Research document

**Student:** Jorn Kersten

**Student number:** 483331

**Class:** DB-S3-DB03

Table of contents

[1. Front-end frameworks 3](#_Toc117154951)

[1.1 Vue.js 3](#_Toc117154952)

[1.2 React 3](#_Toc117154953)

[1.3 Angular 4](#_Toc117154954)

[1.4 Conclusion 4](#_Toc117154955)

[1.5 Sources 4](#_Toc117154956)

[2. Back-end frameworks 5](#_Toc117154957)

[2.1 Quarkus 5](#_Toc117154959)

[2.2 Spring 5](#_Toc117154960)

[2.3 Conclusion 6](#_Toc117154961)

[2.4 Sources 6](#_Toc117154962)

[3. SQL-based vs NoSQL-based 7](#_Toc117154963)

[3.1 SQL-based 7](#_Toc117154965)

[3.2 NoSQL-based 7](#_Toc117154966)

[3.3 Conclusion 8](#_Toc117154967)

[3.4 Sources 8](#_Toc117154968)

[4. Database 9](#_Toc117154969)

[4.1 PostgreSQL 9](#_Toc117154971)

[4.2 MariaDB 9](#_Toc117154972)

[4.3 MySQL 10](#_Toc117154973)

[4.4 Conclusion 10](#_Toc117154974)

[4.5 Sources 10](#_Toc117154975)

# 1. Front-end frameworks

In the sections below I’ll be talking about the advantages and downsides of each front-end framework. Eventually based on the aspects of each framework I’ll chose the one that most fits the implementation within my website.

## 1.1 Vue.js

Vue.js is a progressive framework for JavaScript used to build web interfaces and single-page applications. Besides web interfaces, Vue.js is also used for desktop development with Electron framework and for building apps for Android and iOS. `

**Advantages**

* Detailed documentation to make developing a lot easier.
* Uses components to create a webpage instead of creating everything on a single page. This allows for reusability of components and much easier create a webpage.
* Vue.js is a community driven framework, this allows it to be updated more often and thus it is more modern and used in a lot of projects.

**Disadvantages**

* There is less community support than React and Angular.
* There is a limited number of plugin availability.

## 1.2 React

React is remarkably flexible. Once you have learned it, you can use it on a vast variety of platforms to build quality user interfaces. React is a library, not a framework. Its library approach has allowed React to evolve into such a remarkable tool.

React was created with a single focus: to create components for web applications. A React component can be anything in your web application, this makes React very flexible and easy to use.

**Advantages**

* The principles of SOLID are easier to apply as separation of data and presentation is possible.
* React is an easy to learn library based on JavaScript, so it’s easier for beginners to learn JavaScript.
* A single file contains both the logic and the markup of a page.
* There are a lot of libraries to help assist in creating the project, this helps with creating some functions of the application.

**Disadvantages**

* Implementing a MVC structure is not possible with React, therefore Not all principles of SOLID can be implemented into the project.

## 1.3 Angular

Angular is an open-source front-end framework developed by Google for creating dynamic modern web apps. It uses JavaScript-based TypeScript programming language to eliminate dispensable code and faster apps.

Angular helps build interactive and dynamic single page applications through its features that include templating, two-way binding, modularization, RESTful API handling, dependency injection, and AJAX handling

**Advantages**

* A very wide known and used framework, this makes updates frequent.
* There is a lot of community support on questions or problems about the framework, this makes problems while developing easier to deal with.
* There is the ability to create a large scale application, which makes expandability easier.
* Angular has an well maintained and good documentation.

**Disadvantages**

* Libraries are not very supported by the framework, so it’s more difficult to easily add new libraries to the project.

## 1.4 Conclusion

Based on my research I’ll be using Angular, as it is very developer friendly. It gives the developer a lot of options on linking the front-end with the back-end of the project. Angular also has makes it easy to implement the SOLID principles into the project which keeps code clean and easy to read through. This keeps maintainability easy and not a lot of work.

Also because Angular is used by many companies and developers, I want to learn more about this framework. This gives me more skills and more knowledge and more possibilities when I need to search for work after finishing school.

## 1.5 Sources

Link 1:

<https://javascript.plainenglish.io/angular-vs-react-vs-vue-js-which-is-the-best-choice-for-2022-5ef83f2257ab>

Link 2:

<https://www.altexsoft.com/blog/engineering/pros-and-cons-of-vue-js/>

Link 3:

<https://www.freecodecamp.org/news/why-use-react-for-web-development/>

Link 4:

<https://www.grazitti.com/blog/8-proven-reasons-you-need-angular-for-your-next-development-project/#:~:text=Angular%20is%20a%20highly%20popular,Google%20to%20assist%20web%20development.>

# 2. Back-end frameworks

In the sections below I’ll be talking about the advantages and disadvantages of each back-end framework. Eventually based on the aspects of each framework I’ll chose the one that most fits the implementation within my website.



## 2.1 Quarkus

Quarkus is full-stack, Kubernetes-native Java framework made for Java virtual machines and native compilation, optimizing Java specifically for containers and enabling it to become an effective platform for serverless, cloud, and Kubernetes environments.

Quarkus is backed by Redhat and its quickly gaining ground for creating high-performance, and scalable Java applications. One of the key features of Quarkus based applications is fast boot time.

**Advantages**

* There’s lots of simple documentation online, when problems arise its easy to find a solution. Otherwise a lot of developers are active so help from stack overflow is also possible.
* The application boot time is much faster with Quarkus compared to other frameworks such as Spring.
* Quarkus provides faster hot reloads than Spring Boot since it can automatically detect changes made to Java and other resource/configuration files, and transparently re-compile and deploy the changes.

**Disadvantages**

* The community forum does not necessarily provide solutions to all the problems one may encounter when using Quarkus.

## 2.2 Spring

Spring Boot is an open-source Java framework for developing enterprise applications and microservices. It is used to build production-ready applications using features like auto-configurations and starters. Due to its extensive features, Spring Boot is a commonly chosen application framework for a microservices architecture. Apart from up-to-date and in-depth documentation, there are also a lot of resources available online when it comes to learning Spring.

**Advantages**

* Spring has excellent documentation and outstanding community support. Almost every solution can be found in its community pages.
* Spring provides a template design pattern, allowing developers to include their dependencies in the pom.xml file.
* It is more secure than Quarkus.

**Disadvantages**

* Spring has higher start-up times compared to Quarkus.
* Spring requires a lot of starter modules and contains a lot of other dependencies, which affects overall performance.

## 2.3 Conclusion

Based on my research I’ll be using Quarkus. Based on it being faster and it starting up quicker. Also having Quarkus update on code changes and other changes in files, it makes coding and testing a lot easier while developing an api. Also having good documentation it’s easier to start developing from scratch, though Spring also has good documentation it’s not solay based on this.

Quarkus looks a lot like C# to me which I’m already used to, so stepping into this might be a bit easier than Spring.

## 2.4 Sources

Link 1:

<https://dev.to/javinpaul/5-best-java-frameworks-to-learn-in-2022-for-microservices-and-cloud-native-development-4732>

Link 2:

<https://rollbar.com/blog/quarkus-vs-spring-boot/>

# 3. SQL-based vs NoSQL-based

There are more ways than one to store data in a database, in this section I’ll be researching what way to store data is the best way. At the end of this section I’ll be choosing a way of storing data, in section [4. Database](#_Database) I’ll cover what database I will use for keeping my data.



## 3.1 SQL-based

Relational database management systems use SQL, a database management language that offers a organized and structured approach to information management and storage. It stores data by specific columns like a name, a number or maybe an value like true or false. Using for each line of data, relational databases apply strict, categorical parameters that allow database users to easily organize, access, and maintain information within those parameters.

**Advantages**

* SQL-based databases are highly stable and reliable, if errors occur the given data in a database might not be correct. The database itself will always work as it should. This means an SQL-based database is a very ACID compliant system.
* The data stored in a SQL-database is always consistent as the data is stored by identifiers such as name, or date.
* Because Relational database management systems have been around for over 40 years, it's easier to get support, add-on products, and integrate data from other systems. On top of this there are a lot of database managers that are SQL-based vs that are NoSQL-based.

**Disadvantages**

* Scalability into adding new columns onto an existing table is more difficult. So having old data and adding a new property isn’t as easy.

## 3.2 NoSQL-based

When a system uses or creates a large amount of unstructured data, such as text from emails or surveys. The information within these texts are unorganized and difficult to put in assigned columns, the way a regular SQL-based databased would store them. But you can store it with a NoSQL-based database system. NoSQL-based databases let you organize information in a looser fashion, kind of like dropping the information in different file folders.

**Advantages**

* NoSQL-based databases are excellent for storing information that isn’t usually the same, or are very large texts.
* There are no limits on the length of the information that is being stored. This makes it easier to store a lot of different types of length text.
* As for a NoSQL-database the length and columns don’t matter, it’s a lot easier to scale this kind of database without much difficulty.

**Disadvantages**

* Because the NoSQL community doesn't have 40 years of history and development behind it, it could be more difficult to find experienced users when you need to troubleshoot.
* Since the system is relatively new compared to SQL-based RDBMS solutions, there aren't as many database managers that are NoSQL-based.
* the data in your non-relational database management system doesn't readily integrate with other products and services.

## 3.3 Conclusion

When comparing the two types of data storage for my application a SQL-based database is need. The data that I’ll be storing in my project is definable into different columns. On top of this the data won’t be extremely large, so the lengths used in conventional SQL-based databases are plenty long enough.

Because SQL-based databases have been around for this long, it will be easier to use and find help with these kinds of databases.

## 3.4 Sources

Link 1:

[https://www.integrate.io/blog/which-database/](https://www.integrate.io/blog/which-database/%20)

# 4. Database

In the sections below I’ll be talking about the advantages and disadvantages of three different databases. Afterwards I will pick a database based on the advantages and disadvantages I’ve found of the three databases.



## 4.1 PostgreSQL

PostgreSQL is an open-source, free database engine with unlimited scaling capabilities. PostgreSQL supports both relational and non-relational data formats.

An interesting feature of PostgreSQL is its history of working with both structured (SQL) and unstructured (NoSQL) data. PostgreSQL doesn’t simply store information to identify tables and columns. It allows you to define data types, index types, and functional languages. It's also compatible with most operating systems, including Linux platforms, and it integrates well with data from a wide variety of databases. PostgreSQL also works with, both, on-site servers and cloud-based servers.

**Advantages**

* PostgreSQL has a lot more features than other database management systems. These extra features include table inheritance, a rich set of data types, ability to define a column as an "array" of column types, among others.
* PostgreSQL can work with massive database tables, this makes it easy to scale up the database massively if its needed.

**Disadvantages**

* PostgreSQL doesn’t have the best documentation. This means if you run into problems it’s difficult to quickly find an solution to your problem. A few options would be to submit you problem or find help with community support.

## 4.2 MariaDB

MariaDB is a fork of MySQL. The developers built the relational database management system to preserve MySQL’s structure and features. They feared that the system’s acquisition by would jeopardize the database.

MariaDB’s developers make sure that each release is compatible with the corresponding version of MySQL. MariaDB not only adopts MySQL’s data and table definition files but also uses identical client protocols, client APIs, ports, and sockets. The goal is to let MySQL users switch to MariaDB hassle-free. Much like MySQL, MariaDB is modifiable using SQL statements.

**Advantages**

* MariaDB offers a better query execution so finding and processing data into and from the database is much easier. On top of this, it’s great at handling large data sets.
* As MariaDB can handle large data sets it’s easy to scale up the database if its needed.
* MariaDB is faster in running it’s query’s therefore it has a high performance when updating or searching data in tables.

**Disadvantages**

* When encountering problems there aren’t a lot of options to getting help. For professional help users have to pay. Another option is finding help from the community.

## 4.3 MySQL

MySQL is a free, open-source RDBMS solution that Oracle owns and manages. Even though it’s freeware, MySQL benefits from frequent security and features updates. Large enterprises can upgrade to paid versions of MySQL to benefit from additional features and user support.

**Advantages**

* MySQL is highly compatible with other systems, it is also very compatible with many other database systems. This makes testing with other databases very simple.
* MySQL is a free database manger that allows a lot of features for the user. This makes it so when encountering problems there is a lot of useful help online.

**Disadvantages**

* Because MySQL prioritizes speed and agility over features, you might find that it’s missing some of the standard features found in other solutions. For example, the ability to create incremental backups.

## 4.4 Conclusion

Based on the research I’ve done, I’ll be going with MariaDB. MariaDB brings a bit more than MySQL does content wise, but doesn’t bring the extensive and big features PostgreSQL brings. As I just need a simple database PostgreSQL’s features aren’t needed in my case.

Therefore MariaDB is more than sufficient for my case, and because it’s faster than MySQL it’ll do just fine.

## 4.5 Sources

Link 1:

[https://www.integrate.io/blog/which-database/](https://www.integrate.io/blog/which-database/%20)

Link 2:

<https://cloudinfrastructureservices.co.uk/mariadb-vs-mysql/>

Link 3:

<https://www.hostinger.com/tutorials/mariadb-vs-mysql>