccc;"> </td> </tr> </tbody> </table> <p>Tiền dịch vụ</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Tên dịch vụ</th> <th>Số lượng</th> <th>Đơn giá</th> <th>Thành tiền</th> </tr> </thead> <tbody> <tr> <td>Khám thường</td> <td>1</td> <td>100.000</td> <td>100.000</td> </tr> </tbody> </table> <p style="color: orange; font-weight: bold;">Tổng tiền cần thanh toán: 100.000</p>	Tên thuốc	Số lượng	Đơn giá	Thành tiền					Tên dịch vụ	Số lượng	Đơn giá	Thành tiền	Khám thường	1	100.000	100.000
Họ và tên	Số điện thoại	Chẩn đoán	Thời gian khám																						
Trần Thị B	01234567...	Viêm họng...	20/12/202...																						
Tên thuốc	Số lượng	Đơn giá	Thành tiền																						
Tên dịch vụ	Số lượng	Đơn giá	Thành tiền																						
Khám thường	1	100.000	100.000																						
<input type="button" value="Hoàn tất thanh toán"/>																									

Hình 14. Giao diện màn hình thanh toán

5. Phân quyền Bác sĩ – Màn hình khám bệnh

Giao diện cửa sổ hỗ trợ bác sĩ xem danh sách của các bệnh nhân đang chờ khám, thực hiện ghi nhận triệu chứng và chẩn đoán. Sau khi kết luận chẩn đoán, bác sĩ có thể thực hiện kê đơn thuốc: hệ thống tự động kiểm tra số lượng kho thuốc và tính tổng tiền.

DANH SÁCH CHỖ			
RecordID	Họ tên	SĐT	Lý do khám

DƠN THUỐC

Bệnh nhân đang khám:

Chẩn đoán	Chọn loại thuốc			
Panadol Extra	<input type="button" value="Thêm"/>	<input type="button" value="Xóa"/>		

Tên thuốc	Số lượng	Đơn vị	Đơn giá	Cách dùng
*				

Tổng chi phí:

Hình 15. Giao diện màn hình khám bệnh