Marvin Löhlein – if23b002
Github: https://github.com/Methuzalem/MTCGNew

App-Design

This app is based on a layered architecture in which each layer plays a clearly defined role.

Firstly, the HTTP request is forwarded to the application. Depending on which path is specified, the application routes the request to the corresponding controller - for example, the path '/users' is passed to the UserController.

The various HTTP methods such as GET, POST, PUT and DELETE are processed within the controller. For example, a GET request to '/users' is received in the UserController and then forwarded to the relevant methods in the UserService.

The services are responsible for implementing the logic and communicating with the repositories. A service can address several repositories, depending on which database tables are required. The service layer ensures that the requirements placed on the various paths of the REQUEST are implemented correctly.

The repositories take care of setting up the database connections and filling the corresponding tables. Each table has its own repository that manages all requests to this table centrally. The repositories work with prepared statements so that SQL-injections are not possible.

Lessons learned

In the course of this project, I learnt how essential a well thought-out structure of the methods and their return values is, if you want to implement such an extensive project. A standardized approach makes it much easier to reuse the methods in later functionalities and should always be considered.

DTO objects were new to me at first, and I hope that I have implemented them right in my project. The reduction of an object with numerous properties to the essentials seemed logical to me in principle, although I did not always notice a noticeable simplification in practical application, but rather an additional effort.

The in-depth study of unit tests gave me valuable insights. I realized how important it is to validate almost every method with appropriate tests - a practice that I can now understand much better.

In addition, working with IntelliJ was a completely new and positive experience for me. The IDE supported me in many ways: Be it executing curl scripts, writing and running the unit tests or managing the database - working with IntelliJ was consistently pleasant and efficient. Overall, this was the best experience I've had with a development environment.

Marvin Löhlein – if23b002
Github: https://github.com/Methuzalem/MTCGNew

Describes unit testing decisions

As part of my unit tests, I initially focussed on the services, as these represent the core of the logic. At first, it was challenging to get started with testing as I was initially unsure how to begin. Over time, however, I gained confidence and made sure to test every method in the services at least once.

I also decided to implement unit tests for the controller, as I added additional methods there to embellish the console output. Especially for tasks that have to do with formatting, it was important to me to ensure functionality and consistency through testing.

In the case of the BattleService, I could have written even more tests. However, as the task focussed on a variety of test cases, I concentrated on mapping different scenarios instead of repeating similar tests.

Describes unique feature

I would like to show three approaches for the Unique Feature:

Extended Card Enums:

I have extended the Card Enums so that, in addition to the normal Dragon or Wizzard cards without an element type, variants such as FireDragons, WaterWizzards etc. can also be created. This means that the element type can be used flexibly for each monster - in principle, it would even be possible to generate such cards via the curl script.

Free 10th pack in the shop:

Another feature is that every tenth pack purchased in the shop is free. This bonus scheme is intended to give something back to players and reward them for participating in the game.

Dynamic battle feature:

I have noticed that the games often last up to 100 rounds as the cards are very balanced and change players regularly. To create more excitement from round 50 onwards, I have implemented a buff:

- For one player, all spell cards in the current deck are boosted by +10 damage.

- For the opponent, all monster cards in the deck receive a +25 damage bonus, as monsters tend to be slightly weaker than spell cards.

It is chosen completely random who gets which buff. This should contribute to fairness among the players. This adjustment ensures that games more often end before reaching 100 turns, which is also reflected in the Elo development in the curl file.

These features show my approach to integrating innovative and player-friendly ideas into this gaming project.

Marvin Löhlein – if23b002
Github: https://github.com/Methuzalem/MTCGNew

Tracked Time for the final Hand-in

21.02.2025	*24.02.2025*	*25.02.2025*	*27.02.2025*	*28.02.2025*	*01.03.2025*	*02.03.2025*	*03.03.2025*	*05.03.2025*	*06.03.2025*
4 hours Reading and understanding Intellij, the different builds and the folder/package structure	**2 hours** Learning about postgreSQL	**2 hours** Learning about threading	**4 hours** Create new project with right build and database connection	**4 hours** Implement users, card logic and repo	**8 hours** Implement decks, transactions and packages	**8 hours** Implement user update finished Decks	**10 hours** Implement ScoreBoard and Stats working on threads	**10 hours** Implement Battle logic started with tests, implement Tradings, finished threads	**8 hours** Finished Tests, reworked code, implement unique feature, working on protocol

Overall round about **60 hours** which I invested in this final hand-in for the project.