Course Number: 420-DW4-AS
Course Title: Web Server Applications Development II
Session: Winter 2024
Group: 7270

FINAL PROJECT REPORT

TITLE: Property Rental Management Web Site

DUE DATE: April 22, 2024

SUBMITTED

TO

TEACHER: QUANG HOANG CAO

BY

Bonnie Ives de Castro Nunes – 2230019

Contents

I. PROJECT DESCRIPTION

Property manager website is a web application that aims to make your search, for an apartment to rent less stressful and the process to get it easier. For property owners, this application is useful to display their apartments and sign their contracts. For managers, we can make your communication between potential tenants and property owners more efficient. For administrators, we aim to make them to have full control over their customers and employee’s actions.

Developed using ASP.Net using the method Code First, the application has full control over the database operation applying CRUD in the registered tables.

II. PROJECT DEVELOPMENT

PHASE I – ANALYSIS

The application has four user roles:

- **Administrator and Property Owner**: Has full control over the website, can perform CRUD operations for any property; can update, delete, search, and list any potential tenant account.
- **Property Manager**: Can perform CRUD operations for the properties and apartments under their management, can confirm and schedule appointments with tenants, can answer to tenant's messages, can keep track on apartment status (available or unavailable), and can report any event to the property owner.
- **Potential Tenants**: Can create an account through the website, can see a list of the available apartments, send a message, and schedule an appointment with the property manager.

PHASE II – DESIGN

The project was made using as method the MVC code first. MVC is a design pattern often used in software development to separate the responsibilities and make the code more organized and easier to maintain. The model represents the data and business logic, the view is responsible for presenting the user interface, and the controller is responsible for handling incoming requests from the user, process it, then return the appropriate response.

Code First is an approach in Entity Framework that allows developers to define the data model of an application without creating a database schema manually.

Models

Apartments.cs

ApartmentId (Primary Key), ApartmentNumber, OwnerId (Foreign Key), ManagerId (Foreign Key), PropertyId (Foreign Key), StatusId (Foreign Key), Image, ApartmentTitle, ApartmentValue.

Appointments.cs

AppointmentId (Primary Key), ApartmentId (Foreign Key), SenderId (Foreign Key), Confirmation, DateTime.

Categories.cs

CategoryId (Primary Key), Description.

Enum Description: Administrator, Owner, Tenant Manager.

Contracts.cs

ContractID (Primary Key), InitialDate, FinalDate, PayDay, TenantId (Foreign Key), ApartmentId (Foreign Key).

Events.cs

EventId (Primary Key), ApartmentId (Foreign Key), Description.

Messages.cs

MessageId (Primary Key), ApartmentId (Foreign Key), SenderId (Foreign Key), ReceiverId (Foreign Key), Description.

Properties.cs

PropertyId (Primary Key), CivicNumber, Address, City, Zip, Province.

Enum Province: AB, BC, MB, NB, NL, NS, NT, NU, ON, PE, QC, SK, YK.

Status.cs

StatudId (Primary Key), StatusDescription.

Enum StatusDescription: Available, Unavailable.

Users.cs

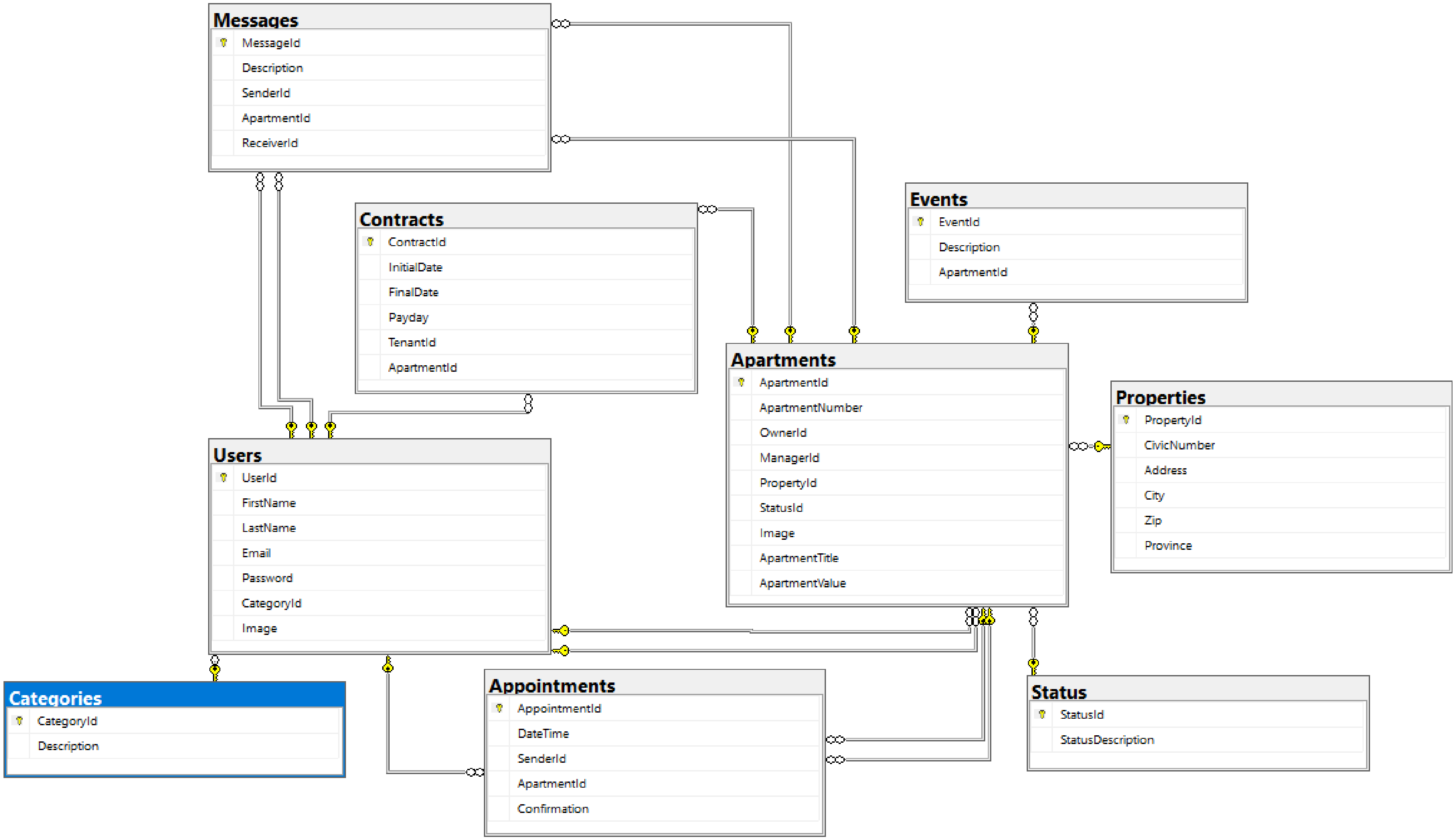
UserId (Primary Key), FirstName, LastName, Email, Password, CategoryId (Foreign Key), Image.

The following classes are used as auxiliary to help on a better execution of the web application.

ApartmentModel.cs, AppointmentModel.cs, ContractModel.cs, EventModel.cs, LoginViewModel.cs, MessageModel.cs, PropertyApartmentModel.cs, PropertyModel.cs, UserApartmentViewModel.cs, UserModel.cs

Database

The database is hosted locally using SQL Management Studio with the name FinalProject_DB. Here follows the ERD.



Controllers

- **ApartmentsController:** Responsible for the actions related to apartments.
- **AppointmentsController:** Responsible for the actions related to appointments.
- **CategoriesController:** Responsible for the actions related to contracts.
- **EventsController:** Responsible for the actions related to events.
- **HomeController:** Responsible for actions related to authentication, login and logout.
- **MessagesController:** Responsible for the actions related to Messages.
- **PropertiesController:** Responsible for the actions related to Properties and apartments since the rent is related to both.
- **StatusController:** Responsible for the actions related to the status.
- **UsersController:** Responsible for the actions related to the different categories of users.

Views

- **Apartments:** Create.cshtml, Delete.cshtml, Details.csthml, Edit.cshtml, Index.cshtml.
- **Appointments:** Create.cshtml, Delete.cshtml, Details.csthml, Edit.cshtml, Index.cshtml.
- **Categories:** Create.cshtml, Delete.cshtml, Details.csthml, Edit.cshtml, Index.cshtml.
- **Contracts:** Create.cshtml, Delete.cshtml, Details.csthml, Edit.cshtml, Index.cshtml.
- **Events:** Create.cshtml, Delete.cshtml, Details.csthml, Edit.cshtml, Index.cshtml.
- **Home:** About.cshtml, Contact.cshtml, Login.csthml, Signup.csthml, Index.cshtml.
- **Messages:** Create.cshtml, Delete.cshtml, Details.csthml, Edit.cshtml, Index.cshtml.
- **Properties:** Create.cshtml, Delete.cshtml, Details.csthml, Edit.cshtml, Index.cshtml.
- **Shared:** _Layout.cshtml, Error.cshtml.
- **Status** Create.cshtml, Delete.cshtml, Details.csthml, Edit.cshtml, Index.cshtml.
- **Users:** Create.cshtml, Delete.cshtml, Details.csthml, Edit.cshtml, Index.cshtml.

PHASE III IMPLEMENTATION

- **IDE:** Visual Studio 2022.
- **FRAMEWORK:** ASP.NET MVC .NET 8.0
- **DATABASE:** SQL Server Studio.

PHASE IV TESTING THE PROGRAM

FUNCTIONAL REQUIREMENT	TEST RESULT
Administrator	
Login on the website	Done
Logout the website	Done
Read all appointments / properties	Done
Read all apartments / properties	Done
Create an apartment / property	Done
Delete an apartment / property	Done
Update an apartment / property	Done
Read all categories	Done
Create a contract	Done
Update a contract	Done
Delete a contract	Done
Read all contracts	Done
Read all events	Done
Read all messages	Done
Read all status	Done
Read all users	Done
Update a user	Done
Delete a user	Done
Create a user	Done
Owner	
Read appointments from their own apartments	Done
Read their own apartments / properties	Done
Update their own apartments / properties	Done
Delete their own apartments / properties	Done
Create their own apartments / properties	Done
Read all categories	Done
Read their own contracts	Done
Read their own events	Done
Read their own messages	Done
Read all status	Done
Create	Done
Read all users	Done
Update a user	Done
Delete a user	Done

Create a user	Done
Tenant	
Read their own appointments	Done
Update their own appointments	Done
Delete their own appointments	Done
Create their own appointments	Done
Real all available apartments / properties	Done
Read their own messages	Done
Update their won messages	Done
Delete their own messages	Done
Create their own messages	Done
Create their own user	Done
Update their own user	Done
Delete their own user	Done
Read their own user	Done
Manager	
Create their own appointments	Done
Update their own appointments	Done
Read their own appointments	Done
Delete their own appointments	Done
Reat their own apartments / properties	Done
Update their own apartments / properties	Done
Delete their own apartments / properties	Done
Create an apartment / property	Done
Read their own contracts	Done
Update their own contracts	Done
Delete their own contracts	Done
Create their own contracts	Done
Create their own events	Done
Update their own events	Done
Delete their own events	Done
Read their own events	Done
Create their own messages	Done
Update their own messages	Done
Read their own messages	Done
Delete their own messages	Done

III. CONCLUSION

Developing a web application is a process that requires high skills in programming and critical thinking. The programmer must be able to manipulate the chosen framework and understand the concepts behind the used models.

The choice for code first comes from the idea that the creation script for the database is not needed, and the programmer will be able to perform the migration without the need of manipulating the database directly into the server, which makes the application building a faster task to be performed.

Additionally, the use of MVC makes the application easier to maintain, where the programmer can directly identify the issues and address correctly the problem to be solved. Also, the programmer will not be concerned about styling and structure the website, leaving this responsibility for the views. Also, using the controllers can easily separate the responsibilities and control what each user can see and control into the application.

IV. BIBLIOGRAPHY

1. OpenAI. (2024). *ChatGPT*. https://chat.openai.com/chat
2. Microsoft. (2024). *.NET C# Documentation*. *https://learn.microsoft.com/en-us/dotnet/csharp/*
3. Microsoft. (2024). *ASP.NET Documentation*. https://dotnet.microsoft.com/pt-br/apps/aspnet
4. Stack Exchange In. (2024). *Stack Overflow*. https://stackoverflow.co/