

# Protokoll

## 1. Link zu GIT

Link zu Semesterprojekt auf eigenem GitHub Profil:

<https://github.com/FireGhostPhino/MonsterTradingCardsGame-3>

Da aus unbekannten Gründen kurz nach Start des Projekts das Projekt zwei Mal kaputt ging (komische Bugs, ließ sich nicht mehr starten, ...), habe ich das Projekt nach längerem, gescheiterten Versuchen zu reparieren neu aufgesetzt. Da durch das neu aufsetzen jedoch auch die bisherige Historie verloren ging, hier die Links zu den alten Projekten:

(Aktuelles Endergebnis des Projekts, siehe obiger Link!)

<https://github.com/FireGhostPhino/MonsterTradingCardsGame>

<https://github.com/FireGhostPhino/MonsterTradingCardsGame-2>

## 2. Design

## 3. Lessons Learned

Nachdem in diesem Fach das erste Mal C# in diesem Studium gelehrt wurde, gab es neue Informationen und Wissenswertes zu lernen.

Visual Studio:

Mit Visual Studio 2022 habe ich schon gearbeitet (Beispielsweise wird parallel zu SWEN1 in Innolab1 Visual Studio verwendet). Neu war jedoch die Versionsverwaltung GIT und GitHub in und mit Visual Studio zu verwenden. Das ist ein praktisches Feature, um beispielsweise bei Fehlern oder aus anderen Gründen den aktuellen Entwicklungs-Branch zu verwerfen oder bestimmte Versionen zurück zu gehen.

Weiters war die Möglichkeit einer Verbindung von der Datenbank (PostgreSQL Docker Container) zu Visual Studio neu. Es kann eine Verbindung erstellt werden und dann über diese alle Tabellen an- und einsehen. Genauso können Daten in Tabellen eingefügt, verändert und auch gelöscht werden.

C#:

C# hat die gleichen Grundeigenschaften und Grundfunktionen, wie andere (in diesem Studium) erlernten Programmiersprachen. In C# gibt es Klassen, Das bedeutet, dass man verschiedene Funktionen oder Teile des Codes in verschiedene Klassen einteilen

kann und sollte. Damit wird die Übersichtlichkeit gefördert und somit können andere Personen den Code leichter verstehen, sowie erleichtert es die Fehlersuche nach Bugs oder anderem unerwünschtem Verhalten.

Es gibt einiges erlerntes, deshalb wird hier eine Auswahl beschrieben:

- C# erlaubt bei Arrays statt mit „count-1“ die letzte Position zu bekommen, [^1] zu verwenden. Ist sehr viel kürzer und erhöht die Übersichtlichkeit.
- Visual Studio bietet die Möglichkeit dem aktuellen Projekt ein weiteres hinzuzufügen. Somit ist es beispielsweise möglich einfach Unit-Tests für das Projekt zu erstellen, trotzdem noch sichtbar und strukturell die Unit-Tests getrennt vom Hauptprogramm zu haben
- Es gibt in C# den Datentyp „Bool“, welcher TRUE oder FALSE sein kann, welchen es nicht immer in anderen Programmiersprachen auch gibt. Somit muss dieser Typ gegebenenfalls mittels „Integer“ simuliert werden. In C# gibt es „Bool“ jedoch voll integriert, sodass dieser auch einfach in Vergleichen (wie in „if“ Statements) verwendet werden kann.
- Es ist leicht möglich Threads zu erstellen und somit mehreren Code „gleichzeitig“ ausführen zu lassen

## 4. Unit Testing Entscheidungen

## 5. Unique Feature

Als unique Feature wurde ein Chatroom gewählt.

Es kann jederzeit mit „GET“ eine Anfrage an: localhost:10001/chatroom mit dem Sicherheitstoken gesendet werden. Dabei bekommt man alle Chatnachrichten retour. Es wird pro Zeile eine Chatnachricht angezeigt und die Chatnachrichten sind für eine bessere Lesbarkeit und Übersichtlichkeit durchnummeriert.

Jede Chatnachricht enthält folgende Informationen:

- Der Username, von welchem Account die Nachricht gesendet wurde
- Die Nachricht / den Nachrichtentext
- Der Zeitpunkt (Datum, Uhrzeit (dd.mm.yyyy hh:mm:ss)), zu welchem die Nachricht vom Server empfangen wurde  
Es wird der Zeitpunkt vom Server genommen, da sonst der User eine falsche / manipulierte Zeit mitschicken könnte

Genauso kann jederzeit mit „POST“ an den gleichen Pfad eine Nachricht gesendet werden, dabei ist genauso der Sicherheitstoken nötig.

Die Nachricht selbst wird im JSON-Format mitgesendet

Struktur des Bodys der Anfrage:

```
{"Username":"beispielusername", "MessageText":"Beispiel Text der Nachricht"}
```

Struktur der Datenbank:

- id: primary key, INT, Autoincrement der Datenbank, NOT NULL
- username: VARCHAR(50) , NOT NULL
- message: VARCHAR(510) , NOT NULL
- messagetime: timestamp without time zone, NOT NULL

## 6. Tracked Time