Project plan

Version 1.0

March 7th, 2021

Document history

Versions

Version	Date	Author	Changes	Status
1.0	03-07-2021	Kevin Heijboer	Document setup and content	Concept

Table of Contents

1 Assignment

1.1 Purpose of the project

The system to be developed is a social media platform named Quacker. It has all the basic social media functionalities one would expect. Posting quacks (tweets), following other users, a personalized feed/timeline, trends and a basic moderation platform for moderators and administrators.

The purpose of this project is to develop an enterprise-scale application. This means that the focus for this project is mainly on non-functionals such as performance and security. Because of this, the functional requirements are basic in comparison to the non-functionals.

1.2 Purpose of this document

The purpose of this document is to clearly define the project for the student and the stakeholders. This project plan also defines the agreements and strategies for the project. Furthermore, a time plan is made for a rough division of the project. Agreements are made about quality control and the risks and dependencies of the project are described.

1.3 Boundaries

This section describes which products do and which do not belong to the project.

Term	Definition
Quacker	Name of the application
Quacks	Quacker’s version of tweets

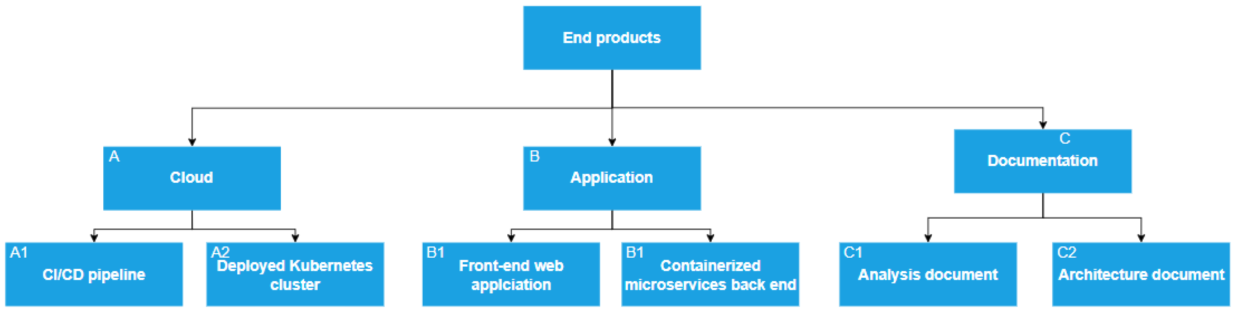
1.4 Strategy

The project will be carried out in an Agile manner. This has been decided to ensure that the product meets the requirements and wishes of the product owner as close as possible. Working Agile means reacting quickly and effectively to changes. This project will be divided in sprints. Each sprint will have multiple iterations for feedback. This allows for quick and iterative reflections on the product to ensure a good final product.

Agile working also means efficient documentation. Documentation such as the analysis, architecture and test documents only contain items that add value to the development of the system. The documentation is also updated per sprint when necessary.

1.5 Final products

Below is a product breakdown structure of the end products that the project will deliver.



2 Activities and planning

2.1 Overall planning

The project is divided into 6 sprints of 3 weeks. Each week, an average of 2 workdays are available to work on the project.

Sprint 0 (week 1 – 3)

The first sprint consists of defining the assignment and a basic setup of the architecture. This means defining functional and non-functional requirements, researching tools and technologies, and designing a basic architecture for the system. A project management tool (Jira) is also set up for the project to keep track of all the requirements.

Sprint 1 (week 4 – 6)

This sprint focuses on the DevOps part of this project. Before implementing functional requirements, a fully working CI/CD pipeline should be up and running. This sprint will most likely require a lot of research and learning about cloud services, CI/CD, Docker and Kubernetes. The first two weeks will most likely be spent learning these technologies. All the new learned skills will be put to use in the final week to set up the pipeline for a simple front-end and back-end application. From here, new features can easily be implemented.

Sprint 2 (week 7 – 9)

This sprint focuses on the microservices architecture of the application. It will require researching and learning about microservices, the different kinds of communication between them, and their health. The purpose of this sprint is to set up a few basic microservices that communicate with each other and monitor their health.

Sprint 3 (week 10 – 12)

This sprint will be used to implement functional and non-functional requirements. These requirements will most likely be implemented per epic. Each functionality will be implemented from the front end to the back end and tested.

Sprint 4 (week 13 – 15)

This sprint will also be used to implement any functional requirements that are left. The focus on this sprint, however, will be on non-functional requirements regarding performance and security. Tests will be executed to make sure that the application is enterprise ready, meaning it should be able to handle many requests and be secure to the most common security risks.

Sprint 5 (week 16 – 17)

The final sprint will be used to implement any requirements that have not yet been finished. Other than that, it focuses on updating documentation and making sure everything is ready for production.

3 Quality assurance, testing and configuration management

3.1 Testing strategy

Different types of tests are performed on both the front end and back end of the application. The back end will mainly be tested by unit and integration tests. Unit tests will be executed for any complex logic within the application. Integration tests will be used to ensure the application works and flows correctly.

Unit and integration tests will be written and executed per microservice. To test communication between microservices, end-to-end tests are written.

These end-to-end tests will be executed from the front end and are executed on a staging server of the application. This means that all databases will contain dummy data, and the production server is not affected.

All tests are executed during the CI/CD pipelines of the application and test coverage is generated from the tests.