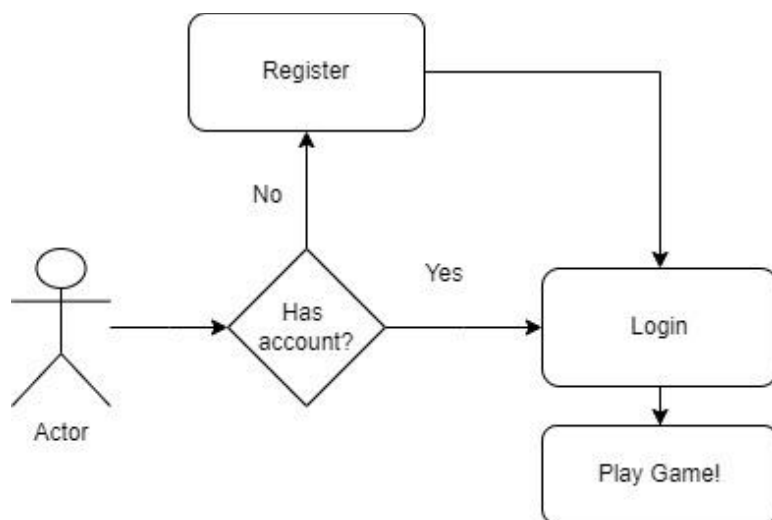


# Math Game Project Documentation

## Brief Introduction

Math Game is a web application where the user can register, login and play a game. The game structure is simple, in a session with 4 other players he competes to answer an expression the fastest. The expressions are generated and added to the tables of guessing on 5-minute intervals. If the guess is correct the player earns points, if not better luck next time.

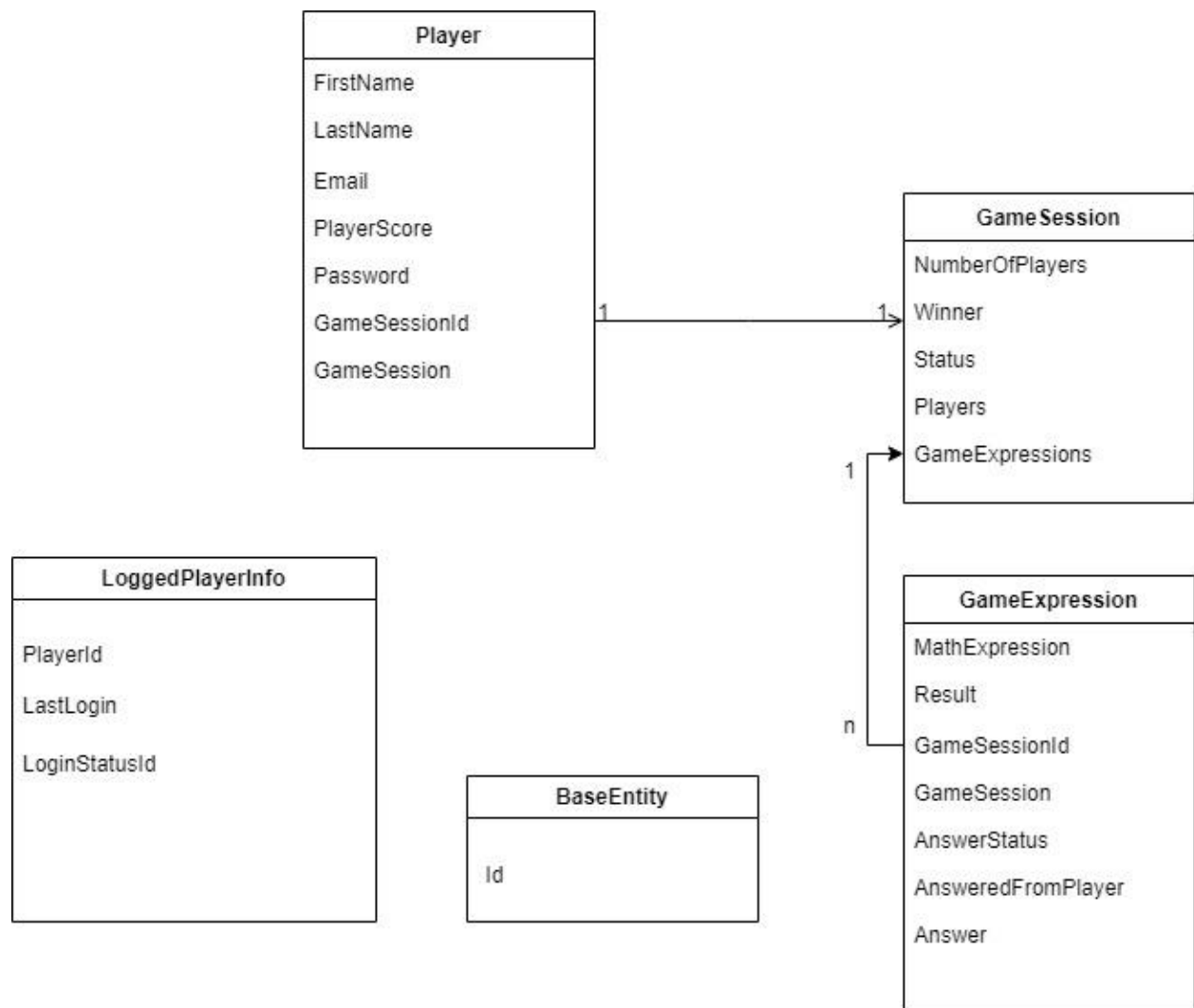
Here is a simple Use Case scenario.



## Infrastructure

- The game is written in .NET Core API as the server and Angular as the client side. The backend is written in N Tier architecture. I am using Dependency Injection in order to reduce coupling between classes and make the code more readable and maintainable. I would also suggest segregating the application into microservices in the future, maybe creating separate service for player management and one for game hosting and everything related to the game sessions and expressions.

## UML



## Security

For security i decided to go with my own authentication process which is using JWT tokens. When the player logs in successfully to our application, we generate a token and we store the token inside of the browser cookie. Then the client checks for the cookie and sends the token to us with an interceptor.

As for Password management we are Hashing the user password with SHA512 algorithm. We also have a redis check implemented that allows user to miss login 3 times. After 3 missed times we lock the account and he needs to wait 24 hours to re-try. In the future maybe we can implement 2FA on login or captcha, and on register we could implement smtp client and a verification email that the user needs to confirm and then access the account.

I was unsure whether to go with .NET Identity or use my own custom authentication but i decided to go with my own custom version since it gives me full control over the user authentication process, allowing me to tailor it to my specific requirements.

## Used Libraries

So for the game logic i decided to go with Hangfire as my background service that is responsible of generating new expressions for the users to guess. Then I implemented SignalR in order to update the client of any new expression , and also when an expression is guessed by different player we need to update the expression for all players in that session. I am also using singalR for updating the number of current online users so the player can see how many other players are currently playing the game. I also decided to use AutoMapper for easier mapping of entities and Dtos.

On the front end I am using libraries like toastr for displaying success and error messages. I am using angular materials too.

For database i decided to go for relational database mssql , and for version controll i was using git.

## Game Preview

A sample of how the game looks like for 2 different players on the same GameSession

The image displays two side-by-side browser windows showing the MathGameApp interface. Both windows are at the URL `localhost:4200/game` and feature a top navigation bar with "Back to Home" and "Logout" buttons.

**Left Window (Player 1):** The score is 4. The table shows the following data:

Number	Expression	Your Answer	Result
1	$10 + 4$	14	OK
2	$2 - 8$	-6	OK
3	$3 + 7$	MISSED	Failed
4	$1 + 1$	<input type="text" value="Enter your answer"/>	Pending

**Right Window (Player 2):** The score is 10. The table shows the following data:

Number	Expression	Your Answer	Result
1	$10 + 4$	MISSED	OK
2	$2 - 8$	MISSED	OK
3	$3 + 7$	9	Failed
4	$1 + 1$	<input type="text" value="Enter your answer"/>	Pending