quiero hacer un proyecto .net con c# y sql, es sobre un gymnasio el mismo tendra un Administrador que gestionara todas las pantallas y funciones del proyecto, empleado de recepcion que se encarga, cobrar la mensualidad gestionar los instructores uno seran de maquinas y otros de clases de salon (les gestionara horarios y clases y funcion),y clientes los cuales seran comunes que son los que podran iniciar sesion en la pagina y registrarse para tener usuario, navegar,ver horarios,la instalacion ect.. y los clientes miembros(que son los que pagen la mensualidad ) los cuales tendran 3 tipos de membresias(solo clases de salon, solo entrenamiento maquinas, mixto) los clientes comunes se pueden registrar en la pagina pero solamente podran hacerse miembros en el local pagando la suscripcion y actualizando asi su estatus de comun a miembro segun el tipo de membresia q elijan; ademas los clientes miembros podran reportar en la pagina averias en las maquinarias o las instalaciones, y si son miembros de clases de salon reservar su lugar en el salon para una determinada clase.. el empleado tambien se encarga de revisar e informar el estado de los reportes de averias, y mas opciones que pienso agregar segun vayan surguiendo

ChatGPT

**ChatGPT**

¡Suena como un proyecto emocionante! Para comenzar, aquí hay una estructura básica de clases que podrías considerar para tu proyecto en C#:

1. **Clase Gym**: Esta clase representará el gimnasio en sí y contendrá listas de empleados, clientes y equipos. También podría contener métodos para administrar estos elementos, como agregar nuevos empleados, clientes, equipos, etc.
2. **Clase Empleado**: Esta clase representará a los empleados del gimnasio. Podría tener propiedades como nombre, cargo, salario, etc. Dependiendo del tipo de empleado, puedes considerar subclases como Administrador, EmpleadoRecepcion, Instructor, etc.
3. **Clase Cliente**: Esta clase representará a los clientes del gimnasio. Puedes tener propiedades como nombre, dirección, tipo de membresía, etc. Además, podrías tener una subclase para ClientesMiembros que contenga información adicional sobre la membresía, como tipo de membresía, fecha de vencimiento, etc.
4. **Clase Membresía**: Esta clase representará los diferentes tipos de membresía que un cliente puede tener. Puedes tener propiedades como nombre, precio, descripción, etc.
5. **Clase Equipo**: Esta clase representará los equipos en el gimnasio, como las máquinas de entrenamiento. Podría contener propiedades como nombre, estado (funcionando, en reparación, etc.), y métodos para manejar su estado.
6. **Clase ClaseSalon**: Esta clase representará las clases que se llevan a cabo en un salón. Podría tener propiedades como nombre de la clase, horario, instructor asignado, etc.
7. **Clase ReporteAveria**: Esta clase representará los reportes de averías hechos por los clientes miembros. Puedes incluir propiedades como descripción del problema, estado del reporte, etc.

Una vez que hayas definido estas clases básicas, puedes empezar a implementar la lógica de negocio y la interfaz de usuario utilizando .NET y SQL Server para la persistencia de datos. Recuerda también considerar la seguridad y la validación de datos en tu aplicación.

Si necesitas ayuda con algún aspecto específico de la implementación o tienes preguntas más detalladas, no dudes en preguntar. ¡Buena suerte con tu proyecto!

User

**You**

y para manejar el acceso de cada uno a la app, lo quiero hacer a traves de un login que dependiendo de su jerarquia puedan aparecerles unas u otras funciones, necesito la informacion para manejar el logeo y luego el sistema gestione que permiso tienen cada uno

ChatGPT

**ChatGPT**

Para manejar el acceso a la aplicación a través de un sistema de login con diferentes niveles de permisos, puedes seguir estos pasos:

1. **Base de datos de usuarios y roles**: Crea una tabla en tu base de datos SQL Server para almacenar la información de los usuarios, incluyendo sus credenciales de inicio de sesión (como nombre de usuario y contraseña) y su rol o nivel de permisos asociado. También crea una tabla para almacenar los roles y los permisos asociados a cada uno.
2. **Implementar la autenticación**: Utiliza ASP.NET Identity para manejar la autenticación de usuarios en tu aplicación .NET. ASP.NET Identity es un marco de trabajo que permite administrar la autenticación y la autorización de usuarios de manera sencilla.
3. **Definir roles y permisos**: Asigna roles a los usuarios según su jerarquía en el gimnasio. Por ejemplo, podrías tener roles como Administrador, Empleado de Recepción, Instructor, Cliente Miembro, etc. Define también los permisos asociados a cada rol, es decir, las funciones específicas que pueden realizar los usuarios de cada rol.
4. **Control de acceso basado en roles**: Utiliza la funcionalidad de autorización basada en roles de ASP.NET para controlar el acceso a las diferentes funciones de la aplicación. Esto te permitirá restringir el acceso a ciertas partes de la aplicación según el rol del usuario que ha iniciado sesión.
5. **Interfaz de usuario adaptativa**: Diseña la interfaz de usuario de tu aplicación de manera que muestre solo las funciones y opciones relevantes para el rol del usuario que ha iniciado sesión. Esto puede implicar ocultar ciertos elementos del menú o deshabilitar botones y enlaces según los permisos del usuario.
6. **Implementar el login y la gestión de sesiones**: Crea formularios de inicio de sesión donde los usuarios puedan ingresar sus credenciales. Una vez que un usuario inicia sesión correctamente, guarda su información de sesión (como su ID de usuario y su rol) en una cookie o en algún otro mecanismo de almacenamiento seguro para que la aplicación pueda identificar al usuario en futuras solicitudes.

Al seguir estos pasos, podrás implementar un sistema de login con diferentes niveles de permisos en tu aplicación .NET, lo que te permitirá gestionar el acceso de los usuarios de manera segura y eficiente.

Principio del formulario

-- Creación de la base de datos Gimnasio

USE master;

GO

DROP DATABASE IF EXISTS Gimnasio2;

GO

CREATE DATABASE Gimnasio2;

GO

USE Gimnasio2;

GO

-- Tabla TiposMembresias

CREATE TABLE TiposMembresias (

ID INT IDENTITY(1,1) PRIMARY KEY,

Descripcion VARCHAR(100),

Precio MONEY

);

GO

-- Tabla Personas

CREATE TABLE Personas (

ID INT IDENTITY(1,1) PRIMARY KEY,

DNI VARCHAR(10) UNIQUE,

Nombre VARCHAR(50),

Apellido VARCHAR(50),

Email VARCHAR(70) UNIQUE,

FechaNacimiento DATE

);

GO

INSERT INTO Personas (DNI, Nombre, Apellido, Email, FechaNacimiento)

VALUES ('12345678', 'Admin', 'Admin', 'admin@example.com', '1990-01-01');

-- Tabla CargosEmpleados

CREATE TABLE CargosEmpleados (

ID INT IDENTITY(1,1) PRIMARY KEY,

Descripcion VARCHAR(100) UNIQUE

);

GO

-- Tabla Empleados

CREATE TABLE Empleados (

ID INT IDENTITY(1,1) PRIMARY KEY,

IDPersona INT FOREIGN KEY REFERENCES Personas(ID),

IDCargoEmpleado INT FOREIGN KEY REFERENCES CargosEmpleados(ID)

);

GO

-- Tabla Miembros

CREATE TABLE Miembros (

ID INT IDENTITY(1,1) PRIMARY KEY,

IDPersona INT FOREIGN KEY REFERENCES Personas(ID),

IDTipoMembresia INT FOREIGN KEY REFERENCES TiposMembresias(ID),

FechaInicio DATE DEFAULT GETDATE(), -- Valor predeterminado configurado a la fecha actual

FechaFin DATE default DATEADD(DAY, 30,GETDATE()),

EstadoActivo BIT

);

GO

-- Tabla Roles

CREATE TABLE Roles (

ID INT IDENTITY(1,1) PRIMARY KEY,

Nombre VARCHAR(50) UNIQUE

);

GO

-- Tabla Usuarios

CREATE TABLE Usuarios (

ID INT IDENTITY(1,1) PRIMARY KEY,

IDPersona INT FOREIGN KEY REFERENCES Personas(ID),

IDRol INT FOREIGN KEY REFERENCES Roles(ID),

NombreUsuario VARCHAR(50) UNIQUE,

Clave VARCHAR(50)

);

GO

-- Tabla Cobros

CREATE TABLE Cobros (

ID INT IDENTITY(1,1) PRIMARY KEY,

IDPersonas INT FOREIGN KEY REFERENCES Personas(ID),

IDTipoMembresia INT FOREIGN KEY REFERENCES TiposMembresias(ID),

FechaCobro DATE

);

GO

CREATE TABLE Imagenes (

ID INT PRIMARY KEY identity(1,1),

IDPersona INT foreign key references Personas(ID),

Archivo VARBINARY(MAX)

);

go

--------------------------- parte de las reservas------------------

create table ClasesSalon(

ID INT IDENTITY(1,1) PRIMARY KEY,

Descripcion VARCHAR(100) unique

);

create table Salones(

ID INT IDENTITY(1,1) PRIMARY KEY,

Nombre VARCHAR(100) unique,

Capacidad int not null,

);

CREATE TABLE HorariosClases (

ID INT IDENTITY(1,1) PRIMARY KEY,

IDClaseSalon INT FOREIGN KEY REFERENCES ClasesSalon(ID),

IDSalon INT FOREIGN KEY REFERENCES Salones(ID),

Fecha DATE,

HoraInicio TIME,

HoraFin TIME,

IDInstructor INT FOREIGN KEY REFERENCES Empleados(ID) -- Nueva columna para el instructor

);

create table Reservas(

ID INT IDENTITY(1,1) PRIMARY KEY,

IDClaseSalon INT FOREIGN KEY REFERENCES ClasesSalon(ID),

IDSalone INT FOREIGN KEY REFERENCES Salones(ID),

IDMiembro INT FOREIGN KEY REFERENCES Miembros(ID),

IDHorarioClase INT FOREIGN KEY REFERENCES HorariosClases(ID),

)

ALTER TABLE HorariosClases

ALTER COLUMN HoraInicio VARCHAR(8);

ALTER TABLE HorariosClases

ALTER COLUMN HoraFin VARCHAR(8);

UPDATE HorariosClases

SET HoraInicio = LEFT(CONVERT(VARCHAR, HoraInicio, 108), 5),

HoraFin = LEFT(CONVERT(VARCHAR, HoraFin, 108), 5);

select\* from HorariosClases

---------------------------------------aca------------------------------------------

INSERT INTO TiposMembresias (Descripcion, Precio)

VALUES

('Básica', 20.00),

('Premium', 50.00);

GO

INSERT INTO Roles (Nombre)

VALUES

('Administrador'),

('Empleado'),

('Cliente');

GO

-- Selección de datos para verificar inserciones

SELECT \* FROM Roles;

SELECT \* FROM Miembros;

select \* from HorariosClases;

update Miembros set FechaInicio= '2024-03-30' , FechaFin='2024-04-30' where IDPersona= 51;

insert into miembros (IDPersona,IDTipoMembresia) values (3,2)

INSERT INTO Miembros (IDPersona, IDTipoMembresia) OUTPUT INSERTED.ID VALUES (1, 1)

select \* from HorariosClases

delete from Roles where Nombre='Instructor de Salon'

insert into Usuarios(IDPersona,IDRol,NombreUsuario,Clave ) values (1,1,'Admin','Admin');