

Honours Project Report

Fitness Logger with Micro-services

Dumitru Vulpe
BSc (Hons) Applied Computing

May 2020

Abstract

This project is a tool to let people be able to log and track workouts overtime easily from their phone in a flexible yet complete way. The initial purpose was to make a phone application which can be used across different disciplines of sport and with different workout types. This would be done by breaking up the data in different manageable units so that the user can create their own workflow for logging workouts.

However, this was not the main appeal for choosing this project. The main appeal of this project was the backend implementation. This is because as a part of the architecture design process, it was decided that a micro-services structure would be followed. This was chosen for a multitude of reasons, including as a learning experience for this kind of backend architecture.

1 Introduction

This project is split into two main objectives; first objective was to be a learning experience of creating a full stack application beginning to end and more importantly to get into micro services backend development. The second objective was to attempt to solve a problem which exists in the fitness industry by creating an application which would be useful to a different range of people and potentially taking this product to the market.

1.1 Why micro services

The basic principle of micro services as supposed to a monolith, is to split up the backend application into multiple smaller application. For example, in theory, a service such as Amazon, could have a separate micro ser-

vice for the cart function, one for listings, one more user account management, one for payment and more. Micro services architectures are becoming more and more relevant simply because the industry is getting becoming more and more complex. And as size, complexity and user bases increase, so does the need for well scaled web applications. Micro services type architectures are simply one way to cleanly and easily scale web applications. So because of industry relevance and my personal enthusiasm, it was decided that this project's backend would also follow this architecture.

1.2 Creating a potential product

1.2.1 The problem

From personal experience and research, there isn't a simple, yet flexible application for logging workout sessions and related notes for said session. The applications which I have tried so far are either made for a specific workout programs or are not complete, buggy or just do not have enough flexibility in the type of logging you can achieve.

1.2.2 My planned solution

The idea was that the system would be designed around the data structure which would be saved by the user. This data structure would be made as simple as possible while still allowing all the common basic exercise options (such as repetitions, resistance values, sets etc) but also others (such as the option to just write a single value). Furthermore, the system would allow the user to create their own exercise types which they would then use when logging a workout.

2 Background

2.1 Architecture design

A lot of time during this project was allocated to research for the backend architecture design, specifically micro services. When it comes to the world of micro services, there does not exist just one generally accepted standard that people adopt, and there are a lot of decisions to make. For instance the questions of how the different services should be split, some say it should be split by the feature, some say by the individual data element. Another popular questions is how the services diagram should look like, one of the popular answers are to have a separate authentication service to provide authentication for the application and proxy the requests to all the other services. However, regardless of the question, there are a lot of different ways solutions to all of these questions that need to be answered when starting a project using micro services and all of them come with their own advantages and disadvantages.

2.1.1 Helpful technologies

When creating micro services systems, there are a lot of technologies which could and would be used to make the development of these services easier. Probably one of the first pieces of technologies to ease the development and deployment of micro services a lot easier was containers. Containers allow you to package applications with all of the needed dependencies and environments, they virtualise the user-land applications and share the hosts kernel, and on a Linux system that means that containers can virtually run anywhere. Furthermore, we have also have container management and orchestration solutions such as Docker[1], docker-compose[1], Kubernetes[2] and more. These help with running the actual containers, and in the case of Kubernetes, it also allows us to easily define configurations and scale the containers either on a definition or on demand. Scaling here would be done by running more instances of that container across a cluster of machines running the Kubernetes and load balancing the traffic amongst them.

This is known as horizontal scaling.

2.2 Market

3 Specification

4 Design

5 Implementation and Testing

6 Evaluation

6.1 Usability

6.2 Other Criteria

7 Description of the Final Product

8 Conclusion

Acknowledgements

References

- [1] Docker (Software) Wikipedia (2021).
[https://en.wikipedia.org/wiki/Docker_\(software\)](https://en.wikipedia.org/wiki/Docker_(software))
- [2] Kubernetes website (2021).
<https://kubernetes.io/>

Appendices