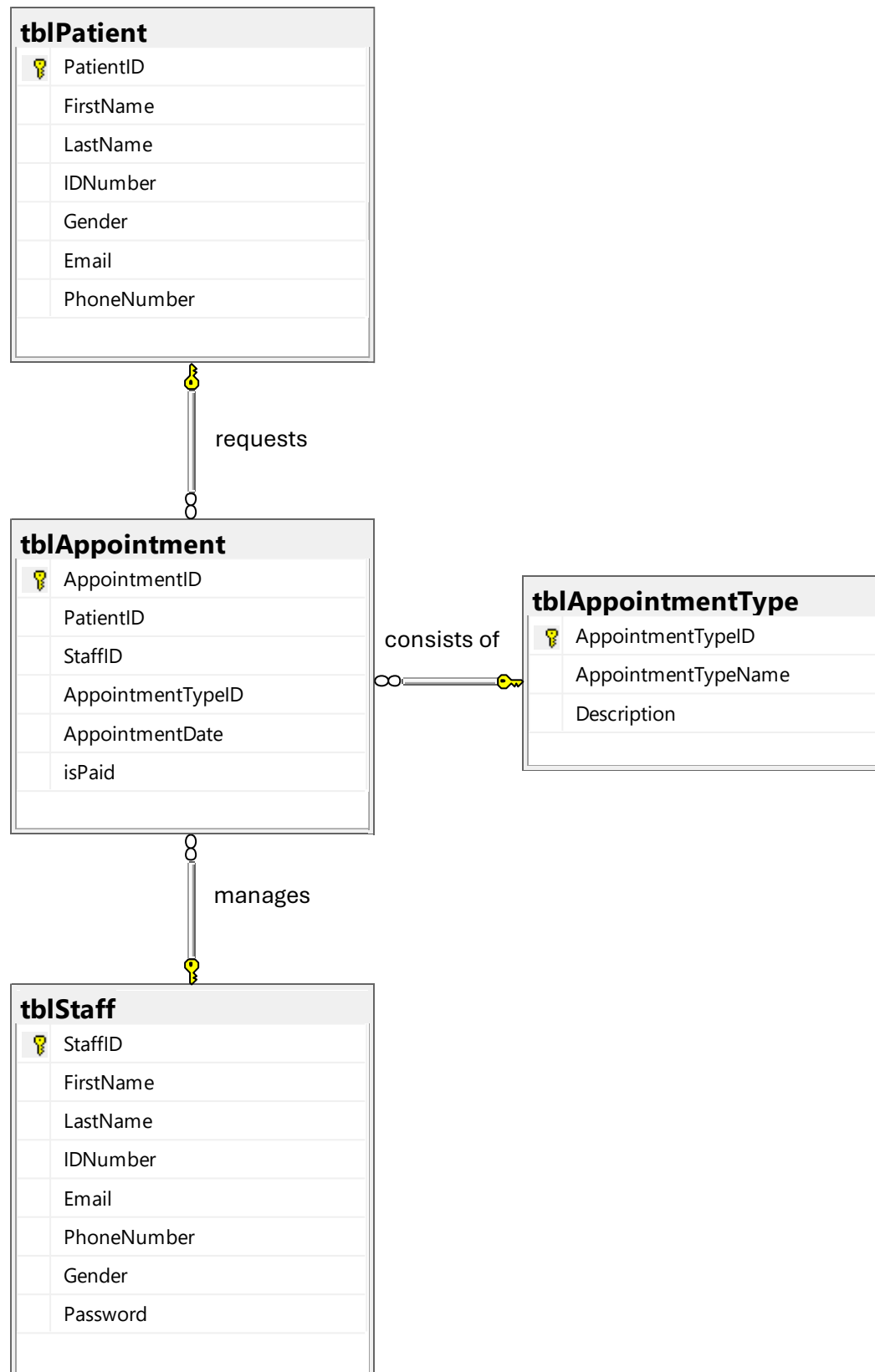





Physical Data Model

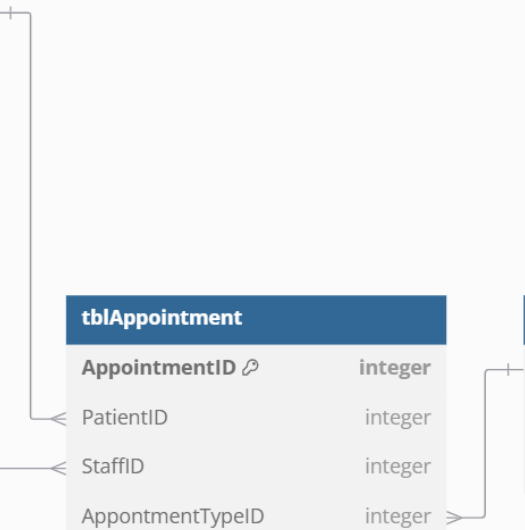


tblPatient	
PatientID 	integer
FirstName	nvarchar(50)
LastName	nvarchar(50)
IDNumber	nvarchar(13)
Email	nvarchar(100)
PhoneNumber	nvarchar(10)
Gender	nvarchar(6)

tblAppointment	
AppointmentID 	integer
PatientID	integer
StaffID	integer
AppontmentTypeID	integer
AppointmentDate	datetime
isPaid	bit

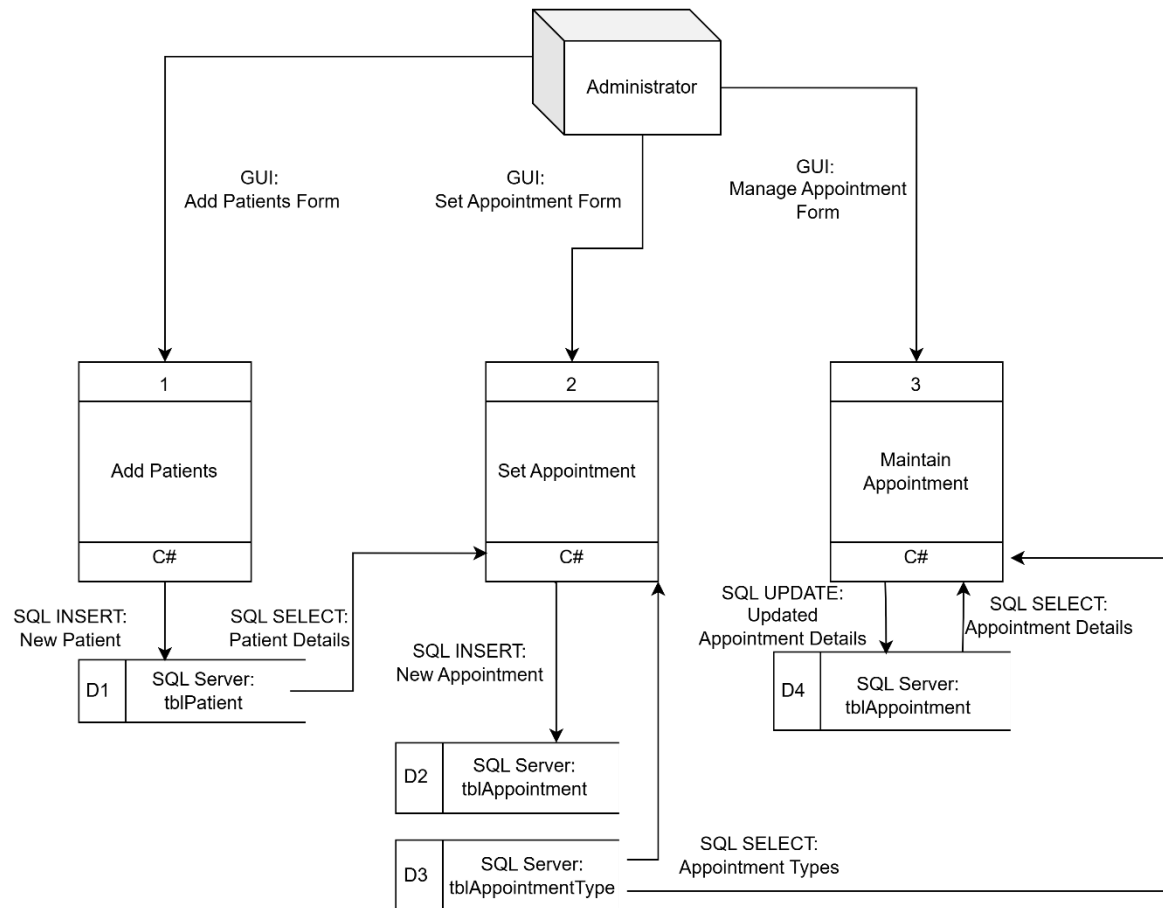
tblAppointmentType	
AppointmentTypeID 	integer
AppointmentTypeName	nvarchar(50)
Description	nvarchar(255)

tblStaff	
StaffID 	integer
FirstName	nvarchar(50)
LastName	nvarchar(50)
IDNumber	nvarchar(13)
Email	nvarchar(100)
PhoneNumber	nvarchar(10)
Gender	nvarchar(6)
Password	nvarchar(55)



Physical Process Model

Appointment Scheduling and Maintaining Process



SQL USED:

Creating Database

```
CREATE DATABASE VitalityCareSoftware;
```

```
USE VitalityCareSoftware;
```

```
CREATE TABLE tblStaff (  
    StaffID INT PRIMARY KEY IDENTITY(1,1),  
    FirstName NVARCHAR(50),  
    LastName NVARCHAR(50),  
    IDNumber NVARCHAR(13),  
    Email NVARCHAR(100),  
    PhoneNumber NVARCHAR(10),  
    Gender NVARCHAR(6),  
    Password NVARCHAR(55)  
);
```

```
CREATE TABLE tblPatient (  
    PatientID INT PRIMARY KEY IDENTITY(1,1),  
    FirstName NVARCHAR(50),  
    LastName NVARCHAR(50),  
    IDNumber NVARCHAR(13),  
    Gender NVARCHAR(6),  
    Email NVARCHAR(55),  
    PhoneNumber NVARCHAR(10)  
);
```

```
CREATE TABLE tblAppointmentType (  
    AppointmentTypeID INT PRIMARY KEY IDENTITY(1,1),  
    AppointmentTypeName NVARCHAR(50),  
    Description NVARCHAR(255)  
);
```

```
CREATE TABLE tblAppointment (  
    AppointmentID INT PRIMARY KEY IDENTITY(1,1),  
    PatientID INT FOREIGN KEY REFERENCES tblPatient(PatientID),  
    StaffID INT FOREIGN KEY REFERENCES tblStaff(StaffID),  
    AppointmentTypeID INT FOREIGN KEY REFERENCES  
tblAppointmentType(AppointmentTypeID),  
    AppointmentDate DATETIME,  
    isPaid BIT  
);
```

Maintaining Tables

Maintaining Appointments

- SELECT * FROM tblAppointment ORDER BY AppointmentDate DESC
- UPDATE tblAppointment SET PatientID = @PatientID, StaffID = @StaffID, AppointmentDate = @AppointmentDate, AppointmentTypeID = @AppointmentTypeID, isPaid = @IsPaid WHERE AppointmentID = @AppointmentID
- DELETE FROM tblAppointment WHERE AppointmentID = @AppointmentID
- INSERT INTO tblAppointment (AppointmentDate, StaffID, PatientID, AppointmentTypeID, isPaid) VALUES (@AppDate, @staffID, @patientID, @AppTypeID, @isPaid)

Maintaining Appointment Type

- SELECT * FROM tblAppointmentType
- UPDATE tblAppointmentType SET AppointmentTypeName = @AppointmentTypeName, Description = @Description WHERE AppointmentTypeID = @AppointmentTypeID
- DELETE FROM tblAppointmentType WHERE AppointmentTypeID = @AppointmentTypeID
- INSERT INTO tblAppointmentType (AppointmentTypeName, Description) VALUES (@AppointmentTypeName, @Description)

Maintaining Patients

- SELECT * FROM tblPatient
- UPDATE tblPatient SET FirstName = @FirstName, LastName = @LastName, IDNumber = @IDNumber, PhoneNumber = @PhoneNumber, Email = @Email, Gender = @Gender WHERE PatientID = @PatientID
- DELETE FROM tblPatient WHERE PatientID = @PatientID
- INSERT INTO tblPatient (FirstName, LastName, IDNumber, Gender, PhoneNumber, Email) VALUES (@FirstName, @LastName, @IDNumber, @Gender, @PhoneNumber, @Email)

Maintaining Staff

- SELECT * FROM tblStaff
- UPDATE tblStaff SET FirstName = @FirstName, LastName = @LastName, IDNumber = @IDNumber, PhoneNumber = @PhoneNumber, Email = @Email, Gender = @Gender, Password = @Password WHERE StaffID = @StaffID
- DELETE FROM tblStaff WHERE StaffID = @StaffID
- INSERT INTO tblStaff (FirstName, LastName, IDNumber, Gender, PhoneNumber, Email, Password) VALUES (@FirstName, @LastName, @IDNumber, @Gender, @PhoneNumber, @Email, @Password)

Querying

Staff Login Form

- SELECT COUNT(*) FROM tblStaff WHERE FirstName + ' ' + LastName LIKE @Username AND Password = @Password
- SELECT StaffID FROM tblStaff WHERE FirstName + ' ' + LastName = @Username AND Password = @Password

Set Appointment Form

- SELECT patientID, FirstName + ' ' + LastName AS 'Patient Name' FROM tblPatient
- SELECT DISTINCT AppointmentTypeName FROM tblAppointmentType
- SELECT * FROM tblPatient WHERE patientID = @PatientID

Manage Appointment Type

- SELECT AppointmentTypeID, AppointmentTypeName FROM tblAppointmentType

Reporting

- SELECT COUNT(*) FROM tblPatient
- SELECT COUNT(*) FROM tblStaff
- SELECT COUNT(*) FROM tblAppointment
- SELECT COUNT(*) FROM tblAppointment WHERE AppointmentDate BETWEEN @startDate AND @endDate
- SELECT TOP 10 p.FirstName + ' ' + p.LastName AS PatientName, COUNT(a.AppointmentID) AS AppointmentCount FROM tblAppointment a INNER JOIN tblPatient p ON a.PatientID = p.PatientID GROUP BY p.FirstName, p.LastName ORDER BY AppointmentCount DESC
- SELECT COUNT(*) FROM tblAppointment WHERE isPaid = 1
- SELECT COUNT(*) FROM tblAppointment WHERE isPaid = 0
- SELECT COUNT(*) FROM tblAppointment WHERE isPaid = 1 AND AppointmentDate BETWEEN @startDate AND @endDate
- SELECT COUNT(*) FROM tblAppointment WHERE isPaid = 0 AND AppointmentDate BETWEEN @startDate AND @endDate
- SELECT at.AppointmentTypeName, COUNT(a.AppointmentID) AS AppointmentCount FROM tblAppointmentType at LEFT JOIN tblAppointment a ON at.AppointmentTypeID = a.AppointmentTypeID GROUP BY at.AppointmentTypeName
- SELECT at.AppointmentTypeName, COUNT(a.AppointmentID) AS AppointmentCount FROM tblAppointmentType at LEFT JOIN tblAppointment a ON at.AppointmentTypeID = a.AppointmentTypeID WHERE a.AppointmentDate BETWEEN @startDate AND @endDate GROUP BY at.AppointmentTypeName
- SELECT Gender, COUNT(*) AS Count FROM tblPatient GROUP BY Gender
- SELECT Gender, COUNT(*) AS Count FROM tblStaff GROUP BY Gender
- SELECT s.FirstName + ' ' + s.LastName AS StaffName, COUNT(a.AppointmentID) AS AppointmentCount FROM tblAppointment a INNER JOIN tblStaff s ON a.StaffID = s.StaffID GROUP BY s.FirstName, s.LastName ORDER BY AppointmentCount DESC

Maintaining a child entity of the data model (Appointment Type)

```
35 1 reference
36 private bool ValidateInputs()
37 {
38     if (string.IsNullOrEmpty(cmbAppointmentTypeName.Text))
39     {
40         MessageBox.Show("Appointment Type cannot be empty.", "Validation Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
41         return false;
42     }
43
44     return true; // Return true if all inputs are valid
45 }
46
47 1 reference
48 private void btnUpdate_Click(object sender, EventArgs e)
49 {
50     //Enabling Input components for editing
51     cmbAppointmentTypeName.Enabled = true;
52     txtDescription.Enabled = true;
53
54     MessageBox.Show("Update enabled. You may edit the appointment type record. Please don't forget to save changes.", "Editing enabled");
55 }
56
57 1 reference
58 private void btnSave_Click(object sender, EventArgs e)
59 {
60     if (ValidateInputs())
61     {
62         if (MessageBox.Show("Do you want to save the changes?", "Confirm Save", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
63         {
64             string query = "UPDATE tblAppointmentType SET AppointmentTypeName = @AppointmentTypeName, Description = @Description WHERE AppointmentTypeID = @AppointmentTypeID";
65
66             using (SqlConnection conn = new SqlConnection(connectionString))
67             {
68                 using (SqlCommand cmd = new SqlCommand(query, conn))
69                 {
70                     cmd.Parameters.AddWithValue("@AppointmentTypeID", txtAppointmentID.Text);
71                     cmd.Parameters.AddWithValue("@AppointmentTypeName", cmbAppointmentTypeName.Text);
72                     cmd.Parameters.AddWithValue("@Description", txtDescription.Text);
73
74                     try
75                     {
76                         conn.Open();
77                         int rowsAffected = cmd.ExecuteNonQuery();
78
79                         if (rowsAffected > 0)
80                         {
81                             MessageBox.Show("Appointment record updated successfully.", "Success", MessageBoxButtons.OK, MessageBoxIcon.Information);
82
83                             //Disabling Input components for editing
84                             cmbAppointmentTypeName.Enabled = false;
85                             txtDescription.Enabled = false;
86
87                             //Reloading Database
88                             string qry = "SELECT * FROM tblAppointmentType";
```

Report Examples

☐ Filter from Date 2024/08/26 to 2024/09/02

Overall Summary Report
Total Patients: 2
Total Staff: 1
Total Appointments: 2
Top 10 Most Frequent Patients:
PatientName: Elon Musk
AppointmentCount: 1
PatientName: Tervin Gates
AppointmentCount: 1
Paid Appointments: 1
Unpaid Appointments: 1
Appointment Types Summary:
AppointmentTypeName: Counseling
AppointmentCount: 0
AppointmentTypeName: Diagnosis
AppointmentCount: 0
AppointmentTypeName: Emergency
AppointmentCount: 0

☒ Filter from Date 2024/08/26 to 2024/09/02

Summary Report from 2024/08/26 to 2024/09/02
Total Patients: 2
Total Staff: 1
Total Appointments: 0
Top 10 Most Frequent Patients:
PatientName: Elon Musk
AppointmentCount: 1
PatientName: Tervin Gates
AppointmentCount: 1
Paid Appointments: 0
Unpaid Appointments: 0
Appointment Types Summary:
Gender Breakdown for Patients:
Gender: Male
Count: 2
Gender Breakdown for Staff:
Gender: Male

User Manual

Welcome to the Vitality Care Software! This software is designed to streamline the management of a healthcare facility, allowing users to efficiently handle patient information, appointments, staff details, and generate necessary reports.

Installing the software

1. Obtain the installation package for the Vitality Care Software from your system administrator or download it from the provided github link.
2. Extract the contents of the zip to a folder on your computer.
3. Double-click the Vitality Care Software logo to start the application. Upon first launch, you may need to configure the database connection settings.

Technical Requirements

- Operating System: Windows 10 or later
- Processor: Intel Core i5 or Ryzen 5 with minimum 2.0 GHz clock speed
- Memory: Minimum: 4 GB
Recommended: 8 GB
- Hard Disk Space: Minimum of 500 MB free space
- Database: Microsoft SQL Server 2017 or later
- Other requirements: .NET Framework 4.7.2 or later

Using the Software

Logging In

1. **Launch the Software:**
2. **Enter Credentials:**
 - On the login screen, enter your assigned username and password.
 - Click the "Login" button.
3. **First-Time Users:**
 - If you are a first-time user, contact your system administrator to receive your login credentials.

Managing Patients

1. **Adding a New Patient:**
 - Click the "Add Patient" button.
 - Enter the patient's details in the provided fields.
 - Click "Add Patient" to add the patient to the system.
2. **Editing Patient Details (Manage Patients Form):**
 - Select a patient from the list and click "Update"
 - Update the necessary fields and click "Save" to apply changes.
3. **Deleting a Patient (Only Admin can delete records):**
 - Select a patient from the list and click "Delete."
 - Confirm the deletion in the prompt that appears.

Managing Appointments

1. Scheduling an Appointment:

- Click the "Set Appointment" button.
- Select a patient from the drop-down list.
- Choose the appointment type and enter the date and time.
- Click "Set Appointment" to schedule the appointment.

2. Updating an Appointment (Manage Appointment Form):

- Select an appointment from the list and click "Update."
- Modify the details as needed and click "Save."

3. Deleting an Appointment (Only Admin can delete records):

- Select an appointment from the list and click "Delete."
- Confirm the deletion in the prompt that appears.

Managing Staff

1. Adding New Staff:

- Click the "Add Staff" button.
- Enter the staff member's details.
- Click "Add Staff" to add the staff member to the system.

2. Editing Staff Details:

- Select a staff member from the list and click "Update."
- Update the necessary fields and click "Save."

3. Deleting Staff:

- Select a staff member from the list and click "Delete."
- Confirm the deletion in the prompt that appears.

Managing Appointment Type

1. Adding New Appointment Type:

- Click the "Add Appointment Type" button.
- Enter the new Appointment Type details.
- Click "Add Appointment Type" to add the appointment type to the system.

2. Editing Appointment Type:

- Select appointment type from the list and click "Update."
- Update the necessary fields and click "Save."

3. Deleting Appointment Type (Only Admin can delete records):

- Select an appointment type from the list and click "Delete."
- Confirm the deletion in the prompt that appears.

Generating Reports

Reports generate automatically when the Reports form is opened.

Choose the date range if applicable.