

MCTG-SWEN

Protocol

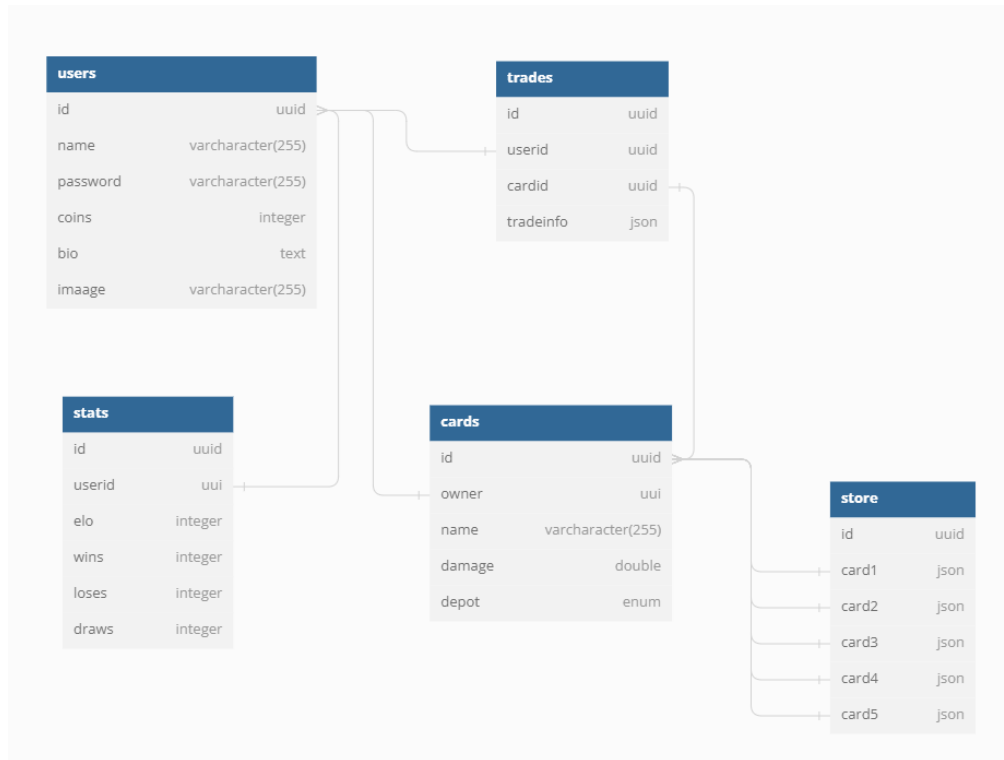
Brian Schneider (if21b072)

Table of Content

Design	3
Lesson learned	4
Unit test design	5
Mandatory Feature	6
Tracked time	7
Git-Repo link	9

Design

The database consists of five tables: "cards", "stats", "store", "trades", and "users".



The cards table store information about each card, including unique ID, owner ID, name, damage, and depot.

The stats table store statistics of the users, including unique ID, user ID, ELO, wins, loses and draws.

The store table store information of card that available for purchase.

The trades table store information about trades made by user, including unique ID, user ID and card ID.

And the users table store information of each user, including unique ID, name, password, coins, bio and image.

Lesson learned

Throughout the course of this project, I have gained a deeper understanding of the importance of proper planning and foresight in the development process. I now recognize the value of thoroughly considering potential issues and taking steps to mitigate them in advance, as it can save significant time and effort in the long run.

I have also realized the benefits of starting a project early, which allows for ample time to address any issues that may arise and stay ahead of schedule.

Furthermore, I have learned the importance of regular communication with a supervisor or mentor. Their valuable input and guidance has helped me gain new perspectives and insights that have improved my work.

Going forward, I plan to incorporate these lessons into my future projects to ensure their success.

Unit test design

I have begun the unit testing process and have considered which parts are the most important. Having previously gained experience with unit testing during my internship, I felt confident in identifying and writing unit tests. After careful consideration, I realized there were many areas to test as there are many methods, databases, etc. in the project. In order to keep within the scope, I decided to write 3-4 tests for almost every class, ensuring that the project is covered in a comprehensive manner and that my knowledge is demonstrated.

I have divided the unit tests into four areas: parsing, models, battle, and authentication. The parsing tests focus on the request container, which parses incoming requests. The model tests focus on testing the methods of the card and test models. The battle tests focus on checking the logger, logic, and battle functionality. The authentication tests focus on testing the authentication methods such as "isAdmin" and so on.

Testlauf abgeschlossen: 27 Tests (27 erfolgreich, 0 fehlerhaft, 0 übersprungen) in 520 ms ausgeführt			
Test	Dauer	Merkmale	Fehlermeldung
✓ MCTG_UnitTest (27)	182 ms		
✓ UnitTest.PARSE (4)	5 ms		
✓ RequestContainerTests (4)	5 ms		
✓ TestParseProtocol	< 1 ms		
✓ TestParsePath	< 1 ms		
✓ TestParseMethod	2 ms		
✓ TestParseHeaders	3 ms		
✓ UnitTest.MODEL (3)	4 ms		
✓ StatsTests (3)	4 ms		
✓ TestUpdateWinStats_Adds10To...	3 ms		
✓ TestUpdateLoseStats_Subtracts...	< 1 ms		
✓ TestUpdateDrawStats_Incremen...	1 ms		
✓ UnitTest.Model (7)	7 ms		
✓ CardTests (7)	7 ms		
✓ TestDetermineElementType_Na...	2 ms		
✓ TestDetermineElementType_Na...	< 1 ms		
✓ TestDetermineElementType_Na...	< 1 ms		
✓ TestDetermineElementType_Na...	< 1 ms		
✓ TestDetermineElementType_Na...	< 1 ms		
✓ TestDetermineCardType_Name...	< 1 ms		
✓ TestDetermineCardType_Name...	5 ms		
✓ UnitTest.BATTLE (8)	4 ms		
✓ BattleTests (1)	2 ms		
✓ TestCatchRndCard_DeckContai...	2 ms		
✓ BattleLogicTests (5)	2 ms		
✓ TestIsMonsterImmune_Card1Kr...	< 1 ms		
✓ TestIsMonsterControlled_Card1...	< 1 ms		
✓ TestIsMonsterAfraid_Card1Gobl...	< 1 ms		
✓ TestCalculate_Card1SpellCard2S...	< 1 ms		
✓ TestCalculate_Card1MonsterCar...	2 ms		
✓ BattleLoggerTests (2)	< 1 ms		
✓ TestClassifyUsers_SetsWinnerAn...	< 1 ms		
✓ TestAddToProtocol_AddsMessa...	< 1 ms		
✓ UnitTest.AUTH (5)	162 ms		
✓ AuthTests (5)	162 ms		
✓ TestUpdateUser	10 ms		
✓ TestLoginUser	< 1 ms		
✓ TestIsUserLoggedIn	< 1 ms		
✓ TestIsAdmin	1 ms		
✓ TestGetAll	151 ms		
Gruppenzusammenfassung			
MCTG_UnitTest			
Tests in Gruppe: 27			
🕒 Dauer gesamt: 182 ms			
Ergebnisse			
✓ 27 Bestanden			

Mandatory Feature

I have decided to create a fusion feature for my project. This method allows the user to select two cards from their stack and merge them into one stronger card. By calling the path "/fusion", the selected cards are passed to the method. The method then checks if the cards are actually in the user's stack and if the user has enough coins to perform the fusion.

If this is the case, the stronger card is determined, and its damage is increased while the damage of the weaker card is reduced. The user loses 10 coins in the process.

The fusion is successful if the method returns a message with a status code of 200 and the text "Fusion between two cards accepted".

Tracked time

commit 52ee10ac7bcc82676c04b508a60cefc320826882 (HEAD -> main, origin/main)		
<if21b072@technikum-wien.at>	Date: Mon Jan 23 15:28:21 2023	refactor path handle
commit 7fd19e65b7beba805c24519017481297e2c6593		
<if21b072@technikum-wien.at>	Mon Jan 23 12:42:31 2023	done hotfix for completion
Commit: eb5557e5fee28f8fd6d3dedf343c2ca6f1a6ad75		
<if21b072@technikum-wien.at>	Sun Jan 22 23:19:36 2023	done little refactor
Commit: 1e54c20a80947dfdbbc4f88ef39c4e38ad25694d		
<if21b072@technikum-wien.at>	Sun Jan 22 22:54:30 2023	Done minor changes for finishing
Commit: d71bc9d6cab147ebf2477fbd3bf444b87c165903		
<if21b072@technikum-wien.at>	Sun Jan 22 22:39:50 2023	Done some hotfix to complete project
Commit: b1c6450036fe01a50a6291f75af3a1bdf8061056		
<if21b072@technikum-wien.at>	Sun Jan 22 22:31:28 2023	Done some hotfix to complete project
Commit: 9e2666221bbc31969b01e489c813b613f43d2759 (refactor)		
<if21b072@technikum-wien.at>	Sun Jan 22 22:27:49 2023	Done some hotfix to complete project
Commit: 164426ed71ab730bfc2fb9a50ec114860e836707		
<if21b072@technikum-wien.at>	Sun Jan 22 01:42:00 2023	done some refactoring and testign
Commit: 4fcdd0665bcedd470ef89c9fbb662abcccca0a37		
<if21b072@technikum-wien.at>	Sat Jan 21 21:19:44 2023	add unit tests
Commit: 50874e43aeb4cb281318cd5a3e8c0732f2035f0b		
<if21b072@technikum-wien.at>	Sat Jan 21 00:19:43 2023	done hotfix
Commit: 343c138e42b9689ba48fbfbce29bc05de23235f7		
<if21b072@technikum-wien.at>	Fri Jan 20 19:52:12 2023	done more refactoring (struktur, db, etc.)
Commit: 4967e946e101120a65be11f702ff9c73ac93b605		
<if21b072@technikum-wien.at>	Fri Jan 20 02:13:22 2023	done big refactoring
Commit: b362d236210fbe158dfdb7915a288dbeda21e5ac		
<if21b072@technikum-wien.at>	Thu Jan 19 07:54:16 2023	Add Login/Session
Commit: a4a471a0f3ef0229e44b24f654e0f4194f8893e8		
<if21b072@technikum-wien.at>	Mon Jan 16 23:21:53 2023	Add BattleLobby and Battle
Commit: 349d7082989b32f6b1f6f6a4aa6b15d8eb1a395a		
<if21b072@technikum-wien.at>	Sun Jan 15 02:49:33 2023	Add many features
Commit: e77ee21243b55296c9a0c00c502cf7c77e449751		

<if21b072@technikum-wien.at>	Wed Jan 4 23:08:42 2023	changed repo and database structur/Add RestController
Commit: fe75532384d3636d3452c50010529e640aa67a3b		
<if21b072@technikum-wien.at>	Wed Dec 28 18:53:56 2022	Done minor changes
Commit: 93cfdc9cf1c1d0fffb020b7af603ac3e949b6ba6		
<if21b072@technikum-wien.at>	Wed Dec 28 18:21:12 2022	Add userService (add, upd, del, get, getAll) to interact with DB
<if21b072@technikum-wien.at>	Wed Dec 28 15:59:22 2022	Add userService to db w/ repo-pattern
Commit: d95e334430f182c4eb42a21ed8d4f08a1025024e		
<if21b072@technikum-wien.at>	Wed Dec 28 00:13:48 2022	Add first DBconnection
Commit: 560d49777100bcc5c727b0715340fde57255507a		
<if21b072@technikum-wien.at>	Tue Dec 27 21:44:09 2022	first commit

Git-Repo link

The following link contains the complete project:

>> https://github.com/lanaklanA/MCTG_Brian <<