Readers Guide

Gökay Atalay

# Sprint 1

In sprint 1 I have been researching a project and the requirements. I started looking at frameworks to find out what works best for my project. When I looked at the “distributed” part, I found out that you build an application by creating multiple projects that communicate with each other. As a result, you could build each project in a different language/framework and it should work together.

I started a number of test projects and tested communication to get an idea of ​​how it all works. I also made some simple documentation to record my findings. I also created Github repositories to keep track of my work.

## LO1 - You design and build user-friendly, full-stack web applications.

## /Doc

In “Doc” there are a number of word documents in which I have noted my research findings. I've had to do research for several products/topics to better understand them. For the frameworks that I have researched, I have written down all kinds of commands and pieces of code with relevant explanations.

### /LaravelDocumentation

In this document I have explained several aspects of the framework. How to install the

Installer and use it to generate a project.

### /ReactDocumentation

This document describes various aspects of this framework/library

(everyone says something different): Generating a project, making HTTP requests and

A well-known npm package specific to React called React Router.

## /Testprojects

In “Test projects” are projects that I have created to test communication and authentication. I tried to have two projects communicate with each other to see which problems arose. I could use those findings for my “real” projects.

By setting up these projects I was able to try out different topics such as a Javascript framework, asynchronous communication and data storage.

### /Laravel-BE

This project retrieves data through an API that can be called from a frontend project. Laravel fetches some data from a database and returns it in a neatly packaged HTTP response.

### /React-FE

This project is the face of the whole application. It contains a UI and makes calls to the

backend project. It uses the *async* and *await* keywords to allow for asynchronous

communication.

## /GithubRepo

In the “FE” repo you’ll find a react application with serves as the frontend portion of the

complete app. It sends requests to other applications to retrieve/send data and uses asynchronous JavaScript. In another repo containing “BE” you’ll find a backend application which is responsible for interacting with a database.

## LO3 - You choose and implement the most suitable agile software development method for your software project.

## /GithubRepo/Projects

The project is divided into sprints. During these sprints a certain portion of the project is focused on and at the end of the sprint we upload the material that has been created during the sprint. Using GitHub Projects I also keep track of tasks and issues that have to be worked on.

## LO8 - You act in a professional manner during software development and learning.

## /Docs

Some of the docs (Authentication) were made with a teammate (Jordy). We researched

the topic and described our findings. We went through the different concepts and their

methods and made a choice as to which one to use.

# Sprint 2

In sprint 2, I continued to create documentation for topics I've done research on. I did some research and worked with OAuth 2 from Google. I tried to implement that together with Access Tokens. I also researched Spring Boot to see what it's like to work with. I have also agreed with Jordy that we will do the two researches together for our individual project. We had a short conversation with Leon about this and agreed that we would look at CORS & CSRF as research topics. We will also see if we can integrate Git Flow into our research.

*Added to:*

## LO1 - You design and build user-friendly, full-stack web applications.

## /Doc

## /SpringBootDocumentation

I wanted to add a service which would have a specific task. I decided to add a Spring Boot

Application which would serve data, which my application can display to the user. To be able

to use Spring Boot I had to do some research, as you have to do with every new framework

you use. I noted the findings of my research in the mentioned document.

## /TestProjects

### /SpringBoot-BE

I did some research on Spring Boot and watched some tutorials to get an idea on how it

works. After that I made a testproject to try and get it working myself. Very little was

needed to get a route working. After sending some requests I got the correct response back,

I could start building microservices with Spring Boot after relatively little testing.

## /Github

### /S3-IP-FE-GokayAtalay

In my repository I’ve added the documentation in markdown files so that they can be

accessible to potential teammates. Those files can be used as guides to the ones who

don’t have much knowledge about the subjects, that way they don’t have to do their

own research.

In the React repo specifically I’ve added basic features like context (to remember the user

throughout the app) and routing, to navigate to different pages. On top of that I’ve added

authorization with access tokens which retrieves the user object and stores it in the

session storage of the browser to keep the user saved during the session.

I made a page containing an async method that calls the Spring Boot API and retrieves all

tasks to show them to the logged in user. A user will later on be able to click on one of the

tasks and view details to determine whether or not this task fits their skillset.

### 

### /S3-IP-BE-GokayAtalay

I configured the project to allow only one origin, my react application, instead of all origins.

I’ve also converted the small research file into a markdown file, which makes it look better.

### /S3-IP-BE-SpringBoot-GokayAtalay

Created a Spring Boot application which serves as an API for my frontend application. It

contains a controller which will catch the requests and execute an according method. The

main method of the controller retrieves data from my MySQL database and returns it in a

proper HTTP request. It also allows only one origin to make calls to the API, which is my

React application.

## LO2 - You use software tooling and methodology that continuously monitors and improve the software quality during software development.

## /Doc

### /AuthDocumentation

I implemented security in my application because there will be a lot of user interaction. Creating your own authentication system can be unsafe, because you’ll likely overlook a lot of security concerns. For that reason, it’s best to leave that portion to someone who knows what they’re doing. That’s why I decided to implement OAuth with Google accounts. That way Google handles most of the authentication and just returns a response with which you can work.

I’ve integrated OAuth w/ Access tokens into my React application and wrote down the

Research in this document. It guides you through the process on how to implement this. This

portion of the app provides a bit of security for the user to have to authorize through their

Google accounts.

## LO8 - You act in a professional manner during software development and learning.

## /GithubRepo/Gokay-Jordy-S3-Onderzoek

I’m going to be working with Jordy for our researches. We created a GitHub repository in

which we will be saving our work. Also are we going to use Projects and pull requests to

keep track of our project.