Verwendete Packages:  
System.Data.SqlClient (4.8.3)

Newtonsoft Json

Github: https://github.com/if20b099/MTCG

Curl Script

Das curl script wurde heavily edited um die umgeschriebenen Funktionen zu matchen. Einige Scripts funktionieren nicht, da sie nicht implementiert wurden.

Design/Lessons learned

Ich habe versucht alles mehr oder weniger simpel zu halten. Das hat auch so anfangs funktioniert bis ich die Funktionalitäten aufgrund der Datenbank ein bisschen ändern musste (da array nicht so gut ist wie list). Später hat es sich auch schon herausgestellt, dass ich mehr einzelne Datenstrukturen (decks) als Klassen hätte machen sollen im Falle, dass nicht alle members sofort assigned werden können. Deswegen gibt es ein paar Überreste, die nicht schön im code aussehen und Fehler verursachen können. In diesem Sinne sollte ich öfter während dem Programmieren schon refactoren.

Lessons learned: Öfter in mitten der Projekte wieder designen, öfter refactoren

Unit Test Design

Die Unit Tests wurden designed einzelne Funktionen zu dahingehend zu testen, ob sie funktionieren, welche Fehler sie haben (da diese angezeigt werden) und größere Abfolgen auszuprobieren ohne immer das Program öffnen zu müssen. Deswegen sind die meisten Tests relativ klein gehalten worden, da meistens kleinere Units/Funktionen getestet wurden. Beispielsweise beim Registrieren wurden die einzelnen kleineren Funktionen wie das überprüfen des Namens und das einfügen von Stat records getrennt geprüft, aber auch zusammen mit der ganzen Registrierung, die beide beinhält.

Unique feature

Kritischer Schaden (2x) mit 20% Wahrscheinlichkeit im Battle.

Time spent

Da gibt’s mehr oder weniger nur ein rough estimate bei ca. 60-70 Stunden.

Gefühlt hat die DB am meisten gedauert, weil es sehr viele Funktionen gebraucht hat inklusive der prepared requirement.

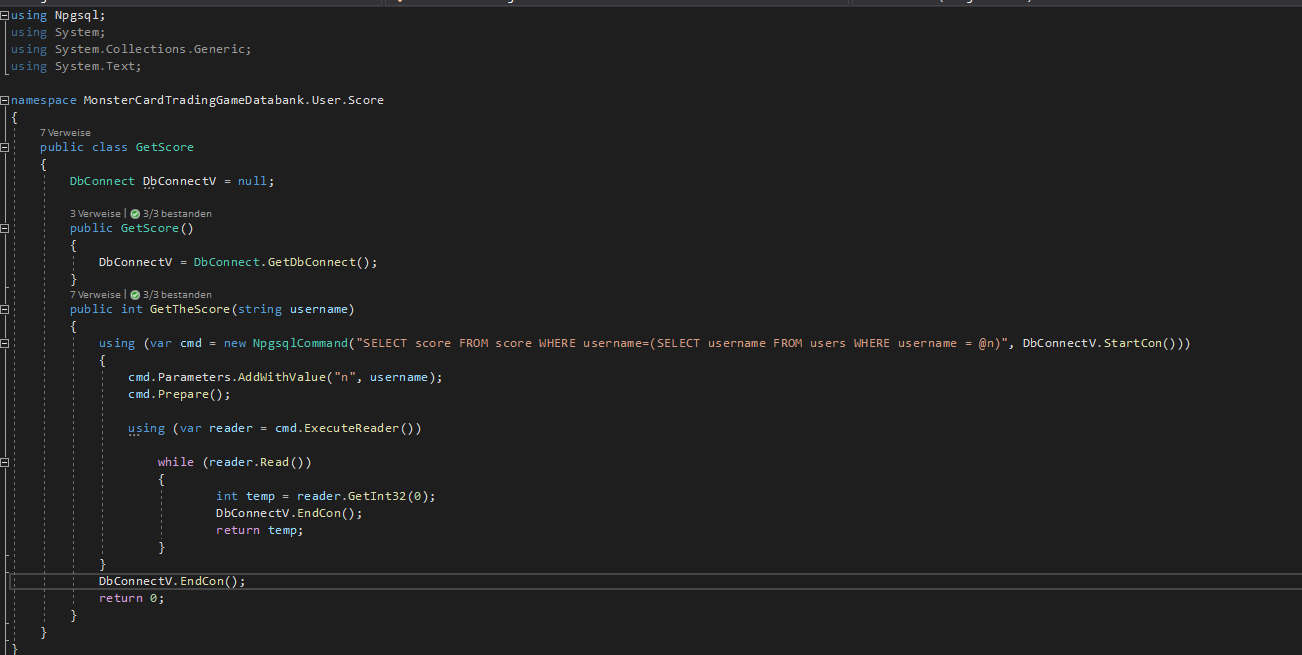
DB SQL Server VS PostgreSQL

Es gibt einige Unterschiede sowohl bei der Benutzung als auch bei der Installierung. Allerersten ist PostgreSQL portable während SQL Server auf einem PC auf einer Server Instanz laufen (in meinem Falle SQL Server 2016 EXPRESS). Zudem braucht man bei SQL Server das Management Studio was man separat downloaden muss. Nach einigen Einstellungen in SQL kann man auch auf die Datenbank zugreifen. In Postgre SQL muss die Datenbank mit SQL Statements erstellt werden und security dort verteilen. Beim SQL Server kann man das auch, hat aber die Option es „grafisch“ zu designen und kann dort auch direkt constraints einstellen.

Sehr wichtig beim Programmieren jedoch ist, dass SqlClient threadsafe ist, während bei postgres muteex verwendet werden muss.

Beim Programmieren selber gab es nicht viele Unterschiede. Beide haben ihr eigenes Package gebraucht. Auch die Verwendung von den Prepared statements ist sehr sehr ähnlich.

Beispiel einer Kollegin:



Meine SQL Server Version:

