Dokumentation

Stronk-M133

IMS Basel

Raphael Gisler

Inhalt

[Use Case Diagram 2](#_Toc120168072)

[Login Beschreibung 2](#_Toc120168073)

[Posts Beschreibung 3](#_Toc120168074)

[ERM 4](#_Toc120168075)

[Datenbank 5](#_Toc120168076)

[User 5](#_Toc120168077)

[Post 5](#_Toc120168078)

[Arbeitsjournal 5](#_Toc120168079)

[10.11.2022 5](#_Toc120168080)

[11.11.2022 5](#_Toc120168081)

[14.11.2022 6](#_Toc120168082)

[15.11.2022 6](#_Toc120168083)

[16.11.2022 7](#_Toc120168084)

[17.11.2022 7](#_Toc120168085)

[20.11.2022 7](#_Toc120168086)

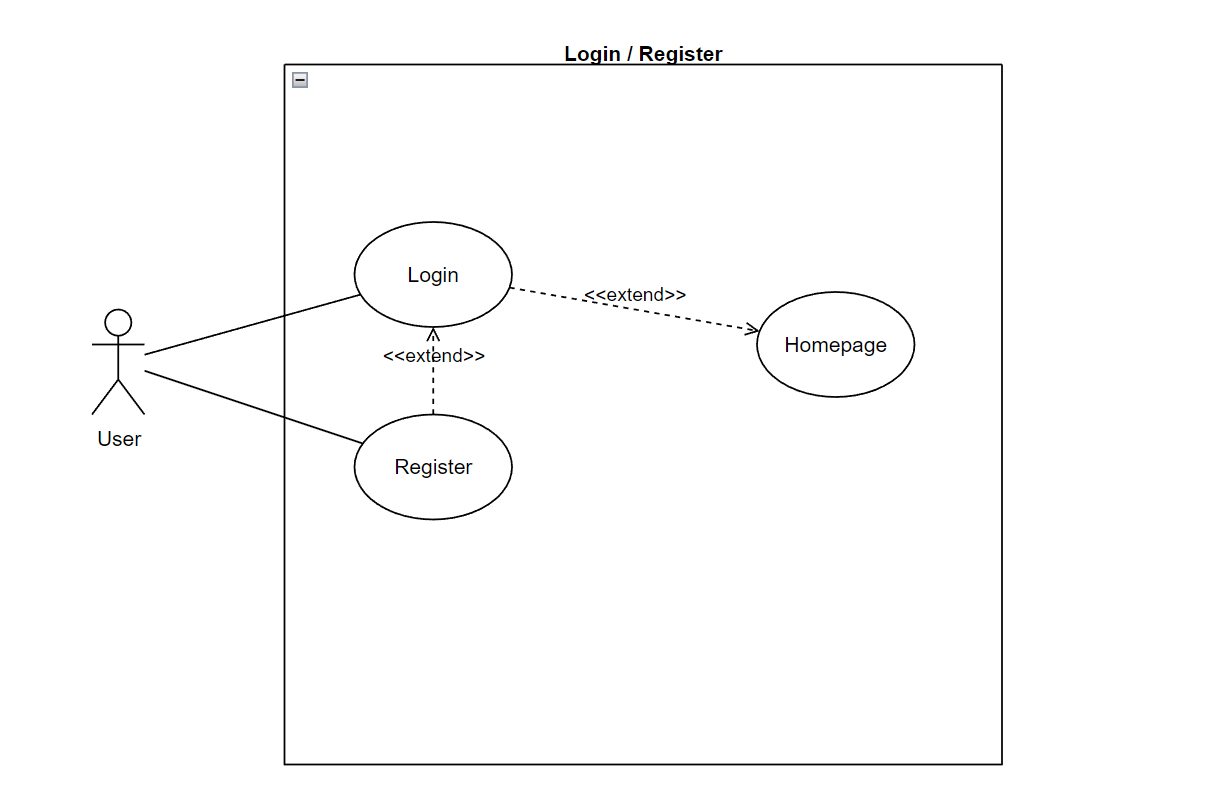
[21.11.2022 8](#_Toc120168087)

[22.11.2022 8](#_Toc120168088)

# Use Case Diagram

## Login Beschreibung

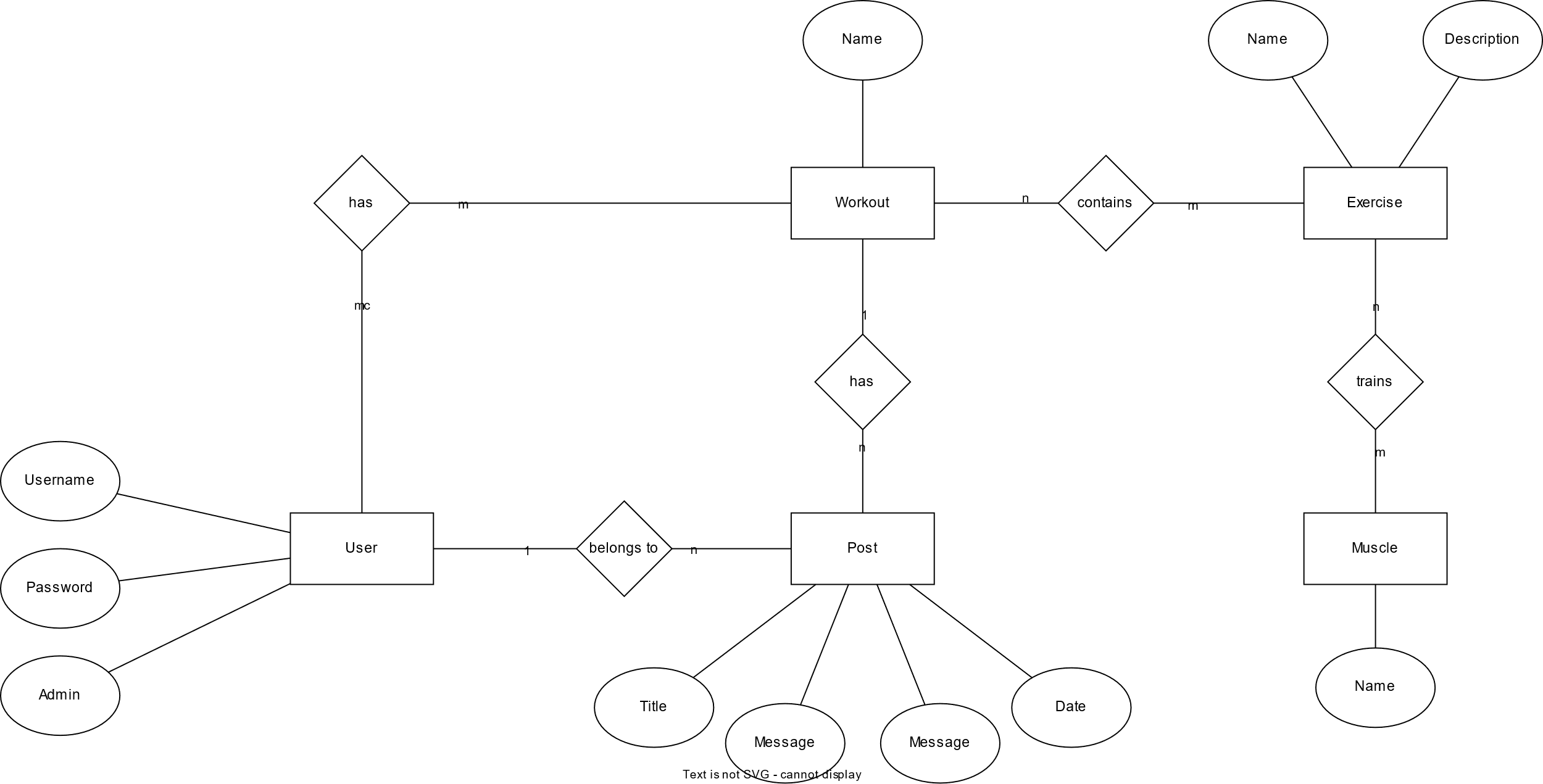
|  |  |
| --- | --- |
| Name | Login / Registrierung |
| Ziel | Eine Person kann sich auf der Website Registrieren, und Einloggen |
| Kategorie |  |
| Vorbedingung | * Eine Person, befindet sich auf der Website und hat noch kein Login gemacht. * Die Datenbank ist vollständig aufgestellt. |
| Nachbedingung | Ein neuer User wurde in der Datenbank erfasst und eine Person kann über diesen User auf die Website gelangen. |
| Akteure | User |
| Auslösendes Ereignis | Person, welche kein Login gemacht hat, ist auf der Website. |
| Beschreibung | 1. Eine Person, welche sich nicht registriert hat, gelangt auf die Website. 2. Sie navigiert von dem Login zur Registrierungsseite. 3. Benutzername und Password werden ins Formular eingetragen und die Person drückt auf den Registrierungsknopf. 4. Ein neuer User wird in der Datenbank erstellt. 5. Die Person wird zur Login Seite gesendet. 6. Mit den registrierten Daten kann sich die Person nun einloggen, um auf die Homepage zu gelangen. |
| Erweiterungen |  |
| Alternativen | * Der gewählte Benutzername der Person ist bereits verwendet und es muss einen neuen gewählt werden. * Die Person ist bereits registriert und betätigt direkt das Einloggen. * Die Person entscheidet sich dagegen sich einzuloggen und verlässt die Website nach dem Registrieren. * Die Person gelangt auf die Website und verlässt sie wieder. |



## Posts Beschreibung

|  |  |
| --- | --- |
| Name | Posts |
| Ziel | Ein Post (Veröffentlichung ein oder mehrerer Workouts) kann gesehen, veröffentlicht, bearbeitet und gelöscht werden. Nur ein Administrator kann die Posts anderer löschen. |
| Kategorie |  |
| Vorbedingung | * Ein eingeloggter User befindet ist auf der Homepage * Posts sind bereits auf der Homepage vorhanden |
| Nachbedingung |  |
| Akteure | User, Administrator |
| Auslösendes Ereignis |  |
| Beschreibung | 1. Eine Person, welche sich nicht registriert hat, gelangt auf die Website. 2. Sie navigiert von dem Login zur Registrierungsseite. 3. Benutzername und Password werden ins Formular eingetragen und die Person drückt auf den Registrierungsknopf. 4. Ein neuer User wird in der Datenbank erstellt. 5. Die Person wird zur Login Seite gesendet. 6. Mit den registrierten Daten kann sich die Person nun einloggen, um auf die Homepage zu gelangen. |
| Erweiterungen |  |
| Alternativen |  |

# ERM



# Datenbank

## User

* **Username [NVARCHAR (90)]**

Hier wird der Benutzername, welcher beim Registrieren eingetragen wurde, festgehalten. Der Benutzername wird benötigt, um sich einzuloggen und zu sehen, wem einem Post gehört. Der Benutzername darf maximal eine Zeichenlänge von 50 Zeichen beinhalten.

* **Password [CHAR (32)]**

Eine Web-Applikation mit verschiedenen Benutzern benötigt Passwörter für die jeweiligen Benutzern. Aus Sicherheitsgründen dürfen sich Passwörter nicht so wie sie sind in der Datenbank befinden. Falls es zu einem Data breach kommen würde, könnten alle Passwörter zusammen mit den Benutzernamen an die Öffentlichkeit gelangen. Deswegen muss man Passwörter hashen oder encrypten. Ich habe mich für das hashen entschieden, weil meine Web-Applikation nicht die Funktion benötigt, die Passwörter wiederherzustellen.

Ich habe mich für die SHA256 Algorithmus entschieden, da es einen optimalen Kompromiss zwischen Performance und Sicherheit hat. Eine weitere Option wäre SHA512 aber mit diesem Algorithmus benötigt man doppelt so viel Speicherplatz und es erhöht die Sicherheit nur ein wenig.

* **Admin [BIT]**

Auf der Web-Applikation gibt es zwei verschiedene Benutzer. Es gibt den normalen Benutzer und den Administrator. Diese Spalte in der Tabelle zeigt an, ob ein Benutzer ein Admin ist oder nicht.

### Post

* **Title [NVARCHAR (50)]**

Title

* **Message [TEXT]**
* **Date [DATE]**

# 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.

## 17.11.2022

Arbeitszeiten: 18:20 – 18:55

Um meine Use Cases zu erstellen, wollte ich mit einem einfacheren Anfangen, nämlich dem Login und Register. Dieser Prozessvorgang ist sehr kurz.

Arbeitszeiten: 20:35 – 21:05

Nachdem wir im Unterricht über das Hashen von Passwörtern gesprochen haben, wurde mir klar, dass meine Passwörter gar nicht Encrypted sein müssen, sondern nur gehashed sein müssen. Deshalb habe ich mich wieder dazu umentschieden, auf meine alte Hash Methode zurück zu greifen, wo ich das Hashen und das Überprüfen, alles in der Datenbank machen würde. Somit bin ich von BCRYPT zurück auf SHA2\_256 zurück gegangen.

Arbeitszeiten: 21:20 – 22:30

Verwendete Ressourcen:

* <https://youtu.be/p2kzp2d0a4A?list=PL82C6-O4XrHde_urqhKJHH-HTUfTK6siO>
* <https://learn.microsoft.com/en-us/aspnet/mvc/overview/getting-started/getting-started-with-ef-using-mvc/implementing-basic-crud-functionality-with-the-entity-framework-in-asp-net-mvc-application>

Um mir die Grundbausteine zu geben, um das Login und Register zu machen, habe wollte ich die Models erstellen. Dafür habe ich die Struktur meiner Datenbank übernommen, um so meine Models zu erschaffen.

## 20.11.2022

Arbeitszeiten: 22:10 – 22:45

Entity Framework gelöscht

System.Data.SqlClient verwendet

## 21.11.2022

Arbeitszeiten: 14:00 – 15:15

Verwendete Ressourcen:

* <https://learn.microsoft.com/en-us/aspnet/core/fundamentals/app-state?view=aspnetcore-7.0>

Problem: Ich habe ein fertiges Login aber keine Session, welche gestartet wird, sobald man sich eingelogged hat.

Ich habe versucht über die Dokumentation von Microsoft eine Lösung zu dem Problem zu finden. Ich wusste wo der Start der Session sein musste, nämlich nach der Verifizierung des Logins, aber das Problem war, dass ich nach dem Einloggen, einen Redirect zur Homepage mache. Das Problem liegt darin, dass die Session nur auf den Seiten funktionieren, auf denen der Controller Zugriff hat. Da ich einen separaten Controller für mein Login und die Homepage habe, konnte ich keine Session starten, mit welcher ich auf der Homepage auf Sessionvariabeln zugreifen kann. Bis jetzt konnte ich das Problem nicht lösen.

Arbeitszeiten: 21: 15 – 22:30

Problem: Select Abfrage einer N zu MC Beziehung.

## 22.11.2022

Arbeitszeiten: 16:40 – 15:20

Verwendete Ressourcen:

* <https://stackoverflow.com/questions/14138872/how-to-use-sessions-in-an-asp-net-mvc-4-application>
* <https://stackoverflow.com/questions/10785245/redirect-to-action-in-another-controller>

Problem: Session funktionieren immer noch nicht