S-ESE6
Portfolio Template

<Mureseanu Gabriel>

Versioning

Version	Description	Date
V0.1	First Version	11/3/2022
V0.2	Second Sprint	10/4/2022
V0.3	Fontys Career Day + Links	14/4/2022
V1.0	Third Group Project Sprint – Third + Fourth Individual Sprint	25/5/2022

Contents

Introduction

The group project is a back-end implementation of an application that helps people manage their stress. The back-end has to be designed as an enterprise system, therefore enterprise architecture has to be used in the making of the product. For this specific product we have chosen to go with microservices communication through an event bus.

My individual project consists of a YouTube inspired web application that will make use of microservices to function. This application aims to improve the chances of content creator to be discovered and become popular based on content quality. The algorithms will use likes, dislikes and a new rating called quality together with the video length in order to see how well a video is doing. On the contrary, it will also be used to identify lazy content and punish long-lazy content like reaction videos.

I have a little bit of knowledge when it comes to microservices, as I have had a similar project back in Semester 3, however it was not based on Enterprise Software Design, which makes this way harder than before.

As of my interests, I chose this project because I used to run a YouTube channel that was terminated a few years ago (2018), and in the experience that I had, YouTube is not creator friendly. I wanted to create a similar experience, with some extra features sprinkled in, where the creators can feel safer than on YouTube.

This semester I am looking to learn how big enterprise software is made, and I would like to achieve a website that can truly be used by millions of people.

Learning outcomes

Indicate where you think you are on the development scale, based on the feedback from your teachers.

Describe to the reader for each learning outcome what you have achieved during the past sprint and why this contributes to the learning outcome. Substantiate the why with feedback from your technical tutors.

The portfolio grows with content; sometimes certain content will no longer be relevant. Describe each sprint from the current status of your portfolio. You use hyperlinks to refer the reader directly to material in your portfolio.

(Reflection on progress) indicate where you are now and what your tutors have given as feedback to grow further on the development scale.

1. Enterprise software development as a team effort.

You develop and deploy enterprise software, both individually and as a team. You select a suitable enterprise development platform and application framework(s). You select and apply a software development process, which complies with professional industry standards. You actively share knowledge within your team and with stakeholders to improve knowledge & processes.

Development (undefined, orienting, beginning, proficient, advanced)

ID	Description	Type	Level
1.1	Sprint 0	Group project	orienting
1.2	Sprint 1	Group Project	beginning
1.3	Sprint 2	Group Project	beginning
1.4	Sprint 3	Group Project	proficient

Substantiation

1.1: Sprint 0

In this sprint we started by getting the requirements from the PO and creating the first concept of the application. We asked questions about everything, data, target audience, UX decisions, etc. and got mostly positive feedback on everything we have shown.

We also created wireframes and presented the proposed architecture of the project, to which we got the green light to.

We also decided on the technologies that should be used for the application, which are React for front-end and C# for back-end.

1.2: Sprint 1

This sprint we have set up more of the DevOps cycle by moving to Jira and reviewing each other's code on pull requests. We have also created a unified backlog with the other groups.

Link to Jira: https://bstoykov.atlassian.net/jira/software/projects/SWSP2/boards/1

We now have 3 stand ups a week, where we discuss what we did and what needs to be done. These stand-ups take place Tuesday, Thursday, and Friday at 10:00.

We have also started communicating more between the 3 groups.

1.3 Sprint 2

This sprint we continued working as a group, having a fully set up DevOps. However, the Jira board provided by Fontys got locked due to unknown reasons, so we moved the tasks and backlog to GitHub.

We “professionalized” the way of working and polished it up, however the Jira task board being locked has been a major setback in our DevOps

1.4 Sprint 3

During this sprint I think we’ve had the most professional way of working so far, we held 3 stand ups per week and had the most amount of contact with the other groups that we’ve had so far, not only that, but since we’re close to the end of the project, and there are still issues and bugs with the software, we have recently started helping each other much more than in the previous sprints. We have also split up the tasks for Sprint 4, for which I got the K8S implementation and Azure Deployment together with Denny

Reflection on my progress

1:1: Sprint 0

I have participated in every group activity so far, giving my input and voicing my opinion when it comes to choices. I also gave a presentation to the class about NATS, a potential Event Bus server. I would say this is a great start to the project. This learning outcome is going to slowly be achieved over the course of the semester by just working as a group.

1.2: Sprint 1

We have started working in a kind-of professional way, which reminds me of the way of working I experienced while having my internship at Philips last semester. I am sure that once we get used to it and fine tune the details, we will have a fully professional way of working as a team.

1.3 Sprint 2 & 1.4 Sprint 3

I think that we have reached the best team set-up we can get without having a dedicated scrum master, since it is hard to orchestrate so many tasks without a person doing this specifically. In terms of our set-up, we are close to what I experience at my current job.

2. Conducting context-based research

You deliver professional products according to planning, which are the result of a structured and methodical investigation. You have a critical view towards your own and other people's work, by comparing them to alternatives, judge the structured and methodical approach and consider general accepted and ethical values. Your products are validated with stakeholders and other available research, and you can judge & communicate the relevance and value of the project in its own context.

Development (undefined, orienting, beginning, proficient, advanced)

ID	Sprint	Type	Level
1.1	Sprint 0	Group project	orienting
1.2	Sprint 1	Individual project	orienting
1.3	Sprint 1	Group Project	orienting
1.4	Sprint 2	Individual project	beginning
1.5	Sprint 3	Individual project	beginning
1.6	Sprint 3	Group Project	proficient

Substantiation

1.1: Sprint 0

In order to gather the requirements, we had to come up with a list of questions, which we did.
After coming up with the questions we asked the PO with the hope of finding answers. Some questions could not be answered because they were on the more technical side, so instead we directed them to Hank.

We also had to question the technologies we had to use, in the form of an "investigation". We can now validate the value of the choices that we made when it comes to the technologies using a list of pros and cons.
Every choice that we made so far can be justified that it is the best decision for the development of the application.

1.2: Sprint 1

I have conducted research on how other enterprise software are designed (available product analysis) , especially YouTube and Twitter, since they contain similar elements to my project. I have created a list of user stories and use cases in order to create requirements for the project.

1.3: Sprint 1

This sprint we mainly did research on how to handle communication between services, we concluded that an event-bus was not the perfect way to handle all the communication, therefore we decided to also implement an API together with a gateway.

The main point of reference to the research was the official Microsoft Enterprise Microservice Based Applications document, specifically the part about communication which can be found at :https://docs.microsoft.com/en-us/dotnet/architecture/microservices/architect-microservice-container-applications/communication-in-microservice-architecture

1.4 Sprint 2

This sprint I have done a lot of research into the following topics: Threads, Three.js, Best practices for C# based microservices and data encryption.

The following sources were used for research:

Threads:

- **https://www.c-sharpcorner.com/article/Threads-in-CSharp/**
- **https://stackoverflow.com/questions/6126616/is-dbcontext-thread-safe**
- **https://docs.microsoft.com/en-us/ef/core/dbcontext-configuration/**

Three.js

- **https://threejs.org/docs/index.html#manual/en/introduction/Creating-a-scene**
- **https://www.youtube.com/watch?v=pUgWfqWZWmM**

C# Microservices

- **https://www.youtube.com/watch?v=DgVjEo3OGBI** (Despite being an 11 hour course, it was extremely fun to watch and follow along)
- **https://dzone.com/articles/7-microservices-best-practices-for-developers**

A document with the research of Three.js will be presented in the sprint delivery.

The reason I did research on Threads is because I am currently trying to pass many copies of a DatabaseContext to different threads. I have also discussed this with Jacco and he gave me a basic idea where to start. I have reached the conclusion that passing a reference to the DBContext instead of the DBContext itself is the correct way to do it when using multiple threads.

I have not started writing the research document; however I am actively looking for information sources.

1.5 Sprint 3

During this sprint have done a lot of research on how to achieve Video Streaming like YouTube does, and after 2 long weeks of experimenting and reading articles, I finally managed to do it, and now I will walk through the steps which I took in order to achieve it.

First, I had to know what streaming refers to exactly, so after some searching around, the best explanation comes from Cloudflare, which can be found at this link.

But even if I knew what the definition of streaming is, I had to know how my “competitors” do it, so I started looking into the way YouTube does it.

I found many articles and videos explaining how YouTube streams videos, but the best explanation that I found came from a YouTube creator by the name of Hussein Nasser and his video, titled : **How YouTube Efficiently Streams Videos through HTTP? - DevTooling YouTube**

Now, knowing what streaming is and how YouTube does it, I had to find a way to do it for myself. When asking one of my friends, who is also a student in the 6th semester at Fontys, he linked me a video called **Creating a video streaming with Node.js** by Gibolder Web-Dev.
While this video pushed me in the right direction, it was not in the language I intended to use, so I had to experiment in C#.
After a lot of prototyping, I created the version that is used for the website, which sends a video in chunks of 4 MBs.

1.6: Sprint 3
During this group sprint, we have done a lot of research on how the stress data should be handled in order to optimize space usage, transfer time and maximize information sent. We have discussed with the other groups about these problems and all of us have researched the best solutions. While some solutions were contradictory, at the end of the sprint we have reached a very solid template for the data, which lowered the amount of disk space used.

At the same time as this, we also completed a few sub-questions of the group research document, to which I have contributed to 2 out of the 6 sub-questions. These will be delivered in week 16.

Reflection on progress
1.1: Sprint 0
In order to come up with the list of question, we had a group meeting where everyone discussed what we have to ask. After the questions were asked and we got a few answers to them, I started working on the pros and cons list for the technologies. I provided my group with a list of possible technologies together with the pro and cons list and then we decided together on which one to use
1.2: Sprint 1
I got some feedback from the technical teachers regarding the documents, which I applied shortly after the meeting. I am still not sure how this learning outcome will be completed, as I do not really understand exactly what the criteria is. I will need to contact the teachers to gather more information.
1.3: Sprint 1 & 1.4 Sprint 2
The past sprint has been full of research in order to achieve the best result of the software, both for group and individual project.
We still have not played the ethics game, but this will be done in the first week of next sprint.
I think I am a lot close to reaching the learning outcomes than last sprint due to the amount of context-based research that has been done in order to deliver the best possible solution.
1.5: Sprint 3 & 1.6: Sprint 3
A lot of research had to go into Video Streaming, since it is quite a “closely guarded” secret, and not many developers have tried to do anything similar in the past. While it was a good experience to try to make something that is not that well known, I would not put so much time into research for a singular feature if I would ever be in a similar situation.

The group research on the other hand has been way easier, since the questions we chose have a lot of research already done into them by big companies and other developers, so we do not have to reinvent the wheel.

I think that for this semester the only thing needed to reach this learning outcome is to deliver the individual research.

3. Preparing for lifelong learning

You acquire skills required for your future career. You are aware of multiple career paths and can reflect which ones fit best, considering your (potential) skills and ambitions. You are aware of developments in software engineering and can signal trends.

Development (undefined, orienting, beginning, proficient, advanced)

ID	Sprint	Type	Level
1.1	Sprint 0	Group project	orienting
1.2	Sprint 1	Individual project	orienting
1.3	Sprint 2	Individual project	Beginning
1.4	Fontys Career Day	Both	Beginning
1.5	Sprint 4	Individual	Proficient

Substantiation

1.1: Sprint 0

We are currently using one of the newest and hottest technologies when it comes to services – microservices. Not only that we are using constantly evolving tools with long term support such as NATS and .NET Core.

1.2: Sprint 1

I think that Enterprise Design is the software design of the future, since the internet is seeing more and more traffic each day, and this requires a solution – Enterprise Design.

I am not sure about my career in the future, however, having knowledge in such an important aspect of Software Engineering is for sure going to help me with it.

1.2 Sprint 2

This sprint I have made sure to research the industry when it comes to standards, I have researched the correct way of handling Microservices in C#. Since this technology is still rapidly evolving, information can change at any time.

I have also researched how scalability is done in the industry, and I have chosen the way that suits my project the best.

Fontys Career Day:

On 13/04/2022 I have participated in the Fontys career day. I have discussed with Michael from game design about the specialization in game design and I found out that he wants to try to actually create a company during the specialization. I went from stand to stand and talked with the people at the companies that were not too busy, but none of them caught my eye. When I got to the Philips stand, some of the members were my colleagues from my internship, I started talking to them and shortly after I was telling other students about how my internship went and a few of them got interested. Free marketing, I guess?

1.5: Sprint 3:

Due to the knowledge I gained this semester in K8S and Enterprise software, I managed to land a better job than my previous one at ProDrive Technologies, working with Rust and K8S.

Reflection on progress

1.1: Sprint 0 & 1.2: Sprint 1

I do not get why this is a learning outcome, why do I need to assess my skills and ambitions in order to search for a specific career when the Software field is one of the most volatile ones? It just does not make any sense to prepare for a specific career as a Software Engineer when everything software-related can be learned in such a short amount of time when compared to other career choices and the flavor of the month technology can change in the blink of an eye (i.e.: Monolithic architecture becoming obsolete when it comes to high traffic).

4. Scalable architectures

Besides functionality, you develop the architecture of enterprise software based on quality attributes. You especially consider attributes most relevant to enterprise contexts with high volume data and events. You design your architecture with future adaptation in mind. Your development environment supports this by being able to independently deploy and monitor the running parts of your application.

Development (undefined, orienting, beginning, proficient, advanced)

ID	Sprint	Type	Level
1.1	Sprint 0	Group project	orienting
1.2	Sprint 1	Individual	orienting
1.3	Sprint 2	Individual project	Beginning
1.4	Sprint 3 & Sprint 4	Individual Project	Proficient
1.5	Sprint 3	Group Project	Proficient

Substantiation

1.1: Sprint 0

We have designed a microservice architecture which accommodates infinite scalability of the microservices.

We are also using Docker and Kubernetes in order to scale, which are two tools specifically made for this.

1.2: Sprint 1

I am also designing an infinitely scalable architecture with microservice, and I am also using docker and Kubernetes in my individual project

1.3 Sprint 2

During this sprint I have researched ways to scale architecture, and following the industry standard, I have Dockerized all my services. I then used and application called MiniKube together with Kubernetes in order to horizontally scale my application. Due to the architecture that I have and NATS, horizontal scalability can be done automatically without changing anything. The most tested as of now, the application works with 30 copies of each service, and the load balancer works as intended.

1.4: Sprint 3 & Sprint 4

During these sprints, a lot of work has been done on the scalability of the project using K8S and Pod Autoscaling.

In order to automatically scale the microservices, I am using them as pods in K8S. This alongside other data gathering methods, such as Ingress-Nginx (bandwidth) and metrics-server (CPU and Memory) offer multiple valid metrics in order to scale. While the CPU and Memory are not valid metrics in a locally hosted K8S Cluster, they are valuable in case the cluster will be put up on the cloud, as each K8S pod is hosted given access to unique CPU cores and Memory clusters.

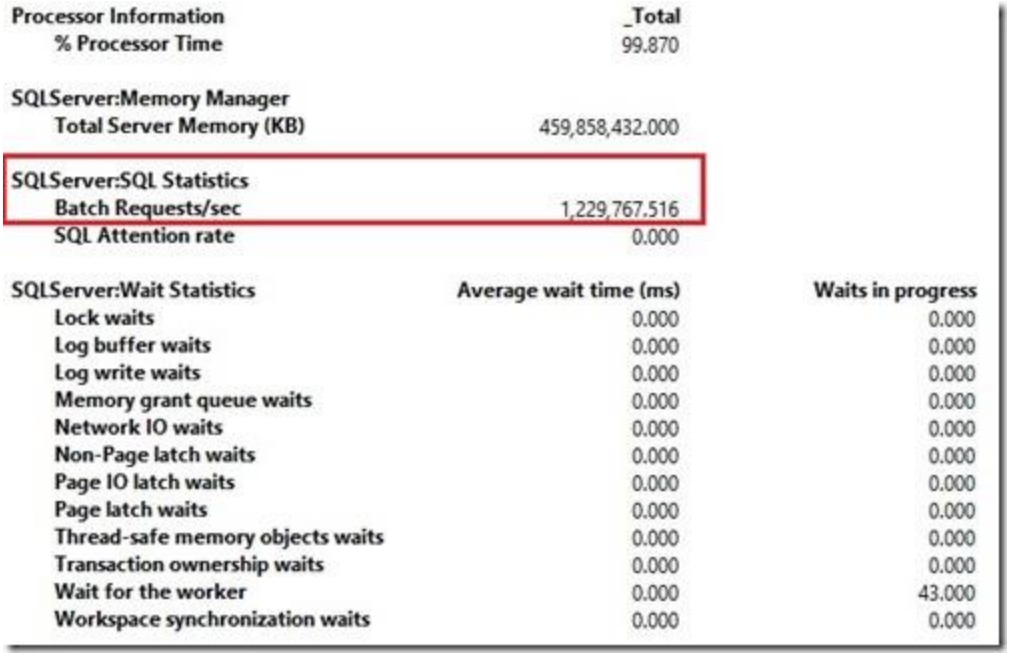
The autoscaling itself is done using the Horizontal Pod Autoscaler provided by Kubernetes, which takes the 3 aforementioned metrics and scales the pods according to the logic provided. In my case, a new pod will be created if one of the following conditions is reached:

- CPU core usage is above 80% for more than 30 seconds.
- A bandwidth of more than 200MB (2GB in case of the video streaming service) is used for more than 10 seconds.
- Memory usage leads to a bigger than 5000ms delay in the response of the pod.

In case any of the aforementioned conditions are reached, a new pod will be created, and Ingress-Nginx will use its Load Balancing features in order to lower the load on the pod that triggered the condition.

Ingress-Nginx also provides information regarding the usage of the microservices, such as unique users, calls per user, etc., but these are currently not used for anything.

In the case of Database scaling, I am currently not doing it, as I am using SQLExpress, which is one of the fastest databases available, however it cannot be scaled. It can handle as much as 200.000 requests per second when hosted on a below-average computer, not only that but it can support up to 37.000 connections.
When hosted on a powerful machine and optimized, it can handle up to 1.2 million requests as proven by this image provided by BWIN. More about how they did it can be found in this Microsoft article.



Processor Information	**_Total**	
% Processor Time	99.870	
SQLServer:Memory Manager		
Total Server Memory (KB)	459,858,432.000	
SQLServer:SQL Statistics		
Batch Requests/sec	1,229,767.516	
SQL Attention rate	0.000	
SQLServer:Wait Statistics	**Average wait time (ms)**	**Waits in progress**
Lock waits	0.000	0.000
Log buffer waits	0.000	0.000
Log write waits	0.000	0.000
Memory grant queue waits	0.000	0.000
Network IO waits	0.000	0.000
Non-Page latch waits	0.000	0.000
Page IO latch waits	0.000	0.000
Page latch waits	0.000	0.000
Thread-safe memory objects waits	0.000	0.000
Transaction ownership waits	0.000	0.000
Wait for the worker	0.000	43.000
Workspace synchronization waits	0.000	0.000

BWIN showcasing their powerful and polished SQL server technology

In the case of switching to a scalable database, I would use the cloud services of Amazon, as they offer custom scalable SQL servers for cheap, and because the connection is made using Entity Framework, the only change needed would be the connection string.

The only problem left is the Video storage, as it requires a lot of storage volume, so other than throwing money at it like YouTube does, I do not think I can fix this problem as a single developer.

1.5: Sprint 3

The same things that were mentioned for the individual project have been implemented in the group project as well. Not only that but we are also deploying the group project on Azure.

Reflection on progress

1.1: Sprint 0 & 1.2: Sprint 1

So far only the architecture plan has been created and no prototype is available.

The learning outcome will be reached once I can showcase the scalability in action.

1.3: Sprint 2

In this sprint I have learned a lot about scalability, from the different kinds of scalability, to how they can be implemented. I have chosen to go with a scalable architecture.

I feel that this amount of progress is getting me much closer to achieving this learning outcome

1.4: Sprint 3 &Sprint 4 & 1.5: Sprint 3

Personally, I think that I have reached the learning outcome, the scalability is complete based on 3 metrics, and the only difference in an actual deployed app would be the scaling logic, which can easily be changed. The architecture that I have made has allowed me to easily implement scalability in my application apart from video storage.

5. Development & Operations (DevOps)

You set up environments and tools which support your chosen software development process. You provide governance for all stakeholders' goals. You aim for as much automation as possible, to enable short release times and high software quality.

Development (undefined, orienting, beginning, proficient, advanced)

ID	Beschrijving	Type	Level
1.1	Sprint 0	Group project	orienting
1.2	Sprint 1	Individual project	orienting
1.3	Sprint 1	Group project	beginning
1.4	Sprint 2	Individual project	beginning
1.5	Sprint 3 & 4	Individual project	proficient

Substantiation

1.1: Sprint 0

We still have not set up the DevOps, however we have set up the development and testing environment, which consists of GitHub.

1.2 Sprint 1

I have set up the DevOps, which uses AzureDevOps, the GitHub repository (which will also be used as CI/CD).
I also have Docker-ized the Event Bus.
GitHub link: https://github.com/Snechar/WatchTime
Event Bus Docker: https://hub.docker.com/repository/docker/snechar/eventbus

1.3 Sprint 1

As a class we have moved together to Jira, where we are working in a professional way. The task is linked to GitHub.
As a group, we have set up even more automatic CD pipelines in GitHub and in the Docker itself.

1.4 Sprint 2

I have continued using Azure DevOps and I have linked user stories to pulls on GitHub. I have also fully set up CD in my pipelines and I am currently working to set up the CI as well, however a few problems have arisen while trying to test a NATS Connection, since GIT does not respect the port of the docker container while running tests, so the tests may fail randomly due to the NATS server being hosted on a different port.

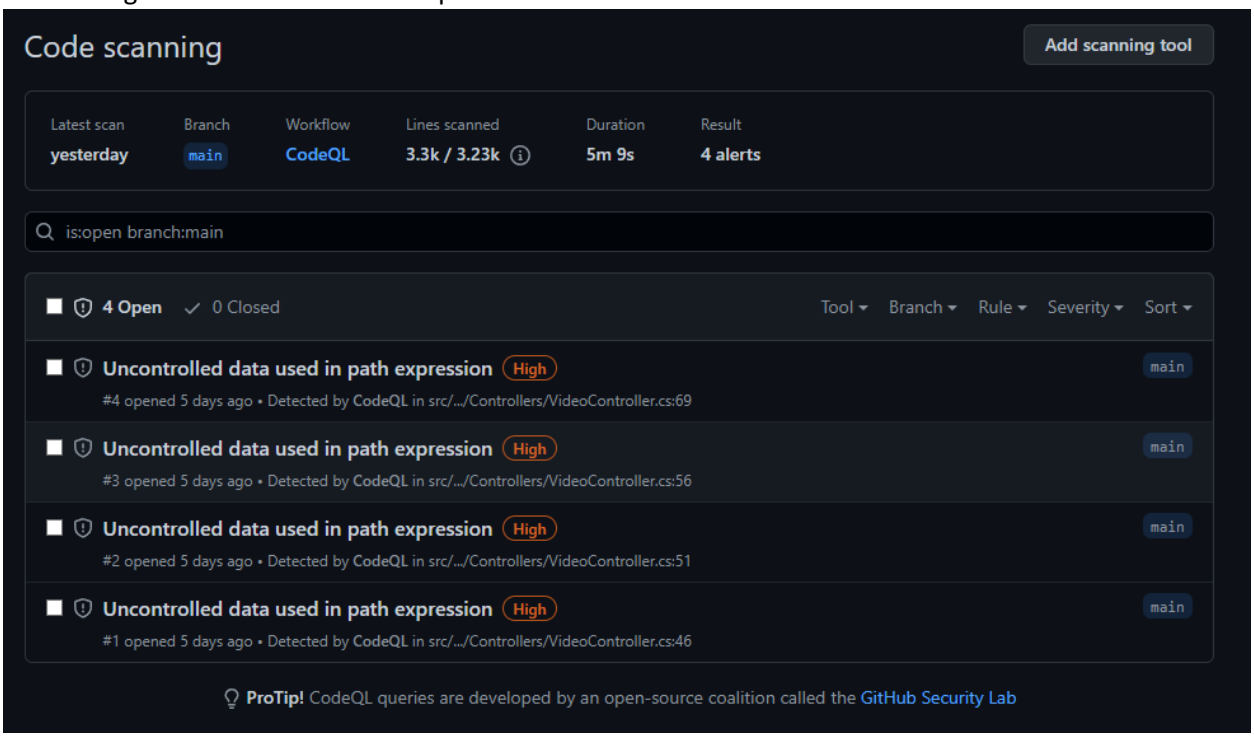
1.5 Sprint 3 & 4

During this sprint I have implemented many DevOps systems. While CI is still not complete, I now have a full CD workflow where I upload changes to Git, the workflows check if the application is built and then it gets updated on Docker Hub.
I have an example of a full CI implementation in my previous project, Voughtify – Music Streaming App, which is an Enterprise Software Application based on the Monolithic Architecture, which recently started being considered as a non-viable architecture for enterprise software.

I am using Keel, a Kubernetes service which allows me to set up auto-updates on DockerHub image updates. Keel checks whether an update has been pushed on the used images every set time interval, and if so, updated the pods affected. It also has a few other features like Notifications, Policies and Polling, which are more in-depth explained on their GitHub page.

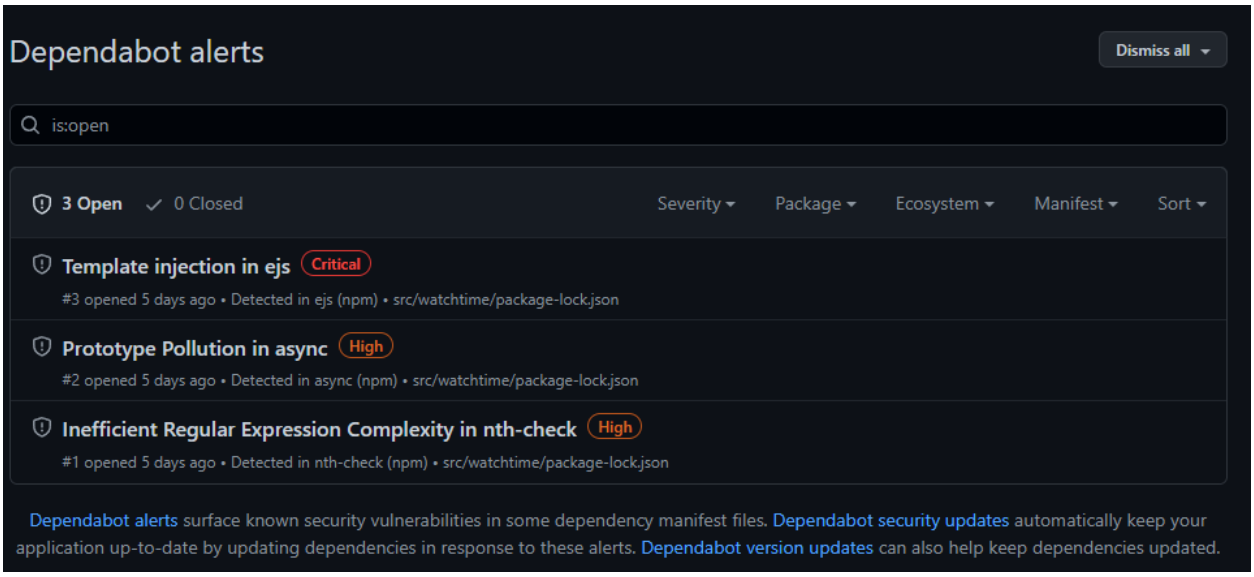
Keel can also be used in cloud deployed applications, but since my deployment is local, I only use the local features of the service.

I started using CodeQL as a "security" tool, however it is vastly inferior to SonarQube. The free version of SonarQube cannot be implemented into GitHub, so I only used it a few times locally.
In the image provided below, you can see that CodeQL finds uncontrolled data used in path expressions even though the variable is not an input from the user.



CodeQL finding 4 false positive of Paths that are not user input.

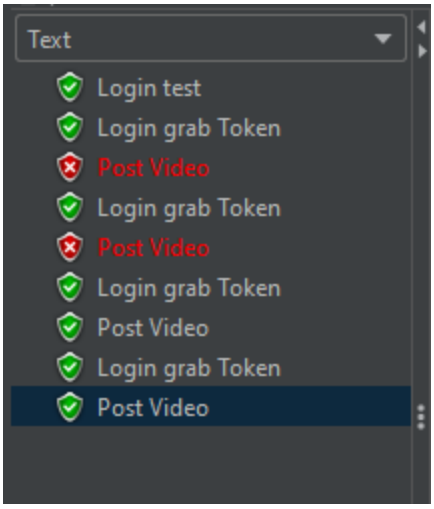
I also started using Depandabot, which so far has warned me of 3 different possible problems in my application, all of them having to do with React updates.



Dependabot alerts.

These alerts have been auto-fixed by Dependabot in pull requests which could auto-merge.

For testing the deployment, I am also using JMeter as a stress test. So far, I have tried to test some of the API calls using 500.000 users in the case of the Login/Renew functions, and 5.000 for the post video functions.



JMeter Tests

As you can see in the image above, most tests have passed since they have been faster than the target time, however the video service struggles when too many videos are uploaded at the same time, and the tests have failed since the speed of the upload was too low. The expected time for 5.000 1MB videos (5GB of data) was set to 500 seconds (10MB/s), however in the 2 failed tests it took 557 and 668

seconds respectively for the upload. Note that the tests have been done using only one copy of the service, as Ingress-Nginx would have blocked the IP if this many requests were made normally.

JMeter can be implemented in GitHub but requires GitHub premium, which I do not have, so the tests will have to be done locally and cannot be automated.

Reflection on progress

1.1: Sprint 0

This learning outcome will be reached when a good DevOps pipeline has been reached – a DevOps linked to GitHub, full CI/CD implementation, etc.

1.3 Sprint 1 & 1.4 Sprint 2

I am slowly achieving a complete DevOps pipeline, once CI has been implemented, the only thing that is left to do is to fine-tune the process. I think that I am much closer to completing the learning outcome that in the previous sprint.

1.5: Sprint 3 & 4

I have achieved a full CD implementation together with CodeQL and Dependabat as Security Tools, and even though CI is not fully implemented, I provided a full CI implementation of my previous Enterprise Software project, which used a monolithic architecture. The testing itself would not be too different anyway.

There will also be a full CI implementation in the group project.

6. Cloud Services

You can explain what a cloud platform provider is and can deploy (parts of) your application to a cloud platform. You integrate cloud services (for example: Serverless computing, cloud storage, container management) into your enterprise application, and can explain the added value of these cloud services for your application.

Development (undefined, orienting, beginning, proficient, advanced)

ID	Beschrijving	Type	Level
1.1	Sprint 0	Group project	orienting
1.2	Sprint 1	Individual	orienting
1.3	Sprint 2	Individual	beginning
1.4	Sprint 3 & 4 & Sprint 3	Individual & Group	proficient

Substantiation

1.1: Sprint 0

Due to the enterprise software architecture, we can easily use cloud services in our project, we have already discussed about cloud database scalability and autoscaling with the group, and will continue research in the following sprints

1.2: Sprint 1

Due to the nature of my individual project, scalable databases might not be needed, however what the project needs is auto-scaling and monitoring, which can easily be achieved with cloud services.
In the following sprint I will do further research

1.3 Sprint 2

I have made my docker containers cloud safe by binding the ports to the containers. Finding a suitable cloud service to host my containers is the only thing left before I can host them.
Due to the architecture that was designed in the beginning, it is extremely easy to have the services hosted in the cloud. As long as the NATS server is accessible, all the other microservices will function as normally.

I have also started setting up the Kubernetes cluster and tested it using 30 copies of the pods, this is the most my home computer can handle without completely crashing.

1.4: Sprint 3 & 4 & Sprint 3

Currently I am not hosting my project on any cloud services in my personal project, however, as discussed in Scalable Architecture 1.4: Sprint 3 & Sprint 4 , I could easily use a cloud database provided by Amazon.
I also use Okteto, which is a cloud staging environment, to test if my K8S YAML files build correctly. Okteto has many other features that would be extremely useful for development, however I am not using them as they would require a lot of change in my existing pipelines.

Due to how my K8S cluster has been set up, putting the files in a cloud service would be as simple as uploading the YAML files on the cloud platform and editing the Ingress-Nginx settings to change the IP. I would also need to find a fix for the database, but if I were deploying the application, I would pay for a cloud hosted scalable SQL database.

Cloud services offer a lot of help for development, as it can be used as a staging or deployment environment and it offers a lot of information in the form of metrics, data, etc. Not only that, but it can help people with slower computers, by taking the load off the local machine.
Most cloud services also come pre-configured, meaning that the developer has less work to do when deploying an application.

Reflection on progress

1.1: Sprint 0 & 1.2 Sprint 1

With good architecture cloud services can be easily implemented later in the development cycle, as all that needs to be changed is the IP of the communication.
Cloud services also take care of a lot of problems developers must deal with, like automate scalability, monitoring, resource management, etc.
This learning goal will be reached when the services have been hosted on the cloud.

1.3 Sprint 2:

While I am making progress on the cloud services in my individual project, I will not host my individual project services on the cloud, the credits provided by Fontys are not enough to host the services for more than a few days because the boot-up of the service is very expensive.
While I do believe I am closer to reaching the outcome, I will try to reach this outcome in the group project using the cloud services offered by the PO and Fortress.

7. Security by design

You investigate how to minimize security risks for your application, and you incorporate best practices in your whole software development process.

Development (undefined, orienting, beginning, proficient, advanced)

ID	Sprint	Type	Level
1.1	Sprint 0	Group project	Orienting
1.2	Sprint 1	individual	Orienting
1.3	Sprint 1	Group	Beginning
1.4	Sprint 2	Individual	Beginning
1.5	Sprint 3 & Sprint 4	Individual	Proficient

Substantiation

1.1: Sprint 0

We will follow the security by design principles which are mentioned in the provided document.

1.2: Sprint 1

I will also follow the security by design principles.

1.3 Sprint 1

In this sprint I have done very important research due to the remarks of the teachers – data encryption. In the group project we are currently encrypting data using the AES-256 14 block GCM, which is currently the industry gold standard.

1.4 Sprint 2

In this sprint I have made the account microservice, which needs to be very secure due to the handling of persona information such as passwords that might be reused by users in other applications.

For the account itself I am using the Microsoft Identity package combined with the JSON Web Token package. These 2 combined offer a huge amount of security in the form of Claims, where the user has to be identified using the JWT validation. On top of that, a valid issuer (microservice) and audience (user listening location) need to be valid.
The passwords are, of course, hashed using a 516bit salt.

JWT Link: https://jwt.io/
Identity Link: https://docs.microsoft.com/en-us/azure/active-directory/develop/

Not only that, but all the data that goes through the event-bus has been also encrypted using AES-256 14 block GCM.

AES: https://en.wikipedia.org/wiki/Advanced_Encryption_Standard
GCM: https://en.wikipedia.org/wiki/Galois/Counter_Mode

I have also made sure to make my code immune to SQL injections by using Entity Framework, sanitizing all inputs, and following their considerations.

Entity Security:https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/ef/security-considerations.

1.5 Sprint 3 & 4

CIA Triad

Confidentiality

Issue	Risk Level	Damage Level	Prevention methods
Mishandled Privacy	Low	High	• Correct implementation of role-based and user-based authorization and verification • Categorization of data into security levels
Sensitive user data safety	High	Very High	• Extra protection for sensitive data such as accounts in the form of 2FA, biometric validation, ID validation, etc. • Systems for checking potential hacks, such as IP address checks, unique token checks, etc.
Data safety	Medium	Low-Very High	• Correct implementation of authorization. • Extra role-based checks. • Deliver sensitive data only when needed to the right people.

Integrity

Issue	Risk Level	Damage Level	Prevention methods
Data theft	High	Low-Very High	• Verification and validation of data should come from a trusted source and travel through secure means that do not allow it to be intercepted or tampered with. • Secure storage of data in a system where it cannot be stolen or tampered with.
Data validation	Very Low	Very High	• Always back up the data regularly, since corruption can randomly happen from anything from a power outage to cosmic rays hitting a resistor.
Data safety	Medium	Low-Very High	• Correct implementation of authorization. • Extra role-based checks. • Deliver sensitive data only when needed to the right people.

Availability

Issue	Risk Level	Damage Level	Prevention methods
DDOS attacks	Very High	Very High	• Implement methods against DDOS attacks such as the NGINX gateway, timeouts, etc. • Implement methods to make the attacker unable to find the servers of the application, such as masking, IP hiding, proxying, etc.
Scalability issues	Medium	Low	•
Data safety	Medium	Low-Very High	• Correct implementation of authorization. • Extra role-based checks. • Deliver sensitive data only when needed to the right people.

OWASP Top 10

1. **A01:2021-Broken Access Control**

In my application I am using Entity Framework together with Identity and JWT.
When the user is logged in, a JWT token is sent and saved as a cookie, which will later be sent as a header when a request is made that requires authorization.
The token has a valid issuer, which is a key produced by the Authorization service, a valid audience, which is a key generated for the user, and Role information of the said user.
When a call is made, the API controller checks the issuer, audience, user, and roles of the JWT and decides whether the user has access or not based on a Header parameter in the API call.

To implement these features, I have consulted the DotNet Security Cheat Sheet

2. **A02:2021-Cryptographic Failures**

No data other than Video Titles and video statistics (likes, dislikes, quality) is transferred without encryption. I am also using an up-to-date encryption standard, AES 256 14 block GCM, which is one of the most widely used encryption due to the security it provides.
There are no legacy protocols being used and all sensitive data is either encrypted or hashed.

The authentication details are hashed using a complicated one-way hash provided by Identity.

3. **A03:2021-Injection**

I am using Entity Framework, which is a highly developed DB communication layer that already has very powerful anti-injection features, however I am also following the official Security Considerations provided by Microsoft.

In the case of user inputs, I verify and sanitize all user inputs like video titles, file names, file meta type, etc.

4. **A04:2021 – Insecure Design**

Most of this category does not apply to my project, because I am using a design that is already OWASP compliant, but there are still a few things I would like to discuss.

In the application, I am using Ingress-Nginx as a gateway, load-balancer, and user-related resource consumption limiter. It allows me to temporarily block users who are abusing the system based on metrics and logic. Currently a user can only upload a maximum of 2 videos at once, and they cannot go over 2GB each.

Because of Ingress-Nginx, it is also hard to use bots, since it already has features implemented to counter these automatically, such as browser validation and more.

5. **A05:2021 – Security Misconfiguration**

The application has a fully automatic CD where the code quality is being tested using CodeQL.
It is also containerized into different pods, meaning that if one of them has a security issue, K8S will instantly let me know and stop traffic to the issue automatically.

The application also has no unused/unnecessary code left, as everything that exists is used, and the unused code is removed/disabled.

6. **A06:2021-Vulnerable and Outdated Components**

In the CD pipeline I am using Depandabot to warn me of possible outdated and vulnerable components. This trusty bot also checks if the components can be automatically updated without making any changes to the code, and if so, automatically fixes them. If an auto-merge is not possible, it will ask the developer to update the vulnerability as soon as possible.

I have also taken steps to remove all unused dependencies in the C# API and React website.

7. **A07:2021-Identification and Authentication Failures**

Let us start by discussing about passwords. A lot of people like to use simple to crack passwords as these are very simple to remember. This can lead to their account easily getting hacked by using brute force and the top 10.000 worst passwords list.

In my application I am using Entity Framework, which is constantly updating in order to meet with the standards of the National Institute of Standards and Technology. Not only that, but it also has very strict rules on passwords, needing to have at least 10 characters, 1 special character, 1 number and 1 capital character. While the latter ones do not affect the complexity that much, it can be easily set up so that a password needs to be at least 18 characters long, which would take an approximation of 7 quadrillion years to brute force (source https://www.security.org/how-secure-is-my-password/)

While currently multi-factor authentication is not implemented in the project, it can be implemented with Identity as an e-mail code, and the only thing it would take to implement is changing a bool from false to true and attaching a PDF mold.

8. **A08:2021 – Software and Data Integrity Failures**

This is a new category in the OWASP top 10, and it refers to an insecure CI/CD pipeline.

I order to make my pipeline more secure, as mentioned before, I am using Depandabot and CodeQL for code verification.

Since I am working on my own, there is no code review, but we are constantly doing code review on pull requests in the group project.

9. **A09:2021 – Security Logging and Monitoring Failures**

Due to the nature of the project, everything is logged either by Ingress-Nginx or the event-bus, the event-bus logging is not connected to anything at the moment, however the Ingress-Nginx logging is constantly used in order to detect possible malicious movements and different types of attack on the server.

The logging data of the event-bus is sanitized and then encrypted.

In my application there are no transactions, and therefore there is no need for transaction integrity.

10. **A10:2021 – Server-Side Request Forgery (SSRF)**

In both the group and individual projects, all data is sanitized, and we are enforcing the URL schema, port and destinations using Ingress-Nginx. Http redirections are disabled by default as they provide a possible exploitation front. In my individual project I also enforce a Deny-by-Default policy to block traffic to non-curated URLs

Reflection on progress

1.1: Sprint 0

This learning outcome will be achieved when I can prove that my application is secure against cyber-attacks.

1.3 Sprint 1 & 1.4 Sprint 2

I have been following the security standards and I implemented hashing and encryption for sensitive data. I think that I am getting closer to reaching the learning outcome

1.5 Sprint 3 & Sprint4

I personally think I have achieved this learning outcome as I have proven that I have the knowledge needed to build a secure application, as well as applying the knowledge gathered so far.

8. Distributed data

You are aware of specific data requirements for enterprise systems. You apply best practices for distributed data during your whole development process, both for non-functional and functional requirements. You especially take legal and ethical issues into consideration.

Development (undefined, orienting, beginning, proficient, advanced)

ID	Sprint	Type	Level
1.1	Sprint 0	Group project	Orienting
1.2	Sprint 1	Individual	Orienting
1.3	Sprint 2	Individual	Beginning
1.4	Sprint 3 & Sprint 4	Individual	Proficient

Substantiation

1.1: Sprint 0 & 1.2: Sprint 1

Due to the microservice architecture, we must use the microservice specific implementation, which is that each microservice should have its own database.

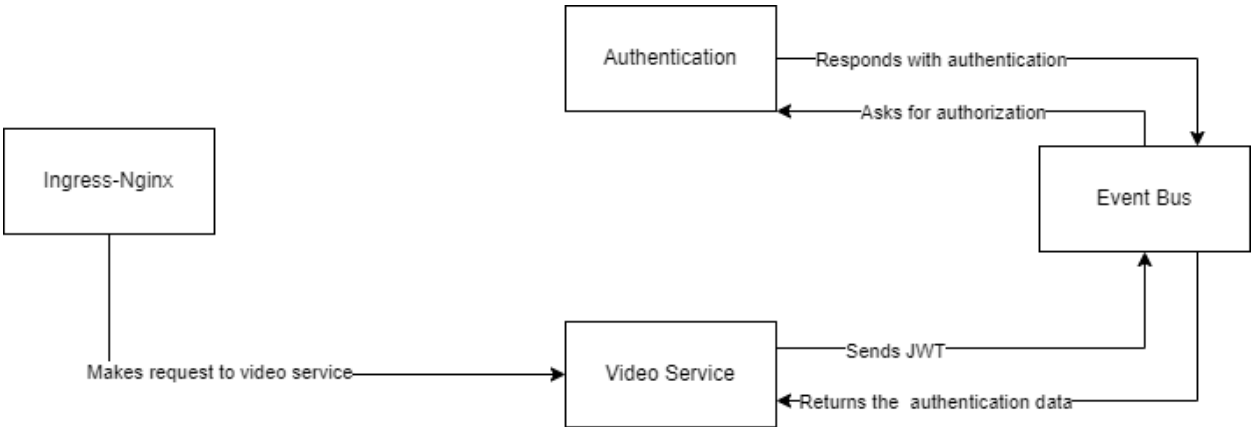
1.3 Sprint 3

I created a Consistency Keeper for the event bus, which keeps data between microservices consistent. As before, I am still following the best practices for Microservices, which is that each microservice has its own database.

1.4: Sprint 3 & Sprint 4

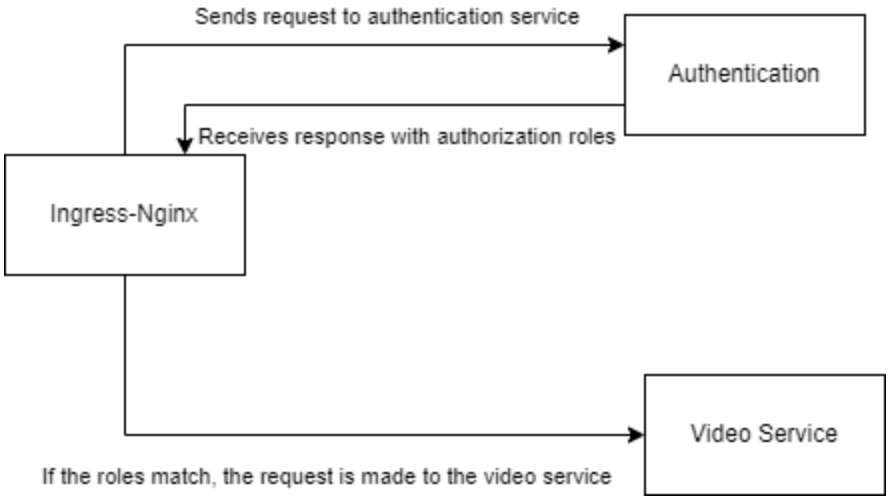
In the microservice architecture, the handling of data is quite complicated, as authentication needs to be done for the entire application, multiple microservices require copies of some data to function, and eventual consistency must be reached.

Below, there is a diagram of how authentication currently works in the application, however this could change for the better if I got access to Nginx Plus.



Current Authentication

Nginx Plus has a special feature where you can specify the authorization call and get authorization from within the gateway itself. This saves on processing power and results in shorter response times.
In that case, the diagram would look like this



Nginx Plus Authentication

For the duplicate data and eventual consistency, I am currently using the following system:

When a new account is added, it is published on the event bus. If no microservice receives it, it will be delivered later.

If it fails to be delivered to a microservice somehow, when the said microservice requires the user, a call will be made to the event-bus, which will ask the Authorization microservice to deliver the data of the user, encrypted.

For the duplicate data, I am currently using the Username, ID, and E-mail of the user.

The data collection is also respects the GDPR rules in the following ways:

1. **Lawful, fair, and transparent processing**

   The only data processed in the application is publicly available videos, comment, likes and dislikes.

   This requirement does not affect the application that much unless a paid service would be offered.

2. **Limitation of purpose, data, and storage**

   The only purpose of the data gathered is to offer a better experience by allowing users to find high quality videos and reward the creators who post said content.

   There is currently no need for personal data other than the account itself, but later the application might need ID verification, for which the data will only be requested to prove that the user is who he says he is.

3. **Consent**

   The application uses data only for legitimate purposes and nothing else, therefore consent will not be needed.

4. **Privacy by design**

   As mentioned in the Security by Design Learning Outcome, the application implements privacy using a combination of authorization, roles, encryption, hashing, etc.

5. **Data Protection Impact Assessment**

   N/A

6. **Data Transfers**

   The application does not use third party data processing. In the case of implementing Ads in the website, protection of potentially sensitive personal data has to be implemented and taken care of when delivered to a third party data processor.

7. **Data Protection Officer**

   N/A

8. **Awareness and training**

   I have been “trained” by studying the GDPR requirements.

9. **Data subject rights**

   A user always has access to data about him.

Reflection on progress

1.1: Sprint 0

I am not sure how to achieve this outcome since we are stuck using the microservice implementation with no real way of explaining why other than the “best practice”.

1.3 Sprint 2

I think I am getting closer to the learning outcome by respecting the design principles of enterprise microsystems.

1.4: Sprint 3 & Sprint 4

I have gained knowledge in microservice distrusted data, and I know what the GDPR requirements are and how to apply them. I personally think that I have reached the learning outcome by showcasing my knowledge and how I implemented it.

Reflection

Sprint 1

For the beginning of a project, I think we are doing well. We have created a strong base for implementation with the enterprise design principles that we have been following so far.

The group communication is going amazing, we all voice our opinions and discuss every problem we encounter. This also applies to the whole class – I feel like the 3 different groups are more connected than groups I had in the previous semester.

I do however have some problems with my individual project. I have run into a problem similar to one from semester 3. This problem in video streaming like YouTube, since a video cannot just be fully loaded before playing, as this would take entire minutes, especially when talking about very long videos.

I do not know who to discuss this problem with, as I have already done a lot of research and I could not find anything.

Sprint 2

Overall, this sprint was good, I have learned a lot about all the learning outcomes and coded quite a bit.

Our group's way of working has become more professional and now we have a simple version of DevOps, and the communication with other groups has been way better.

In my individual project I am learning a lot about back-end microservices, microservice architecture itself, scalability, and cloud services.

I do still have the problem of not knowing how to stream videos, I have created a prototype, but it does not work as expected. I am thinking of creating an API for the video streaming and using protocol 206 to stream content.

Sprint 3

This sprint has been very hard, I have done A LOT of coding, research, and implementation in the past 3 weeks.

First, I finally have an implementation for video streaming, which took 2 weeks of constant research and prototyping. I am proud of what I have done with it, and I think I might turn it into a NuGet package in the future if I have time.

I would say that the most amount of work went into Cloud Services and Scalable Architectures, where I managed to implement a full K8S cluster, that has a gateway, load balancer, autoscaler based on 3 relevant metrics, pod auto-updater, and a few more features.

A lot of work has also been put into DevOps, where I completed my CD pipeline with 3 extra tools – CodeQL, Dependabot, and JMeter (technically also the pod auto-updater), and now once I push changes to one of the microservices, it will automatically get updated in the deployment after a few minutes.

There was also research done for the security part of the application and the distributed data topic.

Conclusion

Here, at the end of the semester, you reflect on your process and end result. You can also refer back to the goals you set in the introduction, to see to what extent you were able to achieve them.

Also mention what you are proud of, what you would like to do differently in the coming semesters and whether you have come to different insights about the field of study. Is there perhaps a particular subject you would like to explore further?