Functional & Technical Design

Kremer,Yordi Y.C.T.J.

[Company name]  [Company address]

# User Stories

* A s a user I can watch a video simultaneously with someone.
* As a user I can invite someone into my room.
* As a user I can pause the video.
* As a user I can play the video.
* As a user I can skip to a certain part in the video.
* As a user I can chat with people in the same room.
* As a user I can create a room.
* As a user I can create an account.
* As a user I can login.
* As a user I can logout.
* As a user I can add videos to a playlist.
* As a user I can vote to skip the video to the next video in the playlist.

# Moscow

|  |  |  |  |
| --- | --- | --- | --- |
| Must have | Should have | Could have | Won’t have |
| As a user I can watch a video simultaneously with someone. | As a user I can chat with people in the same room. | As a user I can add videos to a playlist. | As a user I can kick someone from my room |
| As a user I can invite someone into my room. | As a user I can skip to a certain part in the video. | As a user I can vote to skip the video to the next video in the playlist. |  |
| As a user I can play the video. | As a user I can add users to my friends list |  |  |
| As a user I can create a room. | As a user I can login. |  |  |
| As a user I can pause the video. | As a user I can logout. |  |  |
|  | As a user I can create an account. |  |  |

# C4 Level Diagram

C1 Diagram

A screenshot of a phone

Description automatically generated

C2 Diagram

A screenshot of a computer

Description automatically generated

# Scalable Architecture

How is my architecture scalable? I use the Microservice architecture for my project I chose for this architecture because it is very scalable below I have listed points of how I can make my project more scalable and how microservices will help me with that.

## Service Independence

The point of microservices is that you break down your applications into smaller independent deployable services. This independence has a lot of benefits such as developers can work on different services which is very convenient, you don’t hinder each other in your work and with microservices every microservice could be built with a different programming language, this is very helpful if your functionalities of that microservice are better with a certain language or if your team is better at coding in a specific language.

## Communication Protocols

For my communication between my microservices I will be using Http , I also looked into implementing RabbitMQ but I had my doubts about this because RabbitMQ queues your messages without expecting a response back immediately this means that a request could be in a queue for like 2 hours, because my application is all about real-time communication this does not seem a good idea. So for my frontend to my backend, I will be using web sockets to get this real-time communication feeling and between my microservices I will communicate with HTTP requests.

## Containerization

My whole application will be dockerized this is to ensure consistency across different environments this to solve the “it works on my machine problem”. More reasons such as isolation ease of running my applications, docker makes your applications very portable which is useful for me because I work on this project from my home pc and my laptop so I can easily transfer containers. With the implementation of a container orchestration platform like Kubernetes will it make my application very scalable because this can ensure high availability and optimized resource utilization.

## Observability

With a microservice architecture you could also very easily implement a monitoring and logging tool to get insights into health, performance and interactions between microservices.

# DevOps

To Achieve this learning outcome I will be Implementing DevOps aspects in my project below I will explain how I will implement these and why I choose to implement this into my project.

## Continuous Integration

For my project I will implement CI pipelines to automatically build and test my application whenever changes are pushed to my git repository. This to make sure that bug fixes are quickly integrated into my project this way I can maintain the quality of my application.

## Continuous Deployment

For my project I will also implement CD pipelines to automate the deployment process of changes made to my application. This to ensure user error and it is just very convenient to automate the deployment to the cloud.

## Automated Testing

I could potentially also implement automated testing frameworks into my application to thoroughly test different components of my Video syncing platform. These tests will help me catch bugs early while developing, allowing me to work more rapid.

## Containerization

Reasons for me to dockerize my project are isolation and ease of running my applications, docker makes your applications very portable which is useful for me because I work on this project from my home pc and my laptop so I can easily transfer containers

# Cloud Native

I will explain below how I will achieve this learning outcome and the way cloud native gets involved into my project, I will also explain my ideas of how to implement this. For my project I will be making use of Azure cloud services because I get €100 euros free with my student account.

## Cloud-Native Architecture

My microservice architecture is a important design choice for my Cloud Native Architecture principles such as it being highly scalable, resilient because of my microservice architecture my application is broken down into smaller loosely coupled services that can independently developed, deployed and scaled emphasizing modularity service independence and fault tolerance in my design.

## Containerization

Containerization is also very important when looking at aspects of cloud native development this is important to facilitate seamless deployment and scaling of my project.

## Cloud Services

I will use Azure Kubernetes Services to manage my docker containers and host my application, this will also function as a good load balancer and managing tool for my project. I will also have my PostgreSQL database within my AKS cluster to reduce latency.

# Security By Design

## Authentication & Authorization

For my Watch2gether clone project I will be implementing Authentication & Authorization for my project to control access to certain data. This will also be used so that a user can invite friends to a room and watch videos together.

## Containerization

Dockerizing my project is important for security in my docker compose I implement multiple networks one network is for the communication between my frontend my API gateway and my microservices and another network is used for the communication between the microservices and their databases by doing this for scalability, different security network policies could be added to these networks. I will implement runtime options for my docker containers, this will give my container a set memory, CPU and GPU usage, this will help me get more control of my containers and if one should go rogue due to malicious activity they can only use the set amount of resources.

## Secure coding

By following coding best practices and regularly conducting code reviews with my technical teachers I ensure that my code remains resilient against known security threats, mitigate common vulnerabilities such as injection attacks. Look at OWASP top 10 and think how I can prevent these vulnerability issues.

# Distributed Data

For Distributed data I use microservice architecture this way each microservice that requires a data storage has their own database, this way data is nicely divided into their own area and the microservice only has access to the data specifically needed for that service. By using Azure Kubernetes services I will also have the option for my databases to be duplicated if they have a lot of traffic while still using the same volumes to store the data. Another benefit for this is that if one of my databases fails the rest of my application still stays functional by using the other databases.