Design document

Individual Project Semester 3

Mihail Vasilev

Date: 07.10.2022

# Overview

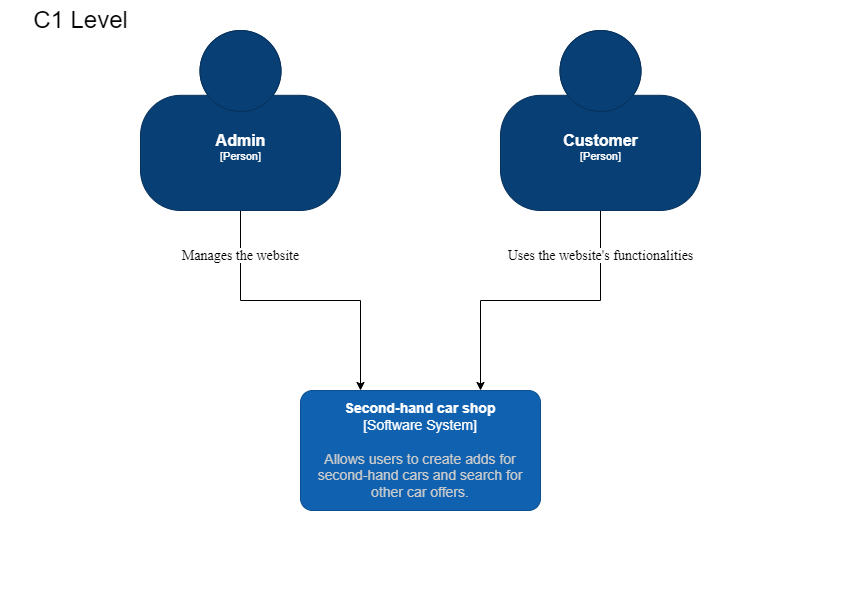
The purpose of this document is outlining the design, design decisions and application architecture of the individual project. The project goals are to create a website application that allows users to make and find adds about second-hand cars. The website will provide features that help when someone wants to find a particular car of their interest.

# Architecture

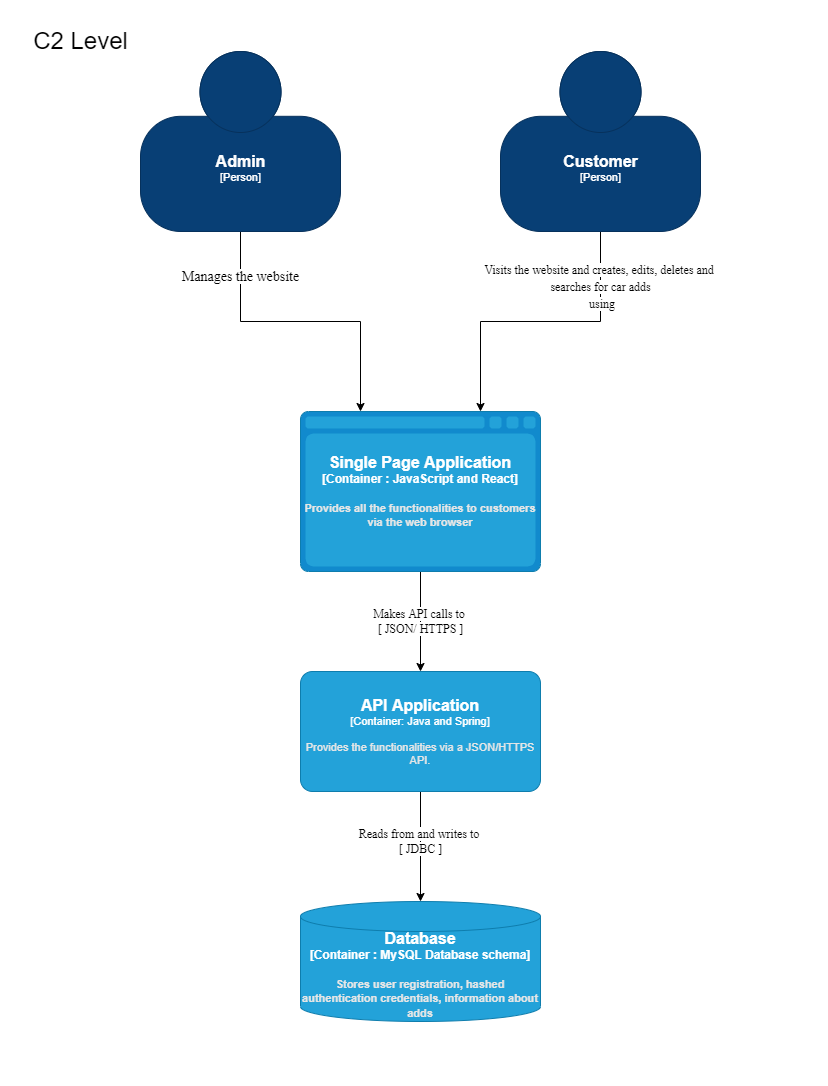
## S.O.L.I.D.

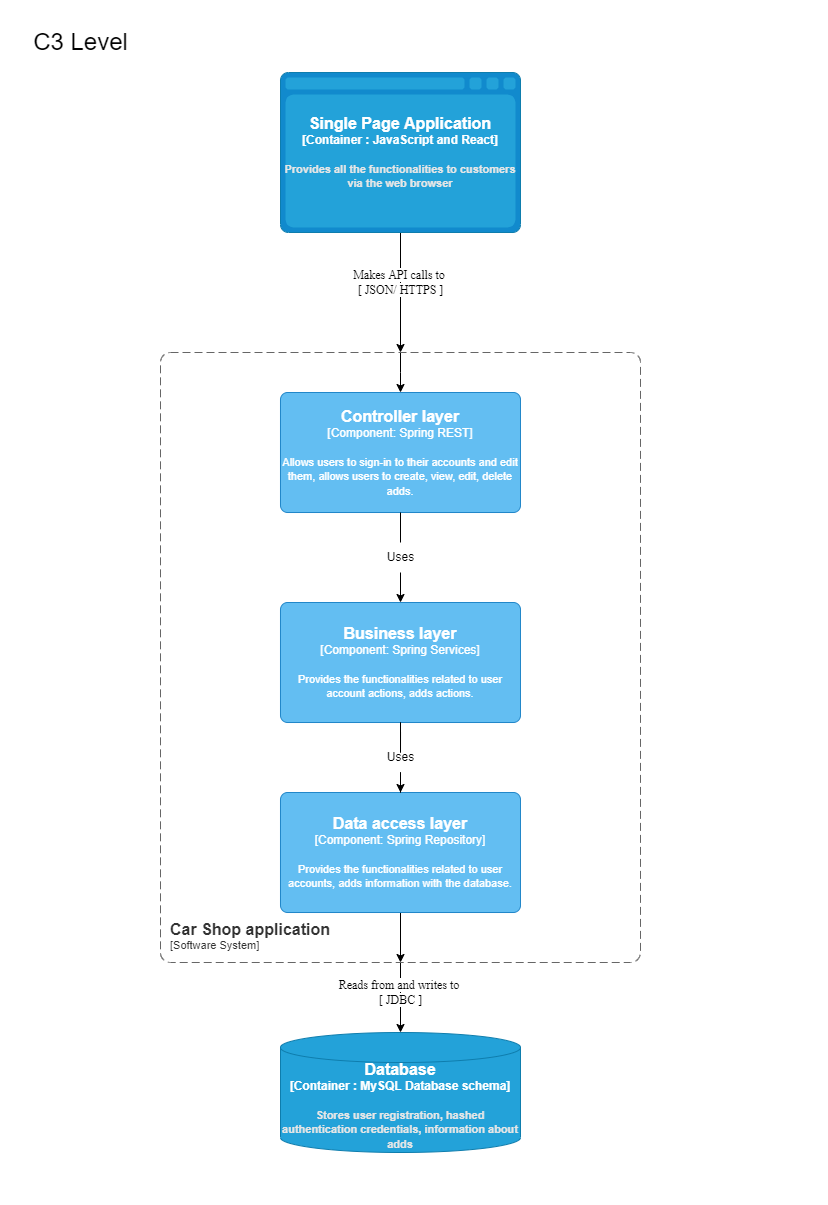
* The single-responsibility principle: Every service class has only one responsibility. There are different classes for the different tasks that need to be executed. This way testing and code management is much easier. All classes in the project follow this principle.
* The open–closed principle: Each entity can be extended and upgraded in the future if it is needed. This way the features stay open for extension and at the same time closed for modifications.
* The Liskov substitution principle: Classes that are subtypes of other classes can be interchanged without breaking the program.
* The interface segregation principle: There is an interface for each use case. This way each interface is responsible only for one thing. The code stays maintainable and easier to fix in case of bugs.
* The dependency inversion principle: Interfaces are used when specific actions are called from the controllers instead of the service classes. This way the program relies on abstractions and not concretions. This also allows for better testing.

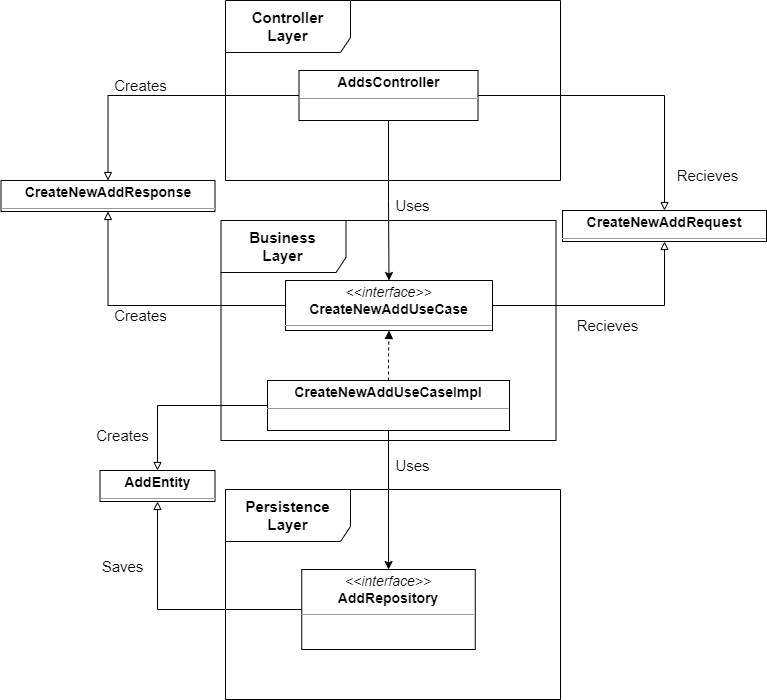
# C4 design diagram



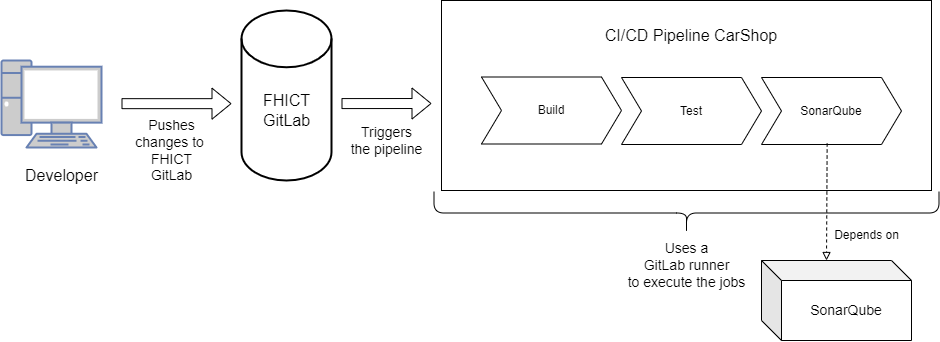
The C1 diagram shows the interaction between the users and the application. There are two types of users: Admin and Customer. The admin can do different actions with the application and manages the application.

The C2 diagram show the main containers that the application is made of. It also shows the technologies that are used for each container. The application is split into 3 main parts – Single Page Application (JavaScript and React), the website that the users interact with, API Application (Java and Spring), the part which takes care of the functionalities for the website, Database (MySQL database), the place where the data is stored.

The C3 diagram shows the main components that the containers are made of. The API Application is divided into 3 layers: Controller layer, which takes care of the HTTPS requests, Business layer, which takes care of the logical parts of the functionalities, Data access layer, which provides the connection with the database.

C4 Level

The C4 diagram show the main layering of the project and the way the elements are connected and interact with each-other. In the Controller Layer stay all the controller classes of the application which deal with the REST services. They then send the acquired requests that they received to the Business Layer of the application where the main logical functions are kept. Use cases like saving, deleting and getting specific elements are situated here. This is where the dependency inversion principle can be most clearly seen. Then these methods send the required data to the Persistence Layer which is the layer responsible for connecting to the database.

CI/CD Pipeline Setup

The above diagram shows the structure of the setup of my GitLab CI/CD pipeline. When new changes are pushed to the FHICT GitLab repository, GitLab triggers a runner, which in my case runs on my local machine, that executes the jobs defined in the .yml file of the application. The first job is to build the application. The second job is to execute the unit tests of the application. The third and last job is the SonarQube quality assurance tests which connects to SonarQube, again running on my machine, and analyses the source code. After that a report on the quality of the code in the project is created. With this the CI/CD pipeline is finished. If any of the jobs fail at some point the pipeline “fails” and an email is sent to me immediately.

# Design decisions

# Front-end

For the front end I have decided to use React and JavaScript since they are very popular and proven technologies for building user interfaces based on UI components. They provide a lot of ways to create and maintain a good and user-friendly interface. React allows websites to be more flexible and efficient by rendering only the specific components when data has changed which creates both a better user experience and also is easier to develop. React is perfect for our single-page website application. (Technostacks, 2022)

# Back-end

For the back-end I have decided to use Java and Spring Boot. Java is a world-wide used programming language that has many advantages and environments that make it easy to learn, use and find efficient solutions to many programming problems. It is an object-oriented language which is what is needed for the current project. Java is easy to learn and understand, it is platform-independent and provides secure ways to work with data. Spring Boot is a wide set of libraries that allows developers to create almost anything that they might need. It is widely used by millions of users every day, it is flexible and has tools which provide many options to the developers to create fast, secure and convenient applications. Because of these reasons I have decided to use Spring Boot. (SCAND, 2020)

# Database

For the database I have decided to use the MySQL database since it is easy to use and provides a lot of functionalities and flexibilities both to users as well as to developers. It is fast, reliable, secure and scalable which means that the data stored will be easily accessible to users while at the same time kept securely from outside threats. Because of MySQL’s features and ease of use I have decided to have it as the main database of my project. (Castro, n.d.)

*Used methods:*

1. Available product analysis – I had to research what the options to create an application such as mine are. I wanted to see what were the best choices and how to get the most of their functionalities.
2. Community research – I had to visit forums and research what are the good and bad practices as well as get more information about the capabilities and actual uses of each of the technologies.
3. Document analysis – reading and researching the documentation is a good way to research the specifics and go a bit deeper into the technologies.

# Conclusion

The project follows the main SOLID principles to ensure an efficient, easy to maintain and update software and provide the best services possible. The technologies that I have decided to use for this project will allow me to create a reliable, fast, secure and user-friendly website application that customers will have a pleasant time experiencing.

# References:

1. Technostacks. (2022, September 19). 10 best frontend frameworks in 2022: Technostacks. Technostacks Infotech. Retrieved November 3, 2022, from <https://technostacks.com/blog/best-frontend-frameworks/>
2. Castro, S. (n.d.). 5 Reasons Why MySQL Is Still The Go-To Database Management System. <https://www.jobsity.com/blog/5-reasons-why-mysql-is-still-the-go-to-database-management-system>
3. Mihalchenko, A. (2020, June 26). Pros and Cons of Using Spring Boot. SCAND. https://scand.com/company/blog/pros-and-cons-of-using-spring-boot/