



# Monster Trading Card Game

## Hauptkomponenten

Komponente	Beschreibung
Program.cs	Initialisiert den Server (`HttpSvr`), lädt die Konfiguration (`appsettings.json`) und startet den `SessionHandler` und den `LobbyRepository`.
SessionHandler.cs, UserHandler.cs	Verarbeiten HTTP-Anfragen für Benutzerregistrierung, Login, Sitzungsverwaltung, Kartenerwerb, Deckmanagement und Kämpfe.
UserRepository.cs	Verwaltung der Benutzerkonten.
CardRepository.cs	Verwaltung der Karten und deren Eigenschaften.
PackageRepository.cs	Erstellung und Verwaltung von Kartenpaketen.
UserStackRepository.cs	Verwaltung der Spielerkartensammlung.
UserDeckRepository.cs	Verwaltung des Spielerdecks.
BattleRepository.cs	Implementierung der Kampflogik.
TradeRepository.cs	Verwaltung des Kartenhandels.
Battle.cs, Round.cs	Implementieren das rundenbasierte Kampfsystem mit Elementar-Effekten.
CreateTables.cs	Erstellt notwendige Tabellen für Benutzer, Karten, Decks, Kämpfe und die Lobby.

# Monster Trading Card Game

## 2. Implementierte Features

### 2.1. Benutzerverwaltung

- Registrierung und Login
- Token-basierte Authentifizierung
- Logout-Mechanismus

### 2.2. Kartenverwaltung

- Kartenmodell mit Name, Typ, Schaden, Elementtyp
- Methode zur Schadensberechnung basierend auf Elementtyp-Effektivität

### 2.3. Deckverwaltung

- Spieler können 4 Karten in ihr Deck verschieben
- Decks werden für Kämpfe verwendet

### 2.4. Kampfmechanismus

- Rundenbasiertes Kampfsystem
- Spezialregeln für bestimmte Karten
- Automatische Beendigung nach 150 Runden oder 15 aufeinanderfolgenden Unentschieden

### 2.5. Trading-System

- Nicht erfüllt

### 2.6. Scoreboard & Benutzerstatistiken

- ELO-System zur Bewertung der Spielerleistung
- Speichert Anzahl der Kämpfe, Siege, Niederlagen

## 3. Herausforderungen & Lösungsansätze

Herausforderung	Lösung
Implementierung eines eigenen HTTP-Servers	Verarbeitung von HTTP-Requests mit Statuscodes (200, 400, 401, 500)
Token-basierte Authentifizierung	Verwendung eines SessionHandlers zur Speicherung aktiver Token
Effizientes Datenbankdesign	Normalisierte Tabellen für Users, Cards, Battles, Trades, Packages
Kampf-Logik mit Regeln	Implementierung von Spezialfällen wie Kraken immun gegen Zauber, Goblins haben Angst vor Drachen
Testabdeckung	Unit-Tests-MTCG (20+ Tests)

# Monster Trading Card Game

## 4. Zeiterfassung

Task	Zeitaufwand (Stunden)
Planung & Design	5
Implementierung der Benutzer-APIs	8
Implementierung der Kartenverwaltung	6
Entwicklung der Kampf-Logik	12
Datenbankmodellierung & Abfragen	7
Implementierung der Handelsfunktion	4
Tests & Fehlerbehebung	10
Gesamt	52 Stunden

## 5. Infos

**Die Abgabedateien befinden sich im Ordner "Abgabe" innerhalb des Projekts.**

Dieser Ordner enthält:

- Monster\_Trading\_Cards\_Backup.sql → Datenbank-Schema (Tabellen ohne Daten)
- MTCG\_Protokoll\_Anthony.pdf → Projektbeschreibung und technische Details
- ClassDiagram1.png → Klassendiagramm des Systems

## 6. Unique Feature

Mein Unique Feature ist, dass jeder User ein „Motto“ hat. Jedes Mal, wenn ein User ein Battle gewinnt, wird sein Motto gepinnt, um den Verlierer zu provozieren.

## 7. Repository & Codebase

🔴 Das vollständige Projekt ist auf GitHub verfügbar:

[https://github.com/Anthonykey/Swen1\\_Monster\\_Trading\\_Cards\\_Game](https://github.com/Anthonykey/Swen1_Monster_Trading_Cards_Game)

!