

```

using System.Collections.Generic;
using System.Threading.Tasks;
using HMS_API.Models; // Your API models

namespace MyApp.WinFormsUI.Services
{
    public interface IApiService
    {
        Task<List<Patient>> GetPatientsAsync();
        Task<Patient> CreatePatientAsync(Patient patient);
        Task<Patient> UpdatePatientAsync(int id, Patient patient);
        Task DeletePatientAsync(int id);

        // Similar CRUD for other entities, e.g., Doctor, Appointment, etc.
        // For brevity, only Patient is shown.
    }
}

```

```

using System;
using System.Collections.Generic;
using System.Net.Http;
using System.Net.Http.Json;
using System.Threading.Tasks;
using HMS_API.Models; // Your API models

namespace MyApp.WinFormsUI.Services
{
    public class ApiService : IApiService
    {
        private readonly HttpClient _httpClient;
        private readonly string _baseUrl;

        public ApiService(HttpClient httpClient)
        {
            _httpClient = httpClient;
            // Set your API base URL here.
            _baseUrl = "https://localhost:5001/api";
        }

        public async Task<List<Patient>> GetPatientsAsync()
        {
            var url = $"{_baseUrl}/patient";

```

```

        return await _httpClient.GetFromJsonAsync<List<Patient>>(url) ?? new
List<Patient>();
    }

    public async Task<Patient> CreatePatientAsync(Patient patient)
    {
        var url = $" {_baseUrl}/patient";
        var response = await _httpClient.PostAsJsonAsync(url, patient);
        response.EnsureSuccessStatusCode();
        return await response.Content.ReadFromJsonAsync<Patient>(!);
    }

    public async Task<Patient> UpdatePatientAsync(int id, Patient patient)
    {
        var url = $" {_baseUrl}/patient/{id}";
        var response = await _httpClient.PutAsJsonAsync(url, patient);
        response.EnsureSuccessStatusCode();
        return await response.Content.ReadFromJsonAsync<Patient>(!);
    }

    public async Task DeletePatientAsync(int id)
    {
        var url = $" {_baseUrl}/patient/{id}";
        var response = await _httpClient.DeleteAsync(url);
        response.EnsureSuccessStatusCode();
    }

    // Implement other CRUD methods similarly.
}

```

```

using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Hosting;
using System;
using System.Net.Http;
using System.Windows.Forms;
using MyApp.BusinessLogic; // BLL project namespace
using MyApp.BusinessLogic.Services;
using MyApp.WinFormsUI.Services;
using HMS_API.DB;

```

```

using Microsoft.EntityFrameworkCore;

namespace MyApp.WinFormsUI
{
    static class Program
    {
        public static IServiceProvider ServiceProvider { get; private set; } = default!;
        [STAThread]
        static void Main()
        {
            Application.SetHighDpiMode(HighDpiMode.SystemAware);
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);

            // Build the Host to set up dependency injection.
            var host = Host.CreateDefaultBuilder()
                .ConfigureServices((context, services) =>
                {
                    // Register HttpClient for API service.
                    services.AddHttpClient<IApiService, ApiService>();

                    // Register API's DbContext (if the BLL uses it for additional processing).
                    services.AddDbContext<AppDbContext>(options =>
                        options.UseSqlServer("YourConnectionStringHere"));

                    // Register the Business Logic Manager.
                    services.AddScoped<HospitalBusinessManager>();

                    // Register the Notification Service.
                    services.AddScoped<MyApp.BusinessLogic.Services.NotificationService>();

                    // Register the main WinForms form.
                    services.AddTransient<MainForm>();
                })
                .Build();

            ServiceProvider = host.Services;
            Application.Run(ServiceProvider.GetRequiredService<MainForm>());
        }
    }
}

```

```

using System;

```

```

using System.Collections.Generic;
using System.Threading.Tasks;
using System.Windows.Forms;
using MyApp.WinFormsUI.Services;
using MyApp.BusinessLogic; // BLL namespace
using MyApp.BusinessLogic.Services;

namespace MyApp.WinFormsUI
{
    public partial class MainForm : Form
    {
        private readonly IApiService _apiService;
        private readonly HospitalBusinessManager _businessManager;
        private readonly NotificationService _notificationService;

        public MainForm(IApiService apiService, HospitalBusinessManager businessManager,
            NotificationService notificationService)
        {
            _apiService = apiService;
            _businessManager = businessManager;
            _notificationService = notificationService;
            InitializeComponent();

            // Subscribe to BLL events.
            _businessManager.AppointmentProcessed += OnAppointmentProcessed;
            _notificationService.NotificationSent += OnNotificationSent;
        }

        // API CRUD Operation: Load Patients.
        private async void btnLoadPatients_Click(object sender, EventArgs e)
        {
            try
            {
                List<HMS_API.Models.Patient> patients = await _apiService.GetPatientsAsync();
                lstPatients.DataSource = patients;
                lstPatients.DisplayMember = "AppUser.FullName";
            }
            catch (Exception ex)
            {
                MessageBox.Show($"Error loading patients: {ex.Message}");
            }
        }

        // API CRUD Operation: Create a new patient.
    }
}

```

```

private async void btnCreatePatient_Click(object sender, EventArgs e)
{
    try
    {
        // In a real app, you'd get these values from input controls.
        var newPatient = new HMS_API.Models.Patient
        {
            PatientDOB = new DateOnly(1990, 1, 1),
            PatientBloodGroup = "A+",
            PatientAddress = "123 Main St",
            AppUserId = "some-user-id" // Typically set after registering a user.
        };
        HMS_API.Models.Patient createdPatient = await
_apiService.CreatePatientAsync(newPatient);
        MessageBox.Show($"Patient created: {createdPatient.AppUser?.FullName}");
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Error creating patient: {ex.Message}");
    }
}

// BLL Operation: Process pending appointments.
private async void btnProcessAppointments_Click(object sender, EventArgs e)
{
    try
    {
        await _businessManager.ProcessPendingAppointmentsAsync();
        MessageBox.Show("Pending appointments processed successfully.");
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Error processing appointments: {ex.Message}");
    }
}

// BLL Operation: Generate daily report.
private async void btnGenerateReport_Click(object sender, EventArgs e)
{
    try
    {
        string report = await
_businessManager.GenerateDailyReportAsync(DateTime.UtcNow);
        txtReport.Text = report;
    }
}

```

```

    }
    catch (Exception ex)
    {
        MessageBox.Show($"Error generating report: {ex.Message}");
    }
}

// BLL Operation: Send a notification.
private async void btnSendNotification_Click(object sender, EventArgs e)
{
    try
    {
        await _notificationService.SendNotificationAsync("Test notification from WinForms
UI.");
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Error sending notification: {ex.Message}");
    }
}

// Event handler for processed appointments.
private void OnAppointmentProcessed(object sender,
AppointmentProcessedEventArgs e)
{
    this.Invoke((MethodInvoker)(() =>
    {
        MessageBox.Show($"Appointment {e.AppointmentId} processed. Status:
{e.Status}", "Appointment Processed");
    }));
}

// Event handler for notifications.
private void OnNotificationSent(object sender, string message)
{
    this.Invoke((MethodInvoker)(() =>
    {
        MessageBox.Show($"Notification: {message}", "Notification");
    }));
}
}
}

```

```

namespace MyApp.WinFormsUI
{
    partial class MainForm
    {
        private System.ComponentModel.IContainer components = null;
        private System.Windows.Forms.Button btnLoadPatients;
        private System.Windows.Forms.Button btnCreatePatient;
        private System.Windows.Forms.Button btnProcessAppointments;
        private System.Windows.Forms.Button btnGenerateReport;
        private System.Windows.Forms.Button btnSendNotification;
        private System.Windows.Forms.ListBox lstPatients;
        private System.Windows.Forms.TextBox txtReport;

        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
                base.Dispose(disposing);
            }

            #region Windows Form Designer generated code

            private void InitializeComponent()
            {
                this.btnLoadPatients = new System.Windows.Forms.Button();
                this.btnCreatePatient = new System.Windows.Forms.Button();
                this.btnProcessAppointments = new System.Windows.Forms.Button();
                this.btnGenerateReport = new System.Windows.Forms.Button();
                this.btnSendNotification = new System.Windows.Forms.Button();
                this.lstPatients = new System.Windows.Forms.ListBox();
                this.txtReport = new System.Windows.Forms.TextBox();
                this.SuspendLayout();
                //
                // btnLoadPatients
                //
                this.btnLoadPatients.Location = new System.Drawing.Point(12, 12);
                this.btnLoadPatients.Name = "btnLoadPatients";
                this.btnLoadPatients.Size = new System.Drawing.Size(150, 30);
                this.btnLoadPatients.TabIndex = 0;
                this.btnLoadPatients.Text = "Load Patients";
                this.btnLoadPatients.UseVisualStyleBackColor = true;
                this.btnLoadPatients.Click += new
System.EventHandler(this.btnLoadPatients_Click);
                //

```

```

// btnCreatePatient
//
this.btnCreatePatient.Location = new System.Drawing.Point(180, 12);
this.btnCreatePatient.Name = "btnCreatePatient";
this.btnCreatePatient.Size = new System.Drawing.Size(150, 30);
this.btnCreatePatient.TabIndex = 1;
this.btnCreatePatient.Text = "Create Patient";
this.btnCreatePatient.UseVisualStyleBackColor = true;
this.btnCreatePatient.Click += new
System.EventHandler(this.btnCreatePatient_Click);
//
// btnProcessAppointments
//
this.btnProcessAppointments.Location = new System.Drawing.Point(12, 60);
this.btnProcessAppointments.Name = "btnProcessAppointments";
this.btnProcessAppointments.Size = new System.Drawing.Size(150, 30);
this.btnProcessAppointments.TabIndex = 2;
this.btnProcessAppointments.Text = "Process Appointments";
this.btnProcessAppointments.UseVisualStyleBackColor = true;
this.btnProcessAppointments.Click += new
System.EventHandler(this.btnProcessAppointments_Click);
//
// btnGenerateReport
//
this.btnGenerateReport.Location = new System.Drawing.Point(180, 60);
this.btnGenerateReport.Name = "btnGenerateReport";
this.btnGenerateReport.Size = new System.Drawing.Size(150, 30);
this.btnGenerateReport.TabIndex = 3;
this.btnGenerateReport.Text = "Generate Report";
this.btnGenerateReport.UseVisualStyleBackColor = true;
this.btnGenerateReport.Click += new
System.EventHandler(this.btnGenerateReport_Click);
//
// btnSendNotification
//
this.btnSendNotification.Location = new System.Drawing.Point(12, 108);
this.btnSendNotification.Name = "btnSendNotification";
this.btnSendNotification.Size = new System.Drawing.Size(150, 30);
this.btnSendNotification.TabIndex = 4;
this.btnSendNotification.Text = "Send Notification";
this.btnSendNotification.UseVisualStyleBackColor = true;
this.btnSendNotification.Click += new
System.EventHandler(this.btnSendNotification_Click);
//

```



```

// lstPatients
//
this.lstPatients.FormattingEnabled = true;
this.lstPatients.ItemHeight = 15;
this.lstPatients.Location = new System.Drawing.Point(12, 160);
this.lstPatients.Name = "lstPatients";
this.lstPatients.Size = new System.Drawing.Size(300, 150);
this.lstPatients.TabIndex = 5;
//
// txtReport
//
this.txtReport.Location = new System.Drawing.Point(330, 12);
this.txtReport.Multiline = true;
this.txtReport.Name = "txtReport";
this.txtReport.ScrollBars = System.Windows.Forms.ScrollBars.Vertical;
this.txtReport.Size = new System.Drawing.Size(400, 300);
this.txtReport.TabIndex = 6;
//
// MainForm
//
this.ClientSize = new System.Drawing.Size(750, 375);
this.Controls.Add(this.txtReport);
this.Controls.Add(this.lstPatients);
this.Controls.Add(this.btnSendNotification);
this.Controls.Add(this.btnGenerateReport);
this.Controls.Add(this.btnProcessAppointments);
this.Controls.Add(this.btnCreatePatient);
this.Controls.Add(this.btnLoadPatients);
this.Name = "MainForm";
this.Text = "Hospital Management - UI";
this.ResumeLayout(false);
this.PerformLayout();
}

#endregion
}
}

```