**software characterisation research**

**Student:** Jorn Kersten

**Student number:** 483331

**Class:** DB-S3-DB03

Table of contents

[Introduction 3](#_Toc120616851)

[1. Front-end frameworks 4](#_Toc120616852)

[1.1 Vue.js 5](#_Toc120616853)

[1.2 React 6](#_Toc120616854)

[1.3 Angular 6](#_Toc120616855)

[1.4 Conclusion 6](#_Toc120616856)

[1.5 Sources 7](#_Toc120616857)

[2. Back-end frameworks 8](#_Toc120616858)

[2.1 C# 8](#_Toc120616859)

[2.1 Java 8](#_Toc120616860)

[2.3 Main differences between C# and Java 8](#_Toc120616861)

[2.3.1 Overall about C# and Java 8](#_Toc120616862)

[2.3.2 Performance differences 9](#_Toc120616863)

[2.3.3 Syntax differences 9](#_Toc120616864)

[2.3.4 Security differences 9](#_Toc120616865)

[2.3 Conclusion 9](#_Toc120616866)

[2.4 Sources 10](#_Toc120616867)

[3. SQL-based vs NoSQL-based 11](#_Toc120616868)

[3.1 SQL-based 11](#_Toc120616870)

[3.2 NoSQL-based 11](#_Toc120616871)

[3.3 Conclusion 12](#_Toc120616872)

[3.4 Sources 12](#_Toc120616873)

[4. Database 13](#_Toc120616874)

[4.1 PostgreSQL 13](#_Toc120616876)

[4.2 MariaDB 13](#_Toc120616877)

[4.3 MySQL 14](#_Toc120616878)

[4.4 Conclusion 14](#_Toc120616879)

[4.5 Sources 14](#_Toc120616880)

# Introduction

This document was created to keep track of my researches on my software characterisation. By this I mean the researches about which front-end frameworks I can use, the programming languages I can use for my back-end. But also what type of database I want to use, and in addition what kind of database I want to use then.

The structure of this document is as follows. The first chapter is my research on my front-end framework. This research starts with the popularity between the three frameworks I have chosen. I then list all the pros and cons for each framework. All this is followed by a conclusion in which I make my choice and substantiate it.

Next, I research the back-end programming language, the structure of which is slightly different. I first tell something about the languages, with both a short description followed by the features they have. Then I look at the differences between the two, followed by the conclusion. In it, I make my choice and substantiate it.

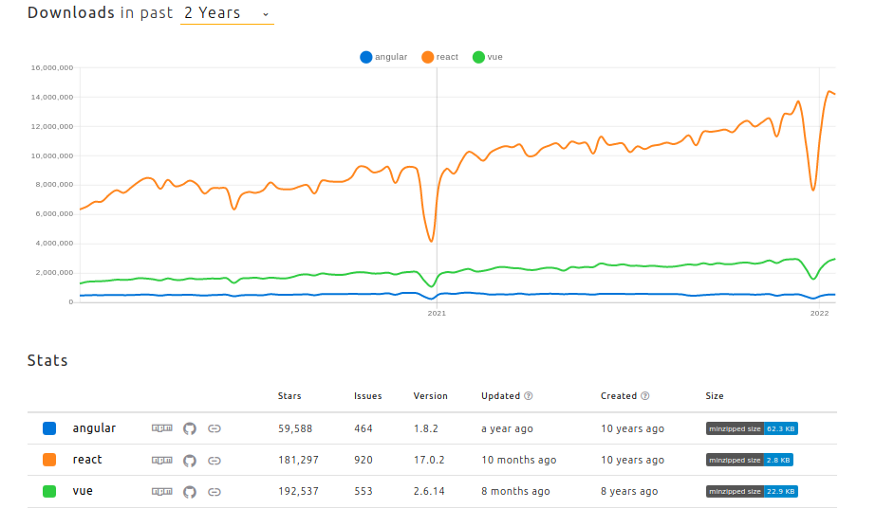
Finally, I compare database types, writing pros and cons for each type. After this comes the conclusion in which I make my choice, and again I substantiate my choice. The next chapter compares three databases I could use for my application. Here I again talk briefly about the database, then I write down the pros and cons for each database. This is followed by a conclusion. In it, I make a choice based on my project and justify the choice.

# 1. Front-end frameworks

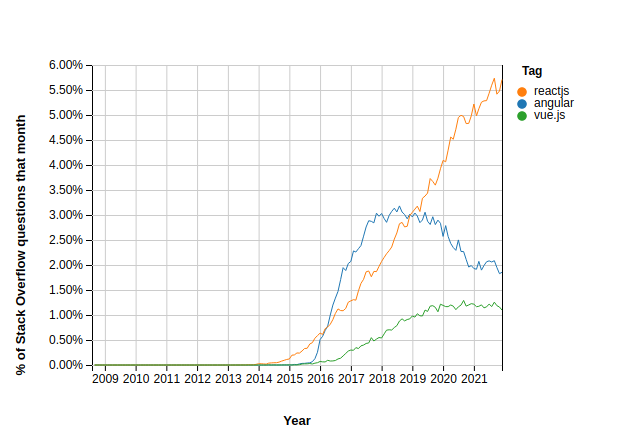
In the sections below I’ll be talking about the advantages and downsides of each front-end framework. Eventually based on the aspects of each framework I’ll chose the one that most fits the implementation within my website. Having no knowledge of JavaScript, I have yet to learn about the different frameworks out there.

The chart below shows the downloads of the three frameworks: React, Angular and Vue.js in the last two years. This result was revealed from a survey by npm, a package manager to easily install frameworks and libraries.

Overall, React is the most downloaded framework in the past two years. This makes it the most used framework of the three. It has been downloaded about 14 million times in this year. This is followed by Vue.js, which is the second most downloaded with around 3 million downloads. And last comes Angular with about 750 thousand downloads.



In contrast, if you look at the questions asked on Stackoverflow, this shows you something quite different. This shows that React is still the most used with a 35.9% usage rate by developers, but in Vue.js's place now comes Angular. Angular now comes second with 25.1% used by developers, and so last comes Vue.js with 17.3% used by developers.



## 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 the charts and my research, I choose Angular. This is still widely used in the market, when I have questions about it I can easily find it on the internet. It is also easy to use SOLID principles in this, as the layers are mostly spaced apart. In addition, because the layers are split, it is easy to link the front-end with the back-end.

Furthermore, since Angular is widely used by many companies and developers, I think it is a good choice to learn this framework. This will broaden my knowledge and skills within this profession, this will give me more choices later on when finding a job.

## 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 will talk about the advantages and disadvantages of the languages C# and Java. From this, I want to know which is easier to use for creating my api.

## 2.1 C#

**What is C#**

Microsoft developed C#, also known as C-sharp, in 2000. It is object-oriented, component-oriented, lexically, scoped, generic, and a strongly typed programming language.

C# provides modern features and simple codes, making it a real game-changer. This language keeps on updating from time to time which allows for numerous integrations and contemporary elements.

**C# features**

* C# is a part of Microsoft Visual Studio. This makes it easier on a Windows laptop as Microsoft Visual Studio is part of Windows.
* Modern features are part of C# has it is powerful, robust, and scalable per the contemporary trends.
* It follows OOP concepts such as inheritance, abstraction, polymorphism, encapsulation.
* C# allows the use of VB NET components, this makes adding libraries easy for a user.

## 2.1 Java

**What is Java**

Java was founded in 1995. ‘Write once Run Anywhere’ is its motto that makes Java an independent language platform. It has the advantage of in-users experience as users can rewrite or reuse a code to build a new code.

**Java features**

* Easy to learn and simple syntax.
* Java follows OOP (object-oriented programming) concepts such as inheritance, abstraction, polymorphism, encapsulation.
* It offers Exception Handling which makes datatype handling easier.
* With the help of JRE (Java Runtime Environment), Java allows its applications to run on any platform by using its WORA functionality.
* Java uses a just-in-time compiler to execute high-performance coding.

## 2.3 Main differences between C# and Java

### 2.3.1 Overall about C# and Java

Both C# and Java were created by keeping C and C++ languages in mind, as they have similar syntaxes. C# uses CLR (Common Language Runtime), while Java uses JRE (Java Runtime Environment). It is also object-oriented, functional, strongly typed and component-oriented, while Java is only object-oriented.

C# supports operator overloading for multiple operators. Java still does not support operator overloading and pointers. C# arrays have the specialisation of System, while Java arrays have the occupation of Objects.

### 2.3.2 Performance differences

With both C# and Java, the idea behind the languages was good performance. C# and Java both use a Just-in-time compiler for that. A Just-in-time compiler uses less of the CPU, putting less load on the system and making it more responsive.

Relative to C#, Java also delivers high performance, but Java uses much more of the memory. This makes C# more efficient than Java and thus a better choice based on performance.

### 2.3.3 Syntax differences

Because both C# and Java are based on C and C++, both languages share a similar syntax. They both have the property of being object-oriented, which is derived from C++. Therefore, it is in both languages.

Nevertheless, there are some differences. A well-known one among C# developers is that they can use properties, in contrast, Java needs get/set methods instead of properties. Another is that Java has built-in annotation processing that C# lacks.

### 2.3.4 Security differences

Java has solid security. Because of its statistical typesetting, it minimises type-related errors. Due to Java's automatic code verification, errors are found before the application is built. As a result, the user finds errors in code faster and does not have to build the entire application first. However, Java is worse when integrating external libraries and applications.

On the other hand, C# is more vulnerable to many threats. Especially SQL and CMD injections are the culprit here. Few features of C# are to some extent the cause of these security problems. This does not mean that C# cannot secure the system it runs on. It has good protection in that respect, but it some parts are missing because of this.

## 2.3 Conclusion

Based on what I have found, I choose Java. Most of its advantages, especially its security advantages, are very useful when creating an api. I already have knowledge of C# and would like to learn something new as well. Since Java is very similar to C#, it is not very difficult to start learning this yet. We also get a demo on Quarkus, which is a framework on Java, which also helps when learning a new language.

I would like to keep my code to SOLID principles. This is possible with both languages, as they both implement Object oriented programming.

On top of that, a lot can be found on the internet about both, so when questions come up I can always find something.

## 2.4 Sources

Link 1:

<https://distantjob.com/blog/c-vs-java/>

# 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>