**Design Document**

***<<Warehouse project>>***

*<<Warehouse & CO>>*

|  |
| --- |
| **Date : 27.11.2020** |
| **Version : 2.1** |
| **State : Sprint 4 Final version** |
| **Author : Bojidar Balabanov** |

#### Version history

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author(s)** | **Changes** |
| 0.1 | 01.10.2020 | B.Balabanov | First draft |
| 0.2 | 02.10.2020 | B.Balabanov | Added front-end justification |
| 1.1 | 13.10.2020 | B.Balabanov | Added back-end justification, updated data base design |
| 1.2 | 28.10.2020 | B.Balabanov | Added C4 Model diagrams |
| 2.1 | 25.11.2020 | B.Balabanov | Updated C4 component diagram |

Contents

[1. Front-end framework 3](#_Toc61813643)

[2. Back-end framework 4](#_Toc61813644)

[3. System context diagram 5](#_Toc61813645)

[4. Container diagram 6](#_Toc61813646)

[5. Component diagram 7](#_Toc61813647)

[6. Class diagram- Product Entity 8](#_Toc61813648)

[7. Database design 9](#_Toc61813649)

# Front-end framework

For this project I chose to work with Angular.



* 1. Justification. Why Angular?

I choose Angular to be the front-end framework of this project because in the group project I am the only one who does not know how to operate with it.

I read a few statistics and everything pointed towards React, because it has the highest popularity out of the big three, but despite being the least popular and the oldest Angular actually has good support and the job market actually requires as much Angular developers as React developers with Vue being in last place.

# Back-end framework

For this project I chose to work with Java and Spring boot

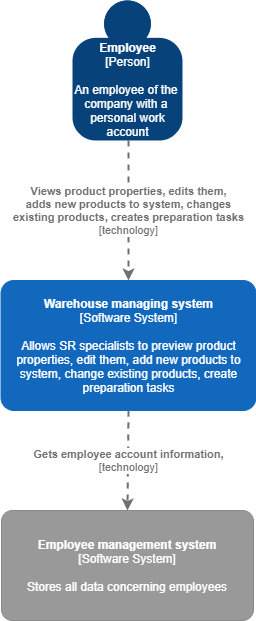
* 1. Justification. Why Java?

I chose Java to be the language for my back-end system because it was recommended to me by the university and it is fairly similar to C# with which I already have experience.

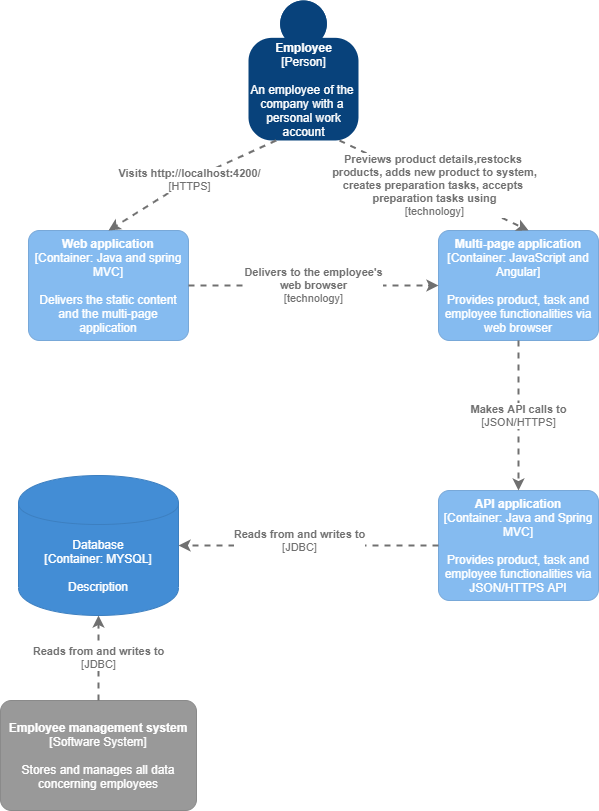
* 1. *Justification. Why Spring boot?*

At first, I tried to work with jersey instead of Spring, but I encountered difficulties and could not solve them, so I decided to redo what I had up to that moment and this time use Spring. I personally find spring as a very good choice, because the tutorials on the internet explain better and there is more up to date information about it. Spring also does pretty much on its own and makes the process of creating an API much more pleasant and fast.

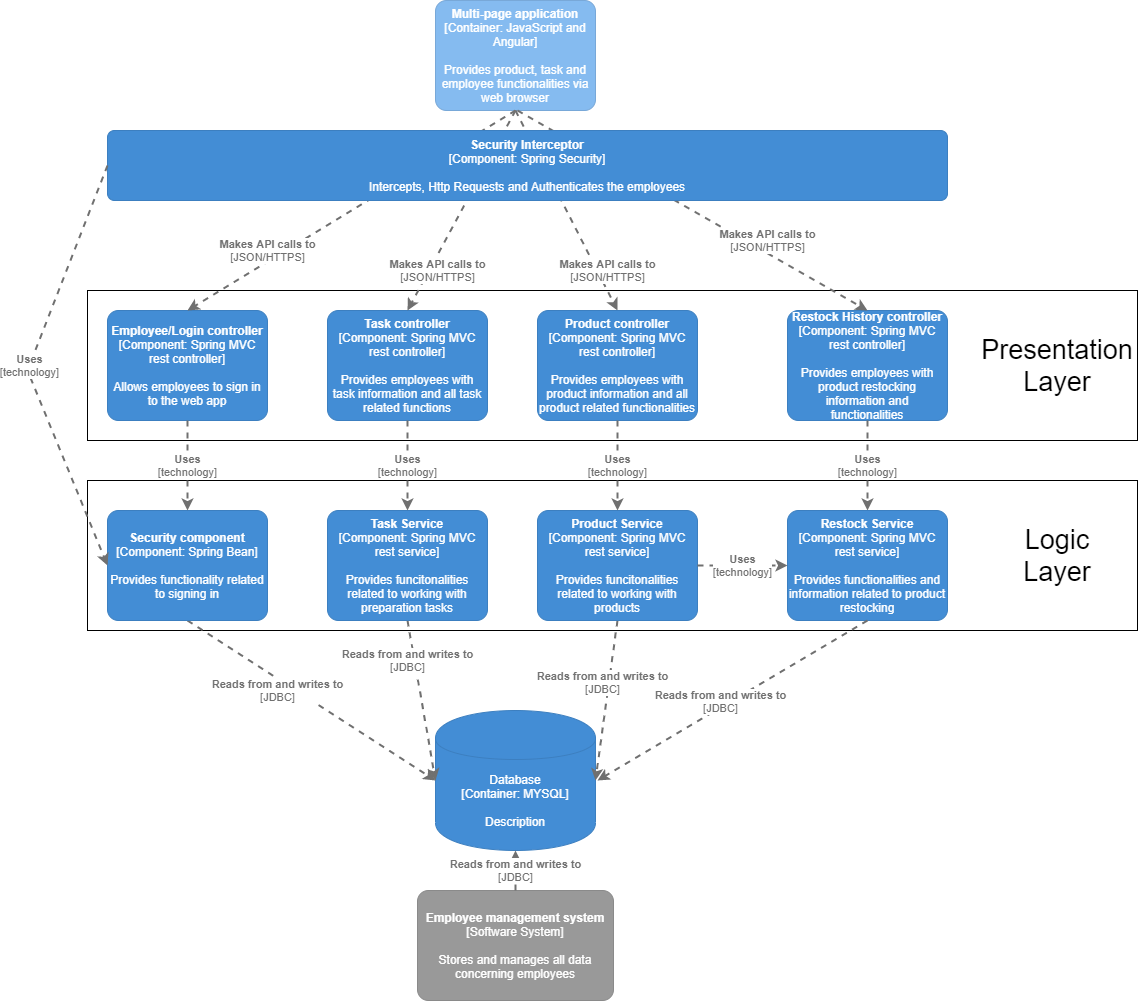
# System context diagram



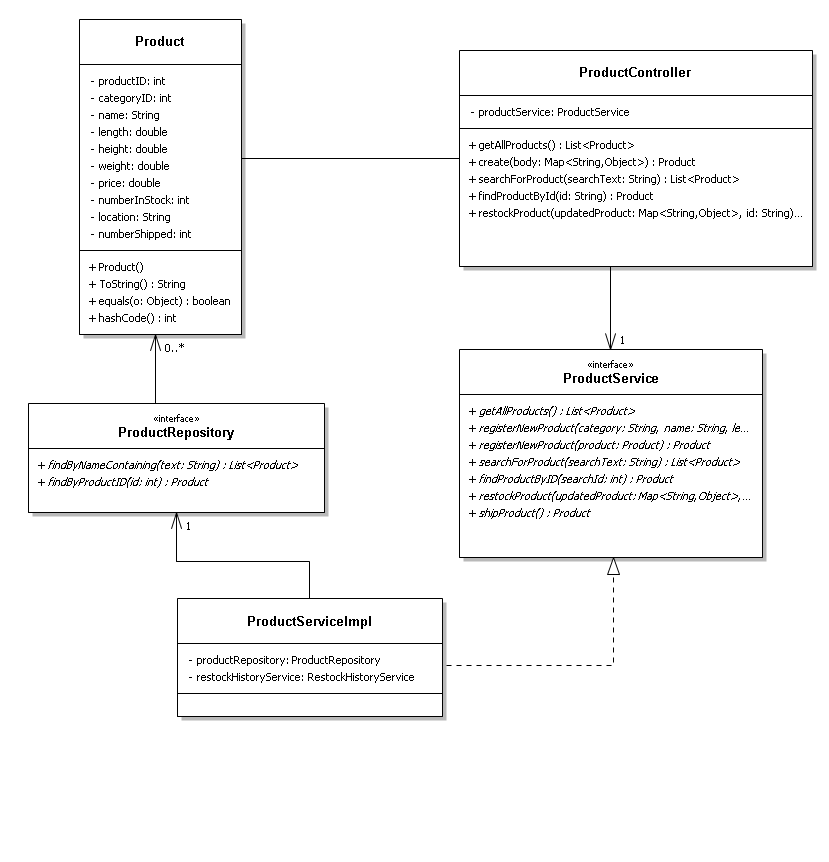
# Container diagram



# Component diagram



# Class diagram- Product Entity



# Database design

The following diagram shows the relationships of entity sets which are going to be stored in a database.

A category can have zero or more products, while a product can have only one category.

A product can have zero or more restock history entries, while a restock history entry can have only one product.

An employee can have only one role in the system, while a role may be assigned to zero or more employees within the system.

An employee can be responsible for zero or more restock history entries, while a restock history entry can have only one employee responsible for it.

An employee can author zero or more preparation tasks, while a preparation task can be authored by only one employee.

An employee can be responsible for the completion of zero or more preparation tasks, while a preparation task can have only one employee responsible for its completion.

