Dokumentation

Stronk

IMS Basel

M133

Contents

[Use Case Diagram 2](#_Toc119532548)

[ERM 3](#_Toc119532549)

[Arbeitsjournal 4](#_Toc119532550)

[10.11.2022 4](#_Toc119532551)

[11.11.2022 4](#_Toc119532552)

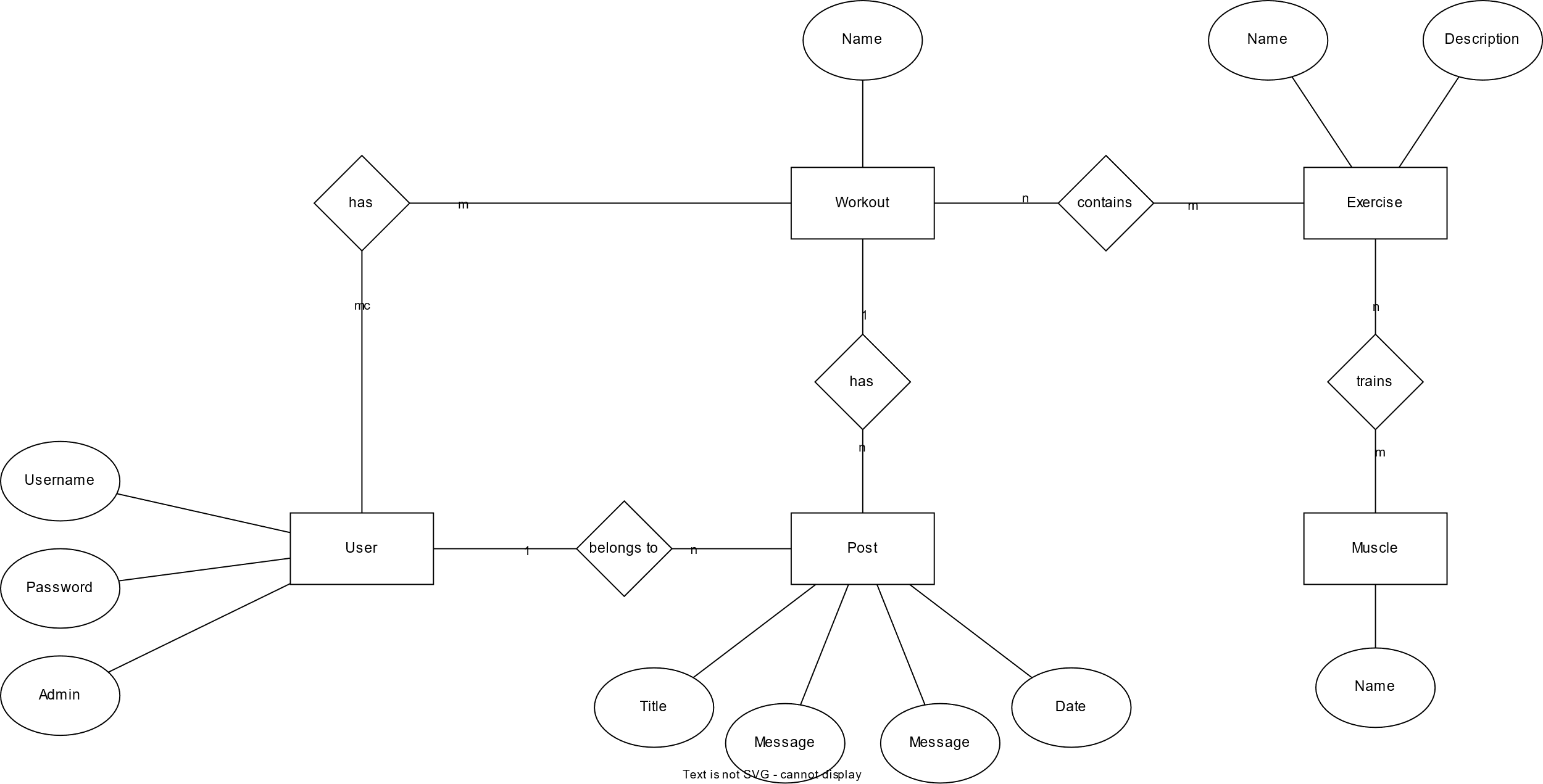
[14.11.2022 4](#_Toc119532553)

[15.11.2022 5](#_Toc119532554)

[16.11.2022 5](#_Toc119532555)

# Use Case Diagram

# ERM



# Arbeitsjournal

## 10.11.2022

Ich habe mich dazu entschieden, Rider von Jetbrains, als Editor zu verwenden. Ich habe mich gegen Visual Studio entschieden, da ich mit den Produkten von Jetbrains öfters gearbeitet habe und mir das Layout von Visual Studio wenig gefällt wie das von den Jetbrains Produkte.

## 11.11.2022

Arbeitszeiten: 17:50 – 18:10

Ich habe es geschafft, die Datenbank mit C# zu verbinden. Dafür habe ich das ‘System.Data.SqlClient’ package heruntergeladen. Um die Verbindung herzustellen benötigte ich einen Connection String. Dieser String beinhaltet unter anderem Username und Passwort des Datenbank Benutzers. Da meine Datenbank aber Windows Authentication benutzt, konnte ich diese zwei Parameter entfernen und sie durch ‘Integrated Security=SSPI’ ersetzt.

## 14.11.2022

Arbeitszeiten: 15:15 – 17:20

Verwendete Ressourcen:

* <https://youtu.be/3THbdVew2WI>
* <https://www.aspsnippets.com/Articles/ASPNet-MVC-Form-Submit-Post-example.aspx>

Problem: Ich wollte ein Form machen, welches an den Controller geschickt wird, um die Daten zu verwalten. Im Parameter des Controllers war eine Model Instanz. Die Daten wurden aber nicht an den Parameter weitergegeben.

Lösung: Ich dachte die ganze Zeit, das Problem wäre beim View oder Controller, es war aber beim Model. Ich habe vergessen, den Attributen des Models Getter und Setter zu geben. Somit konnte auch nicht auf die Attribute zugegriffen werden.

Arbeitszeiten: 17:55 – 18:50

Verwendete Ressourcen:

* <https://learn.microsoft.com/en-us/sql/t-sql/functions/hashbytes-transact-sql?view=sql-server-ver16>

Prozedur erstellt, um sich registrieren zu können. Ich musste lernen wie man in der Datenbank Hashs macht. Ich war mich nicht sicher welcher Algorithmus ich wählen sollte. Ich war mir nicht sicher ob ich ‘MD5’ oder ‘SHA2\_512’ wählen sollte. Ich habe mich am Ende für ‘SHA2\_512’.

Arbeitszeiten: 19:10 – 19:35

Für das Einloggen eine Prozedur erstellt, welches den Benutzernamen und das Passwort kontrollierten.

## 15.11.2022

Arbeitszeiten : 18:30 – 18:40

Ich habe mich dazu entschieden einen Index auf meine Tabelle namens tbl\_Muscle zu setzten. Diese Tabelle beinhaltet verschiedene Muskelgruppen. Benutzer der Webseiten können die Muskelgruppen nicht ändern oder neue erfinden. Sie müssen nur bei dem Erstellen einer neuen Übung ausgewählt werden können. Dies ist optimal für einen Index, weil der Index am besten funktioniert, wenn in der Tabelle viel abgefragt und wenig eingefügt werden muss.

Arbeitszeiten: 18:00 – 18:20

Verwendete Ressourcen:

* <https://rietta.com/blog/bcrypt-not-sha-for-passwords/>

Aufgrund weiterer Recherche ist mir aufgefallen, dass ‘SHA2­\_512’ doch nicht so gut ist, um Hashs von Passwörtern zu erstellen. Ich habe mich dazu entschieden BCRYPT zu verwenden. Das Problem war, dass das Erstellen von BCRYPT Hashs in SQL-Server nicht automatisch vorhanden ist.

## 16.11.2022

Arbeitszeiten 14:45 – 15:00

Verwendete Ressourcen:

* <https://youtu.be/5PYFUscoPCc>

Da ich mich dazu entschieden habe BCRYPT als Hash zu verwenden, musste ich dies in C# selber und nicht in der Datenbank machen. Dazu habe ich mir ein kurzes YouTube Video angeschaut. Darauf habe ich mir ein NuGet Package namens ‘BCrypt.Net’ heruntergeladen und konnte damit ganz einfach Hashs von Passwörtern erstellen. Ich musste aber auch meine Login und Registrierung Prozedur anpassen, da ich die Hashs nichtmehr in der Datenbank machte und sie da ebenfalls nicht mehr kontrollierte.