

# Protokoll

<https://github.com/theeeeeon/Cardgameproject.git>

## Time tracked

C# lernen (syntax, neue features): 12 stunden

Entwickeln (nach curl script routen entwickeln, db verwalten, usw): 45 stunden

Unit Tests: 10 stunden

Documentation: 3 stunden

Fehler suchen, debugging: 9 stunden

## Unique Feature

Das Unique Feature ist eine Art Easter Egg. Wenn man einen Username in dem „420“ enthalten ist wählt, bekommt man eine spezielle Karte in seinen Stack. Am Ende der Db-init Funktion wird diese Karte in den Card table eingefügt und im Userendpoint (users) in den Stack gebracht.

## Unit testing

Die Datenbank bzw die Datenbankzugriffe werden nicht getestet, um zu vermeiden bei den Tests die db init Funktion aufrufen zu müssen. Getestet werden zum Teil die Models, ob die vorgegeben Werte den Vorgaben entsprechen (ELO, coins) oder das Package Model immer die richtige Anzahl an Karten hat. Außerdem wird getestet, ob es einen Error gibt, wenn man Routen mit falschen Methoden aufruft und gewisse Klassen den richtigen Typen entsprechen. Die Packages sollen nur vom admin erstellt werden, dies wird ebenfalls getestet.

## Lessons learned

Die Models, die Klassentypen oder generell die Struktur muss man oft laufend anpassen und deswegen ist es gut, wenn man sich relativ bald die komplexen Vorgänge überlegt und die Struktur danach an die Überlegungen anpasst, um ein unübersichtliches Programm zu vermeiden. Ein curl script oder ähnliches relativ früh

anschauen macht das Planen leichter, da man sich das Programm leichter vorstellen kann.

## Design

Das Diagramm ist in dem Projekt enthalten. Das Projekt ist aufgeteilt auf Models mit Card oder User (in dem es Deck usw gibt), Db (repository ibt es keine das passiert alles in der Business Logic, allerdings gibt es dort ein paar get Methoden die man in ein Repository verschieben könnte), Logic und Server mit Endpoints. Ein paar Entscheidungen sind aus Zeitgründen passiert, wie die volle Businesslogic ohne Repository oder die Models, die Teilweise angepasst werden müssen oder nicht benötigt werden. Der Request funktioniert mit Dictionary mit den Routen als key und den Methoden als Eintrag, damit man leicht neue Routen eintragen kann.