Architecture

Text

Description automatically generated with medium confidence

24/03/2022 Eindhoven

Version: 0.1

Members:

Lars Kluijtmans: 4220269

Table of Contents

[Version 3](#_Toc100156852)

[Introduction 4](#_Toc100156853)

[How is SOLID guaranteed 5](#_Toc100156854)

[Single responsibility 5](#_Toc100156855)

[Open/closed principle 5](#_Toc100156856)

[Liskov substitution 5](#_Toc100156857)

[Interface segregation 5](#_Toc100156858)

[Dependency inversion 5](#_Toc100156859)

[Important design decisions 6](#_Toc100156860)

[Why user spring boot 6](#_Toc100156861)

[Front end library for building user interfaces 6](#_Toc100156862)

[Database to use 6](#_Toc100156863)

[C4 7](#_Toc100156864)

[C1 7](#_Toc100156865)

[C2 8](#_Toc100156866)

[C3 9](#_Toc100156867)

[C4 10](#_Toc100156868)

[Research Questions 11](#_Toc100156869)

[Problem 1: What are model and DTO (Data transfer object) used for. 11](#_Toc100156870)

[Major question 1: When to use a model and when to use a DTO 11](#_Toc100156871)

[Minor questions 1: 11](#_Toc100156872)

[Problem 2: How to identify a product. 11](#_Toc100156873)

[Major question 2: How to identify a product without using a regular id? 11](#_Toc100156874)

[Minor questions 2: 11](#_Toc100156875)

# Version

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Date | Autor | Changes | Status |
| 0.1 | 08/03/2022 | Lars Kluijtmans | First edition | Not finished |
| 0.2 | 02/04/2022 | Lars Kluijtmans | Adding versioning, introduction and index | Not finished |
| 0.3 | 06/04/2022 | Lars Kluijtmans | Remaking the research questions | Not finished |
| 0.4 | 10/04/2022 | Lars Kluijtmans | Decided to use mysql instead of mogodb | Version 0 is finished |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Introduction

Game market is an online market place that allows users to sell and buy used games by connecting buyers and sellers or through an auction. For sellers we also included features such as statistics to see how their games are selling, easy management of products they are selling… For buyers we made the search features as easy to use as possible and made the contacting of sellers very simple.

# How is SOLID guaranteed

## Single responsibility

Every class is responsible for only one type of object, in this application the only typed of objects are users and products, although they are connected there are different classes in charge of managing them.

## Open/closed principle

By using interfaces, I can, if needed extend from them to another class without having to modify any of the already existing code.

## Liskov substitution

In this application only the user object implements this principle, it has 2 subclasses, NormalUser and Admin.

## Interface segregation

At the moment this principle is not used in the application.

## Dependency inversion

In between every layer of this application (controller, service and repository) there is a interface separating them.

# Important design decisions

## Why user spring boot

Spring Boot helps developers to start coding right away without wasting time on preparing and configuring the environment. In contrast to other Java frameworks

## Front end library for building user interfaces

Most of the class wanted to use react so I just went with it. Still I believe that Angular would have been the better option, just because the client of our group project said he would prefer for us to use it and we have to use the same front end for both the individual and group projects .

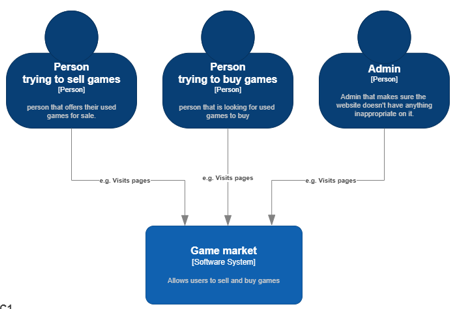
## Database to use

I will be using a mysql database for this project because I’m already familiar with it and the teacher (marcio) explained how to connect the API to a local mysql database (which I couldn’t get working for mongodb).

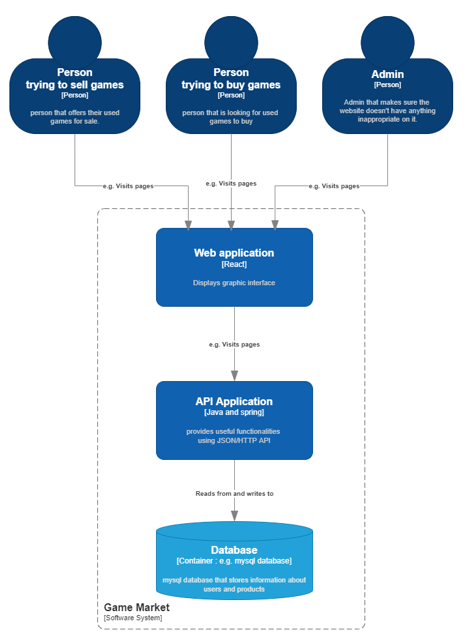
# C4

## C1

* This system has only 2 types of users but one of these, the NormalUser, can preform 2 very different actions, which is why they are shown in the images bellow as different people. One using the system to sell old and or used games and the other to buy these same games. There is also a admin that can remove users and products if he finds the inappropriate.

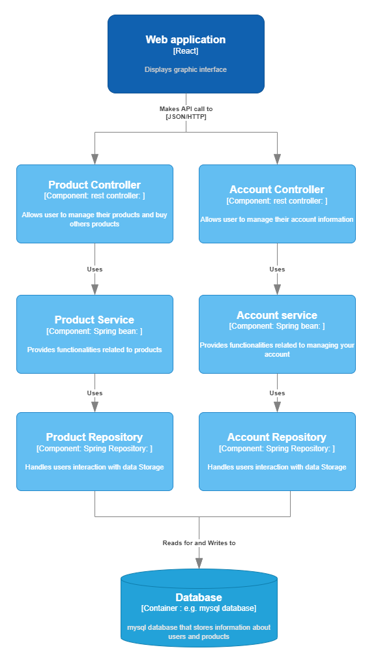


## C2

* This system is divided into 2 separate containers the Web applications and API. The Web application is the front end made using react that the users interact with and the API is responsible for sending and receiving data from and to the Web application.

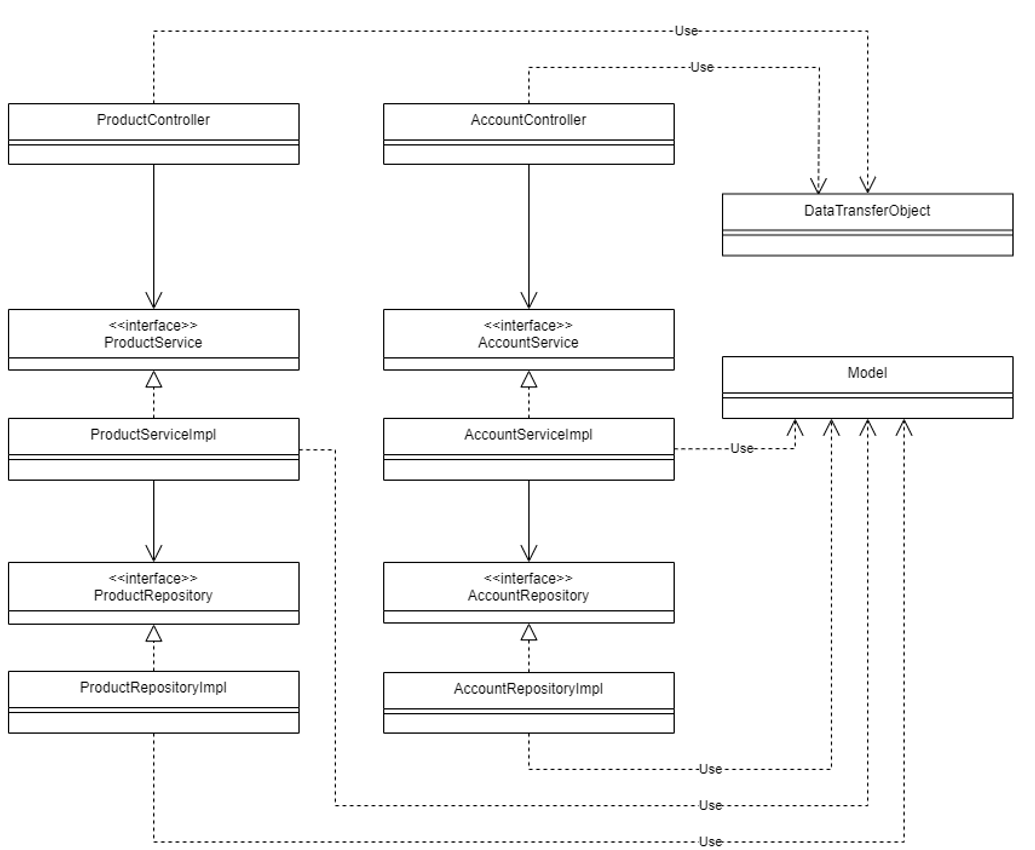
## C3

* The API as display here is divided into 2 sections (Accounts and Applications) and 3 layers(Controller, service, repository).



## C4

* A simple UML, the controller layer uses a DataTransferObject in contract to the Service and Repository layers that uses a Model. I think that this naming makes more sense then the other way around because the DataTranserObject is the object that will be transferred to the other containers that use the API.



# Research Questions

## Problem 1:

What are model and DTO (Data transfer object) used for.

## Major question 1:

When to use a model and when to use a DTO

## Minor questions 1:

What is a model?

* Literature study

A model is used to separate the controller from accessing the data directly according to the MVC flow (Model-view-controller). It is described as representing the business layer of a application. (Choudary, 2021)

Others referee to it as an Model Object or MO for short, these are mapped closely to a corresponding database table and are often used as return values for a Data Access Object. (Java Practices, 2018)

Although the two definitions may seem similar one uses a model a object that can be transferred between the business and database layers but also refers to it as representing the business layer and the other describes it as closely related to the database even saying that it is “often map roughly to the records of a corresponding database table.“. (Java Practices, 2018)

I would have to conclude that a model differs in function depending on the contest it is used in. If it appears related to a MVC flow then it has the function of a transitional object that is used in transferring data from the database layer to the business layer. If found in the context of a Model Object or used in reference to a Data access object it will be more closely related to the database layer and even the database structure itself.

* Available product analysis
* Community research

What is a DTO?

* Literature study

DTO is a abbreviation for data transfer object. The DTO pattern is used to reduce the number of methods called by combining multiple variables in one call, these contain no login and are used as storage. They are often composed of multiple models or parted of them. (Daeldung, 2022)

The DTO pattern is also mentioned often as a “objects that aggregate and encapsulate data for transfer”. It should contain serialization and deserialization mechanisms but no logic. The DTO are custom made to only send the data that is needed and to not include sensitive information increasing the security of the application. (Seniuk, 2020)

In conclusion a DTO is used as a collection of data that can be composed of multiple models or variables that include exclude unrequired or sensitive data and so increases security and speed. It also decreases the number of requests needed.

* Available product analysis
* Community research

Are there alternatives to model and DTO?

* Literature study
* Expert interview
* Design pattern research

Why would you convert a model to a DTO?

* System test
* Security test

What are the advantages and disadvantages of a model?

* Problem analysis
* SWOT analysis

What are the advantages and disadvantages of a DTO?

* Problem analysis
* SWOT analysis

Why would you convert a DTO to a model?

* Design pattern research
* Expert interview

When to convert a model to a DTO

* Design pattern research
* IT architecture sketching

When to convert a DTO to a model

* IT architecture sketching
* Design pattern research

## Problem 2:

How to identify a product.

## Major question 2:

How to identify a product without using a regular id?

## Minor questions 2:

What is a id?

* Literature study
* Community research

What are ids used for?

* Literature study
* Available product analysis

What different types of ids are there?

* Literature study
* Available product analysis
* Expert interview
* Design pattern research

What is a UUID?

* Literature study
* Community research

What advantages and disadvantages are there to a UUID?

* Problem analysis
* SWOT analysis