6/15/2024

Yordan Doykov

Individual Sprint Deliveries

Sprint 1 – Sprint 5

Contents

[Introduction 2](#_Toc168054705)

[Sprint 1 2](#_Toc168054706)

[Work done 2](#_Toc168054707)

[Documentation 2](#_Toc168054708)

[Project Plan 2](#_Toc168054709)

[Architectural choice document 3](#_Toc168054710)

[Research plan + report 3](#_Toc168054711)

[Architectural document 3](#_Toc168054712)

[Next sprint 3](#_Toc168054713)

[Sprint 2 3](#_Toc168054714)

[Work done 3](#_Toc168054715)

[Documentation 3](#_Toc168054716)

[ERD + Database implementation 4](#_Toc168054717)

[Security SDLC + Security report 4](#_Toc168054718)

[Cloud research 4](#_Toc168054719)

[CI/CD Setup 4](#_Toc168054720)

[Next sprint 4](#_Toc168054721)

[Sprint 3 4](#_Toc168054722)

[Work done 4](#_Toc168054723)

[Kubernetes 4](#_Toc168054724)

[Load testing 5](#_Toc168054725)

[GDPR compliance 5](#_Toc168054726)

[Next sprint 5](#_Toc168054727)

[Sprint 4 6](#_Toc168054728)

[Work done 6](#_Toc168054729)

[MongoDB deployment 6](#_Toc168054730)

[Amazon S3 7](#_Toc168054731)

[Google Cloud 7](#_Toc168054732)

[Next sprint 7](#_Toc168054733)

[Sprint 5 8](#_Toc168054734)

[Work done 8](#_Toc168054735)

[Professional standard documentation 8](#_Toc168054736)

[Application monitoring 8](#_Toc168054737)

[Chaos engineering 9](#_Toc168054738)

# Introduction

This document is a compilation of all of the sprints that were carried out for the individual project of the Sem. 6 Advanced software.

All of the relevant resources for the individual project can be found in the canvas portfolio handins, as well as the gitlab repository

<https://git.fhict.nl/I476740/s6-househunters>

# Sprint 1

## Work done

This chapter has the purpose of briefly presenting the progress made during the sprint.

During Sprint 1, a lot of progress was done with the individual project. Everything from architectural choice, building the initial architecture, implementing most FRs of the application, as well as partial quality assurance. T

## Documentation

The following documents have been created during Sprint 1:

* Project Plan
* Architectural choice document
* Research plan + report
* Architectural document 1st draft

## Project Plan

The project plan establishes the context, scope and future work of the project. The delivery of the plan can be found in canvas.

## Architectural choice document

The document goes through the major architectural and framework choices chosen for the individual project.

## Research plan + report

These 2 documents revolved around personal research, which I decided to be message brokers. Since message brokers a part of the fundamental purple blocks of this semester, I conducted this research quite early

## Architectural document

A 1st draft of what the entire architecture of the app would look like. Currently only has C1 and C2 diagrams, but in the future will feature the CI/CD setup and Security explanations too.

## Next sprint

Next sprint the focus will be on the following blocks:

* Finish software quality assurance
* Security
* Optionally ask for guidance on the cornerstone blocks

# Sprint 2

## Work done

This sprint was really productive and all of the goal from the previous sprint were achieved.

During Sprint 2, a lot of progress was done with the individual project. Rounding up quality assurance through creating e2e tests and security tests, as well as expanding my CI/CD pipeline to a near-complete setup.

## Documentation

The following documents have been created during Sprint 2:

* ERD + Database implementation
* Security SDLC + Security report
* Cloud research
* CI/CD Setup

## ERD + Database implementation

Quick explanations on what the entities in the application are, as well how they are implemented in the different databases. This was created on request by the teachers.

## Security SDLC + Security report

Documentation that goes through how I implement security by design for the different stages of the SDLC, as well as a security report with improvements made to the project by following OWASP’s top 10 guidelines.

## Cloud research

A document analyzing the project’s architecture and deriving which parts of it can be pushed to cloud. Costs to do that are also calculated.

## CI/CD Setup

Overview and explanations of how the CI/CD pipeline is setup and functions.

## Next sprint

Next sprint the focus will be on the following blocks:

* Diving into Kubernetes and making the application scale properly
* GDPR + data complexities

# Sprint 3

## Work done

This sprint was really productive and all of the goal from the previous sprint were achieved.

During Sprint 3, the progress on the individual project was mainly on the scalability part of the application, as well as its GDPR compliance

## Kubernetes

This sprint dived deep into Kubernetes and deploying the application to a local Kubernetes cluster using Minikube. By the end, full local deployment was reached using pods, services, autoscalers and an ingress controller where the system was able to automatically scale components independently depending on the load.

A screenshot of a chat

Description automatically generated

## Load testing

After having the application deployed to Kubernetes locally, autoscaling was fine-tuned by conducting various load tests using artillery. With the discoveries of the load tests, various improvements regarding performance and smarter queries were made.

A black background with many small dots

Description automatically generated

## GDPR compliance

GDPR compliance was also reached during this sprint for HouseHunters. The necessary steps were taken, such as adding a privacy policy that the users have to accept in order to register and use the application, as well as implementing the right to be forgotten.  
A screenshot of a computer

Description automatically generated

## Next sprint

Next sprint the focus will be on the following blocks:

* Raise the “Cloud” part of the application to proficient by implementing the findings in the cloud providers research

# Sprint 4

## Work done

This chapter has the purpose of briefly presenting the progress made during the sprint.

During Sprint 4, the work planned during the past sprint was achieved. All of the plans to move certain parts of the individual project to cloud were completed successfully, which achieved proficient on all learning outcomes.

## MongoDB deployment

For production, it was decided that the databases will be deployed to the cloud and managed by MongoDB themselves. It provides both automatic horizontal and vertical scaling for their clusters.  
The cluster that was chosen was the free tier of cluster with shared ram and 512MB of storage. The server is located in Paris and is provided by AWS.

A screenshot of a computer

Description automatically generated

## Amazon S3

For storing media in the application, It was decided to go with Amazon's S3 buckets. They are quite popular, feature nice documentation and a big community. Even though S3 is not a standard, its popularity has an impacting influence on other cloud providers that feature compatibility layers with it.

The bucket used for production is located in London. Below is a screenshot of what it looks like:

A screenshot of a computer

Description automatically generated

## Google Cloud

Google cloud was used for deploying the microservices and frontend.

The deployment takes place automatically in the CI as a last stage of the main branch. It uses kubectl to declaratively apply the manifest files to the kubernetes cluster. The stage takes advantage of a gcloud service account that is priveledged to only execute actions on the specific gcloud project.

A screenshot of a computer

Description automatically generated

## Next sprint

Next sprint will focus on getting as many learning outcomes to the “advanced” level as possible. For this, a few actions can be taken:

* Implement application monitoring, either with a cloud provider or manually
* Create better documentation for Personal leadership and Professional standard
* Explore scalable functions as a comparison to the achieved architecture thus far.

# Sprint 5

## Work done

This chapter has the purpose of briefly presenting the progress made during the sprint.

During Sprint 5, most the work planned during the past sprint was achieved. The finishing touches were made to the individual project that were required to achieve advanced on most technical learning outcomes.

What was also achieved was sorting out the documentation regarding the first 2 soft-skills learning outcomes in order to reach proficient there.

## Professional standard documentation

This sprint a lot of emphasis was put on getting feedback from Nicole in order to make sure that proficient is achieved on the first 2 learning outcomes. This lead to a whole bunch of documentation being created, such as a semester reflection, implementing feedback on the personal research and creating transfer documents for the next group that works on the project.

## Application monitoring

In order to reach advanced on the DevOps learning outcome, robust application monitoring was implemented in the individual project with Sentry.

A screenshot of a website

Description automatically generated

In my case, I created an organization called "HouseHunters" with projects inside for all my microservices that are being monitored. Sentry features a node SDK which is relatively easy to install and setup.

Sentry captured relevant errors/exceptions within the application, along with performance data. As a result, any issues that occur were immediately sent to sentry and could be viewed, along with tons of information regarding what exactly went wrong. To summatize, sentry monitors information regarding events, such as:

* OS with runtime name and version
* Stack trace (if applicable)
* Breadcrumbs from application console
* HTTP Headers
* System contexts like hardware information and IP
* Specific NPM packages that were used when the event happened

**A screenshot of a computer

Description automatically generated**

## Chaos engineering

To reach advanced on the scalability learning outcome, chaos engineering was performed this sprint.

This way I could determine how robust my application ,while fixing any issues and deriving improvements to make the application better.

Usually system components would be shut off under very high load when the system can't possibly maintain such a heavy traffic and experiences a crash. However, it is quite difficult to replicate such a large traffic without a dedicated machine to execute the load testing. For this reason, I decided to simulate a system crash by just manually shutting off certain components when the whole application was running.

Various parts of the system were shut down during operation including

1. Individual microservices
2. Database
3. Message broker

A screenshot of a computer

Description automatically generated

In the end, some issues were found and were resolved, which made the application even more robust