Introduction

Welcome to the Clinic Management Application documentation. This application is a comprehensive multi-tier desktop application designed for managing clinic operations.

Purpose

The Clinic Management Application provides healthcare clinics with a complete solution for:

- **Patient Management**: Register and track patient information
- **Appointment Scheduling**: Book, modify, and manage appointments
- **Doctor Management**: Manage doctor profiles, specialties, and schedules
- **User Administration**: Role-based access control for staff

Target Users

- **Administrators**: Full system access for configuration and user management
- **Receptionists**: Patient registration and appointment scheduling
- **Doctors**: View schedules and patient information

Technology Stack

- .NET Framework 4.8
- C# 7.3
- Windows Forms
- Entity Framework 6.5
- SQL Server

Documentation Structure

This documentation is organized into several sections:

1. **Getting Started** - Installation and setup instructions
2. **Architecture** - Technical overview of the application layers
3. **User Guide** - How to use the application features
4. **API Reference** - Detailed class and method documentation

Getting Started

This guide will help you set up and run the Clinic Management Application.

Prerequisites

Before you begin, ensure you have the following installed:

- **Operating System**: Windows 10 or later
- **.NET Framework**: Version 4.8 or later
- **SQL Server**: LocalDB or full SQL Server instance
- **Visual Studio**: 2019 or later (for development)

Installation Steps

1. Clone the Repository

```
git clone https://github.com/shayandelbari/clinic-management.git
cd clinic-management
```

2. Database Setup

1. Open SQL Server Management Studio or use Visual Studio's SQL Server Object Explorer
2. Execute the database setup script located at:

   ```
   DAL/HealthCareClinicDB_T2_Script.sql
   ```

3. Verify the database `HealthCareClinic` is created with all tables

3. Configure Connection String

Update the connection string in `App.config` if necessary:

```
<connectionStrings>
  <add name="HealthCareClinicEntities"
       connectionString="..."
       providerName="System.Data.EntityClient" />
</connectionStrings>
```

4. Build the Solution

1. Open `ClinicManagement_proj.sln` in Visual Studio
2. Restore NuGet packages (right-click solution > Restore NuGet Packages)
3. Build the solution (Ctrl+Shift+B)

5. Run the Application

1. Set `ClinicManagement_proj` as the startup project
2. Press F5 or click Start to run
3. Log in with appropriate credentials

Default User Accounts

Role	Username	Description
Admin	admin	Full system access
Receptionist	receptionist	Patient and appointment management
Doctor	doctor	View schedules and patients

Next Steps

- Read the Architecture documentation to understand the system design
- Check the User Guide for feature documentation
- Browse the API Reference for detailed class documentation

Architecture

The Clinic Management Application follows a multi-tier architecture pattern, separating concerns into distinct layers for maintainability and scalability.

Overview

```
┌─────────────────────────────────────────┐
│         Presentation Layer (UI)         │
│  Windows Forms, Controllers, Dashboards │
├─────────────────────────────────────────┤
│       Business Logic Layer (BLL)  │
│      Services, DTOs, Utilities, App      │
├─────────────────────────────────────────┤
│     Data Access Layer (DAL)   │
│  Entity Framework, DbContext, Entities  │
├─────────────────────────────────────────┤
│      SQL Server Database      │
│     Tables, Views, Stored Procedures    │
└─────────────────────────────────────────┘
```

Presentation Layer (UI)

The UI layer handles all user interaction through Windows Forms.

Key Components

Component	Description
`Login.cs`	User authentication form
`AdminDashboard.cs`	Administrator interface
`DoctorDashboard.cs`	Doctor-specific interface
`ReceptionistDashboard.cs`	Receptionist interface

Controllers

Controllers manage panel navigation and user interactions:

- **ApptMgmtController**: Appointment CRUD operations
- **DoctorManagementController**: Doctor profile management
- **PatientRegistrationController**: Patient registration

- **SchedulingController**: Schedule management
- **UserManagementController**: User administration
- **ReportsController**: Report generation
- **NotificationController**: Notification handling

Utilities

- **NavigationManager**: Panel navigation and history
- **ImageHelper**: Image processing utilities

Business Logic Layer (BLL)

The BLL contains all business rules and data transformation logic.

Services

Service	Responsibility
`UserService`	User authentication and management
`PatientService`	Patient CRUD operations
`DoctorService`	Doctor management
`AppointmentService`	Appointment scheduling
`DoctorScheduleService`	Doctor availability
`NotificationService`	System notifications
`ViewsService`	Database views access

Data Transfer Objects (DTOs)

DTOs provide a clean data contract between layers:

- `PatientDTO` - Patient data with validation
- `DoctorDTO` - Doctor information
- `AppointmentDTO` - Appointment details
- `TimeSlotDTO` - Time slot representation
- `DoctorScheduleDTO` - Schedule information
- `UserDTO` - User account data
- `RoleDTO` - Role definitions
- `SpecialtyDTO` - Medical specialties
- `AuditAppointmentDTO` - Audit trail data

Application Core

- **ClinicManagementApp**: Main application coordinator

Utilities

- **Notification**: Notification data structure

Data Access Layer (DAL)

The DAL handles all database operations using Entity Framework 6.

Entity Framework Components

Component	Description
`ClinicDbContext`	Database context for EF operations
`HealthCareClinicModel`	Entity Data Model (EDMX)

Entities

Database entities mapped to tables:

- `Patient` - Patient records
- `Doctor` - Doctor profiles
- `Appointment` - Appointment records
- `TimeSlot` - Available time slots
- `DoctorSchedule` - Doctor schedules
- `User` - User accounts
- `Role` - User roles
- `Specialty` - Medical specialties
- `Audit_Appointment` - Audit records

Database Views

Optimized views for common queries:

- `vw_UpcomingAppointments` - Future appointments
- `vw_PatientRecordsSummary` - Patient summary
- `vw_PatientClinicalSummary` - Clinical data
- `vw_DoctorTodaySchedule` - Today's schedule

Stored Procedures

- `sp_GetAvailableSlots` - Available appointment slots

Design Patterns

Patterns Used

1. **Repository Pattern**: Data access abstraction through services
2. **DTO Pattern**: Data transfer between layers
3. **Controller Pattern**: UI action handling
4. **Singleton**: Application state management

Dependency Flow

```
UI Controllers → BLL Services → DAL DbContext → Database
      ↓↓
    DTOs        Entities
```

Security

- Role-based access control (Admin, Doctor, Receptionist)
- Password hashing for user authentication
- Input validation in DTOs
- Parameterized queries via Entity Framework

Database Schema

The database uses the following key relationships:

- Patients have many Appointments
- Doctors have many Appointments
- Doctors have many DoctorSchedules
- Doctors have many Specialties
- Appointments reference TimeSlots
- Users are assigned Roles

User Guide

This guide covers the main features and workflows of the Clinic Management Application.

Login

1. Launch the application
2. Enter your username and password
3. Click **Login**
4. You will be directed to your role-specific dashboard

Role-Specific Features

Administrator Dashboard

Administrators have full access to all system features:

User Management

- **Add User**: Create new user accounts
- **Edit User**: Modify existing user details
- **Delete User**: Remove user accounts
- **Assign Roles**: Set user permissions

Doctor Management

- **Add Doctor**: Register new doctors
- **Edit Doctor**: Update doctor information
- **Manage Specialties**: Assign medical specialties
- **View Workload**: Monitor doctor schedules

System Reports

- **Appointment Reports**: View appointment statistics
- **Patient Reports**: Patient data summaries
- **Doctor Reports**: Doctor activity reports

Receptionist Dashboard

Receptionists manage day-to-day clinic operations:

Patient Registration

1. Click **New Patient**
2. Fill in patient details:
    - First Name (required, max 64 characters)

    - Last Name (required, max 64 characters)
    - Date of Birth
    - Phone Number (required, max 20 characters)
    - Insurance Number (required, max 24 characters)
3. Click **Save**

Appointment Scheduling

1. Select a patient from the list
2. Click **New Appointment**
3. Choose:
    - Doctor
    - Date
    - Available Time Slot
4. Add optional notes
5. Click **Schedule**

Appointment Management

- **View Appointments**: See all scheduled appointments
- **Edit Appointment**: Modify appointment details
- **Cancel Appointment**: Remove an appointment
- **Check-in Patient**: Mark patient as arrived

Doctor Dashboard

Doctors view their schedules and patient information:

Today's Schedule

- View all appointments for the current day
- See patient details for each appointment
- View appointment notes

Patient Information

- Access patient clinical summaries
- View appointment history
- Review patient records

Common Tasks

Searching for Patients

1. Use the search box in the patient list
2. Enter patient name, phone, or insurance number

3. Results filter automatically

Scheduling an Appointment

1. Navigate to **Appointments** section
2. Click **New Appointment**
3. Select patient (or register new patient first)
4. Choose doctor and specialty
5. Pick available date and time slot
6. Add any relevant notes
7. Confirm the appointment

Modifying an Appointment

1. Find the appointment in the list
2. Double-click or select **Edit**
3. Update the necessary fields
4. Save changes

Canceling an Appointment

1. Select the appointment
2. Click **Cancel**
3. Confirm the cancellation

Viewing Reports

1. Navigate to **Reports** section
2. Select report type:
    - Daily Appointments
    - Weekly Summary
    - Patient Statistics
    - Doctor Workload
3. Choose date range if applicable
4. Click **Generate**

Data Validation

The application validates all input data:

Field	Validation Rules
First Name	Required, max 64 characters

Field	Validation Rules
Last Name	Required, max 64 characters
Phone Number	Required, max 20 characters
Insurance Number	Required, max 24 characters
Appointment Notes	Max 512 characters
License Number	Required, max 24 characters

Notifications

The notification system alerts users to:

- Upcoming appointments
- Schedule changes
- System updates
- Important reminders

Access notifications via the bell icon in the dashboard header.

Keyboard Shortcuts

Shortcut	Action
F5	Refresh current view
Ctrl+N	New record
Ctrl+S	Save changes
Escape	Cancel/Close dialog

Troubleshooting

Cannot Login

- Verify username and password
- Check if account is active
- Contact administrator if locked out

Appointment Not Saving

- Ensure all required fields are filled
- Verify selected time slot is available
- Check patient and doctor are valid

Data Not Loading

- Check database connection
- Verify network connectivity
- Restart the application

Support

For technical support or bug reports, contact your system administrator or refer to the API Reference for developer documentation.