Semester 3 Individual track project

S3-S-CB-03

Onea Bianca

Individual Track  
Design Document

October 7, 2021

Contents

[1. INTRODUCTION 2](#_Toc88841369)

[2. System overview 2](#_Toc88841370)

[3. DESIGN CONSIDERATIONS 3](#_Toc88841371)

[FRONTEND 3](#_Toc88841372)

[BACKEND 3](#_Toc88841373)

[DATA STORING 3](#_Toc88841374)

[4. C4 ARHITECTURE DIAGRAMS 5](#_Toc88841375)

[C1- System context 5](#_Toc88841376)

[C2- CONTAINERS 6](#_Toc88841377)

[C3- COMPONENT DIAGRAM 7](#_Toc88841378)

[C4 – CLASS DIAGRAM 8](#_Toc88841379)

[5. CI SETUP 9](#_Toc88841380)

[6. REFERENCES 9](#_Toc88841381)

# INTRODUCTION

For this semester’s individual project, I am required to develop a full stack web application, using the following techniques and principles: JavaScript frontend, REST API service layer and a secure backend with a database access. I decided to create a hotel booking website, where the users can search for hotels, make reservations and display their own hotel to the website.

# System overview

The web application I will create is a hotel booking website which is dedicated for all the people who want to search for a hotel, reserve a room or display their property on the website. The website will have two displays: one for the users and one for the admins.

The users are going to be able to see all the available hotels, search for a specific hotel and make a reservation by inputing the destination they want to go, the check in and check out date and the number of adults and children. They also can contact the admins of the hotel if they have any problems or inquiries, change the provided personal information when they sign up or apply for displaying the hotel on the website.

The admins are going to be able to see all the reservations which were made through the website, see the statistics about the hotels and the registered users. They will also be in charge of accepting/declining the applications from the users regarding the display of their hotel.

# DESIGN CONSIDERATIONS

## FRONTEND

For the individual project, I decided to make use of the React framework, because there is faster development and I can achieve a better user experience and a very fast performance. Furthermore, the possibility of creating reusable components helps me keep my code clean, structured and it eliminates the need to write a lot of repetitive code.

Choosing the framework I want to use for my individual project has been a true challenge because every framework has its advantages and disadvantages and its own popularity, the debate about choosing the framework that suits best the requirements of my project being heated.

For choosing which framework meets best the criteria and requirements of the project, I decided to make use of the ” Multi-criteria decision making” DOT Framework method in order to see which front-end framework is the most suitable for me, based on my criteria: the learning curve, popularity and the connection to the back-end.

## BACKEND

Considering the fact that one of the project’s system’s requirements was to use the Java programming language and the Spring Boot framework, the DOT Framework method I am using to justify my decision is ”Requirements prioritization”, because it is an essential stage in requirements engineering, particularly in agile software development.

## DATA STORING

The database for storing my project’s data plays an important role in developing my product, so I gave a lot of consideration to this viewpoint. To determine which database is the easiest to use, Having in mind that my choice of the database has to be easy to use, has to have an increased level of security and the connection between it and my backend has to be as smooth as possible, the „Community research” DOT Framework method was the most suitable for making a decision.

After conducting my research, I came to the conclusion that MySQL is the database that suits best the requirements of the project and my needs, having strong data protection, simple management, and high performance.

With the help of these tools, I will be able to follow and apply some of the principles that I’ve learned so far, such as SOLID principles.

1. Single-Responsibility Principle

This principle is a good way to identify the classes that are nedeed in the project and structure them in a way that every class has one reason to change. For my project, I decided to have 5 classes: User, Admin, Hotel, Room and Offer. It is possible to add more classes in the future, so the structure is not final yet.

1. Open-Closed Principle

For applying this principle, I am going to make use of interfaces, so when I add new functionality to a class, there will be no need to change the existing functionalities. As an example, the Controller classes, which are used for the comunnication between the backend and the frontend, have one interface each, so when I want to add a new method to the classes, I don’t have to change them, because the changes happen in the interface and the class inherits from it.

1. Liskov Substitution Principle

Because I am not sure I am going to need or use inheritance for this project, the LSP will not be applied yet.

1. Interface Segregation Principle

Because I am going to make use of interfaces for the Controller classes, I decided to separate the concerns and create an interface for each Controller class, instead of having one big interface: UserController class with iUserData interface, AdminController class with iAdminData interface, HotelController class with iHotelData interface, RoomController class with iRoomData interface and OfferController class with iOfferData interface.

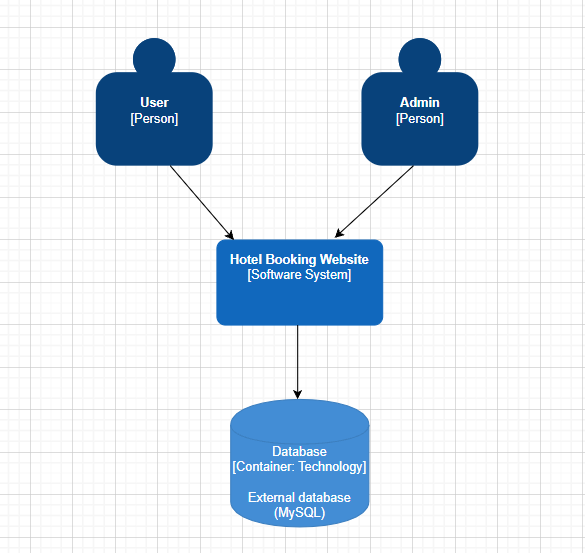
1. Dependency Inversion Principle

For applying this principle, it’s only needed to apply the Open-Closed Principle and the Liskov Substituition Principle, because the classes also comply with the Dependency Inversion Principle. As I wrote above, the OPC is applied and the LSP will probably implemented.

# C4 ARHITECTURE DIAGRAMS

The following diagrams represent the third version of the C4 Architecture diagrams, so additional functionalities will be added during the next sprints, as I will implement more features.

## C1- System context

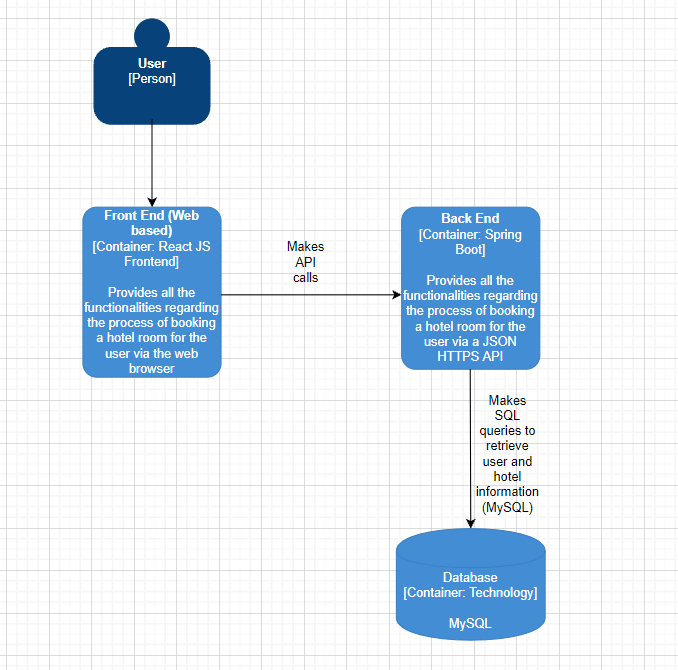


There are going to be two types of people who can access the website: the user and the admin, each one of them having a different view.

The user is going to be able to see the public content, send messages to the admins through the contact form, create an account and log in. Furthermore, if the user has an account and is logged in, he/she will be able to apply for displaying their hotel on the website, see their personal information and make reservations through the website.

The admin is going to be able to see the statistics and overview of the registered users, displayed hotels, reservations made through the website, etc, will be able to manage the personnel and answer the messages the users sent.

## C2- CONTAINERS



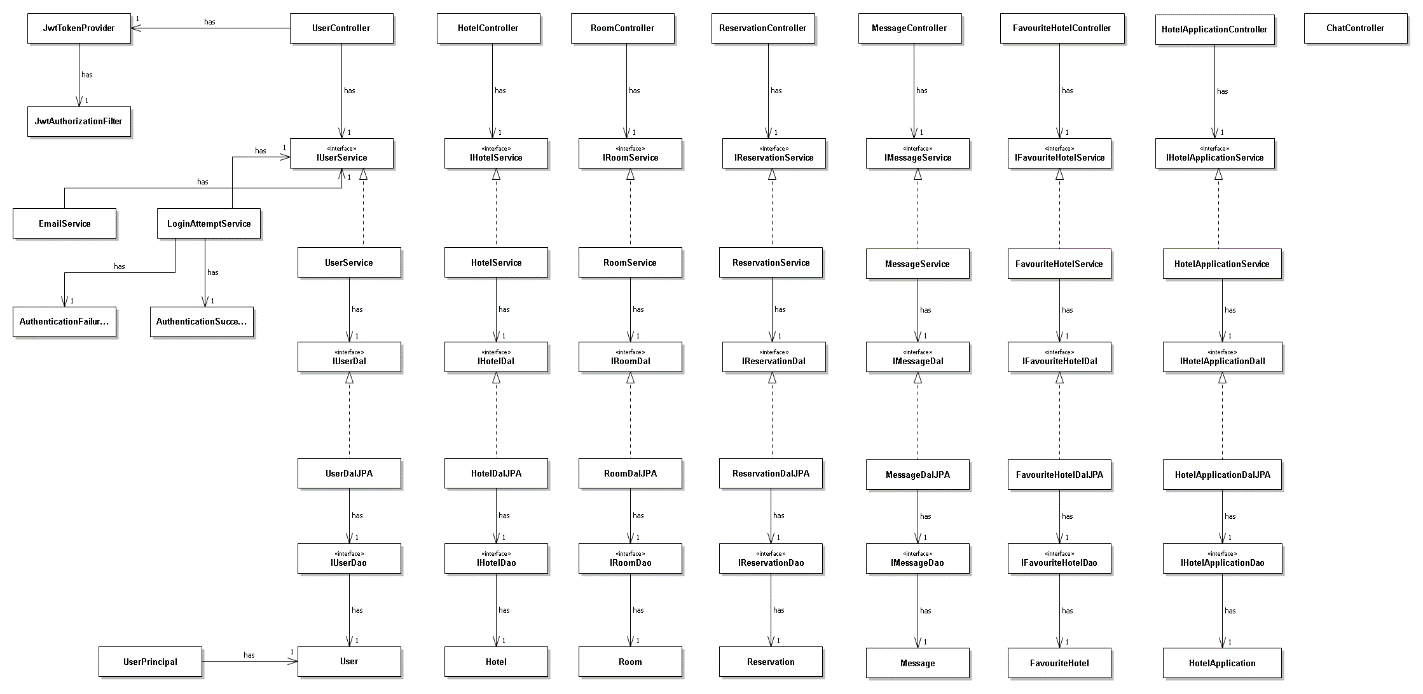
The web application is devided into three main containers: the frontend (JavaScript, React JS), which represent the user interface and provides all the functionalities of the website, the backend (Java, Spring Boot), which is responsible for retrieveing data from the database and communicating with the frontend using HTTP requests and the database (MySQL), which stores the stores the data, such as the users, hotels, reservations, etc.

## c3- component diagram



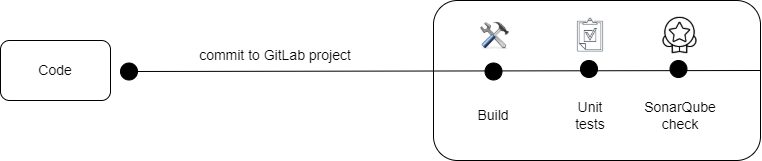
These are the components which help me form the backend. This structure helps me to separate concerns, to follow the SOLID principles and to implement gthe dependency injection.

## C4 – CLASS DIAGRAM



# CI SETUP

The following diagram shows how the CI is set up in my GitLab repository.



Before pushing the code to the GitLab project, there are some updates that have to be done in order to ensure the successful building, testing and SonarQube checking. In my case, I had to make a few changes to the .git-ci.yml file regarding the scripts for the build, the unit tests and the SonarQube check, so the paths for the backend code match and the jobs to be done correctly.

# 6. REFERENCES

Methods - ICT research methods. (n.d.). Methods - ICT Research Methods. <https://ictresearchmethods.nl/Methods>

Top Reasons to Use MySQL. Top Reasons to Use MySQL. <http://www.databasequest.com/index.php/product-service/mysql-dbquest>

AbdElazim, K., Moawad, R., & Elfakharany, E. (2020, February 1). A Framework for Requirements Prioritization Process in Agile Software Development. ResearchGate. <https://www.researchgate.net/publication/340120762_A_Framework_for_Requirements_Prioritization_Process_in_Agile_Software_Development>