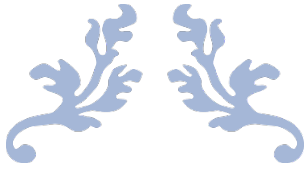

CHATROOMS WEB APP

Etienne Theunissen
Liam Venter

AUGUST 4, 2023

SGTK6FM21
Tyger Valley

Table of Contents

Section A

Question 1

1.1

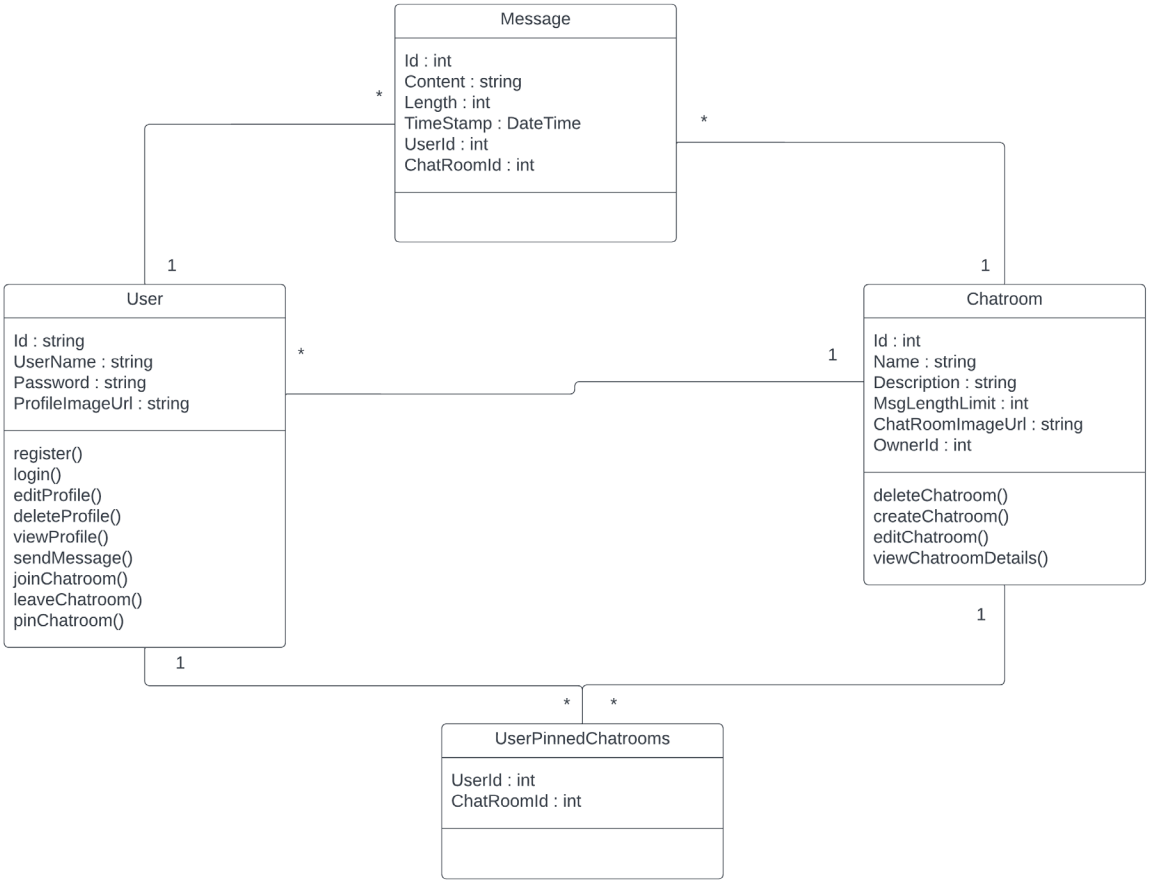

Figure 1: Class Diagram

1.2
Introduction
Aim and Scope
We were tasked with creating a live chat web application using ASP.NET Core where we had to use a framework as well as a pattern. We decided to use the Model View Controller (MVC) framework to provide a solid structure to the application.

There must be at least one chatroom that has seamless real-time communication and encompasses essential messaging functionalities, including: unique usernames for user identification, timestamping of messages, support for simple text formatting such as bold and italic.

Finally, we must incorporate two text files, one for error logging and other for chat logging (to keep a record of the conversation history).

Overview of the implemented software solution
We utilised ASP.NET Core using the MVC framework to create our web application.

Additionally, we used several services, libraries, and plugins to enrich the user experience.

Instead of one global chatroom, we created many chatrooms for the user to join. Additionally, the user can create, edit, and delete their own chatrooms.

Each chatroom has real time chat, with timestamps. Messages are saved to a database and text file. Additionally, messages from the database that were not sent live are also displayed.

For record keeping purposes we used text file logging to log errors to an error log text file, and to log messages to a chat log file. A new errorlog and chatlog file are created each day for comprehensive record keeping.

Finally, we utilized a database, SQL Server, for permanent storage of users, chatrooms, and messages.

We ensured that each user’s messages are displayed under a username and shows a timestamp of the user's message as well as a profile picture that the user chose. We then allowed the user to format their text such as making text bold and allowing them to highlight text. The users can make their accounts and then create their own chatrooms. Users are able to pin chatrooms to their dashboards for them to easily access their chatrooms. The chatrooms show the history of the previous chats.

This web application allows a fun way for users to chat to one another on specific topics related to the chatroom name and description.

Conclusion

We successfully created an ASP.NET Core web application with using MVC framework, simulating a live chatroom application.

We learned how to make use of patterns, services, plugins, libraries, and packages.

SignalR, a JavaScript library, allowed us to update the live chat without having to refresh the page, enhancing the user experience. A service, TinyMCE , allowed us to enhance the chatting experience by enabling users to edit their text such as making it bold and highlighting it, as well as sending pictures and videos via URL's.

Identity Core Framework enabled robust user authentication and security. It is responsible for hashing the user password and enables user roles.

Entity Core Framework enabled us to create a relational SQL database off of our Models. Additionally it allowed us to easily change our models and migrate and update the database, making development and rapid change seamless.

Serilog, a third-party logger that we installed from NuGet Package Manager allowed us to easily log errors and messages to separate text files for comprehensive record keeping.

This project allowed us to learn how to successfully do version control where we made use of Git, and GitHub to collaborate with each other simultaneously. This ultimately gave us the opportunity to learn how to successfully communicate with one another.

Cloudinary, a cloud photo service, allowed us to enhance the user experience by allowing users to have profile pictures and to create chatrooms with pictures. Cloudinary saves the photo and returns a URL that can be stored in our regular SQL Sever database.

Finally, Azure App Service allowed us to effortlessly host the website for all to use.

For us as developers this project gave us the opportunity to learn a lot from making use of services and packages, version control, and effectively communicating with one another.

The application can be used for people that want to talk about different topics and allow their friends to join the chat and have a great time.

Table of Figures

Bibliography

Blumhardt, N., 2021. *Serilog Configuration Basics.* [Online]
Available at: https://github.com/serilog/serilog/wiki/Configuration-Basics
[Accessed 25 August 2023].

Blumhardt, N., 2021. *Serilog Getting Started.* [Online]
Available at: https://github.com/serilog/serilog/wiki/Getting-Started
[Accessed 24 August 2023].

Kılıçarslan, S., 2020. *Deploy an ASP.NET Core App with EF Core and SQL Server to Azure.* [Online]
Available at: https://medium.com/net-core/deploy-an-asp-net-core-app-with-ef-core-and-sql-server-to-azure-e11df41a4804
[Accessed 25 August 2023].

Material Design Bootstrap, 2023. *Bootstrap 5 Chat component.* [Online]
Available at: https://mdbootstrap.com/docs/standard/extended/chat/
[Accessed 20 August 2023].

Material Design Bootstrap, 2023. *Bootstrap 5 Profile page & profile cards.* [Online]
Available at: https://mdbootstrap.com/docs/standard/extended/profiles/
[Accessed 23 August 2023].

Microsoft, 2022. *Overview of ASP.NET Core MVC.* [Online]
Available at: https://learn.microsoft.com/en-us/aspnet/core/mvc/overview?view=aspnetcore-7.0
[Accessed 17 August 2023].

Microsoft, 2022. *Tutorial: Deploy an ASP.NET app to Azure with Azure SQL Database.* [Online]
Available at: https://learn.microsoft.com/en-us/azure/app-service/app-service-web-tutorial-dotnet-sqldatabase
[Accessed 25 August 2023].

Microsoft, 2023. *Logging in .NET Core and ASP.NET Core.* [Online]
Available at: https://learn.microsoft.com/en-us/aspnet/core/fundamentals/logging/?view=aspnetcore-7.0
[Accessed 24 August 2023].

Microsoft, 2023. *Logging in C# and .NET.* [Online]
Available at: https://learn.microsoft.com/en-us/dotnet/core/extensions/logging?tabs=command-line
[Accessed 20 August 2023].

Microsoft, 2023. *Part 3, Razor Pages with EF Core in ASP.NET Core - Sort, Filter, Paging.* [Online]
Available at: https://learn.microsoft.com/en-us/aspnet/core/data/ef-rp/sort-filter-page?view=aspnetcore-7.0
[Accessed 20 August 2023].

Microsoft, 2023. *Quickstart: Deploy an ASP.NET web app.* [Online]
Available at: https://learn.microsoft.com/en-us/azure/app-service/quickstart-dotnetcore?tabs=net70&pivots=development-environment-vs
[Accessed 18 August 2023].

Microsoft, 2023. *Razor Pages with Entity Framework Core in ASP.NET Core - Tutorial 1 of 8.* [Online]
Available at: https://learn.microsoft.com/en-us/aspnet/core/data/ef-rp/intro?view=aspnetcore-7.0&tabs=visual-studio
[Accessed 18 August 2023].

Microsoft, 2023. *Tutorial: Get started with EF Core in an ASP.NET MVC web app.* [Online]
Available at: https://learn.microsoft.com/en-us/aspnet/core/data/ef-mvc/intro?view=aspnetcore-7.0
[Accessed 19 August 2023].

TinyMCE, 2023. *tinymce.Editor API Documentation.* [Online]
Available at: https://www.tiny.cloud/docs/tinymce/6/apis/tinymce.editor/#getContent
[Accessed 20 August 2023].