REVISION HISTORY

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 27/11/2023 | 1.0 | Package Diagram has been added. | Arda Onur |
| 27/11/2023 | 1.1 | GUI parts has been added. | Arda Onur |
| 28/11/2023 | 1.2 | DB Maps has been added. | Meriç Okçu, Buğra Bedir Durmuş |
| 28/11/2023 | 1.3 | Workflow and Algorithms diagrams has been added. | Meriç Okçu, Buğra Bedir Durmuş |
| 29/11/2023 | 1.4 | Component interfaces for the Model has been added | Kaan İşmen, Ömer Özmeteler |
| 29/11/2023 | 1.5 | Component design chart has been added | Kaan İşmen, Ömer Özmeteler |
| 29/11/2023 | 1.6 | COTS identification has been added | Arda Onur |
| 29/11/2023 | 1.7 | Workflow and Algorithms diagrams has been added to the Model part. | Kaan İşmen, Ömer Özmeteler |
| 29/11/2023 | 1.8 | Component Interface has been added | Arda Onur |
| 04/12/2023 | 1.9 | General issues fixed | Arda Onur |
| 05/12/2023 | 2.0 | Changed Sequence Diagrams according to feedback | Arda Onur, Kaan İşmen, Meriç Okçu |
| 05/12/2023 | 2.1 | Database Sequence diagram and class diagram changed | Ömer Özmeteler, Buğra Durmuş |
| 07/12/2023 | 2.2 | General table issues fixed | Arda Onur |

**TABLE OF CONTENTS**

**1** **Introduction 3**

***1.1*** ***References 3***

1.1.1 Project References 3

**2** **Software Architecture overview 3**

**3** **Software design description 4**

***3.1*** ***GUI 4***

3.1.1 Component interfaces 4

3.1.2 Component design description 5

3.1.3 Workflows and algorithms 6

3.1.4 Software requirements mapping 6

***3.2*** ***Model 6***

3.2.1 Component interfaces 7

3.2.2 Component design description 8

3.2.3 Workflows and algorithms 9

3.2.4 Software requirements mapping 9

***3.3*** ***Database 10***

3.3.1 Component interfaces 10

3.3.2 Component design description 10

3.3.3 Workflows and algorithms 11

3.3.4 Software requirements mapping 11

**4** **COTS Identification 12**

# Introduction

This document describes the design of the AVEC software system.

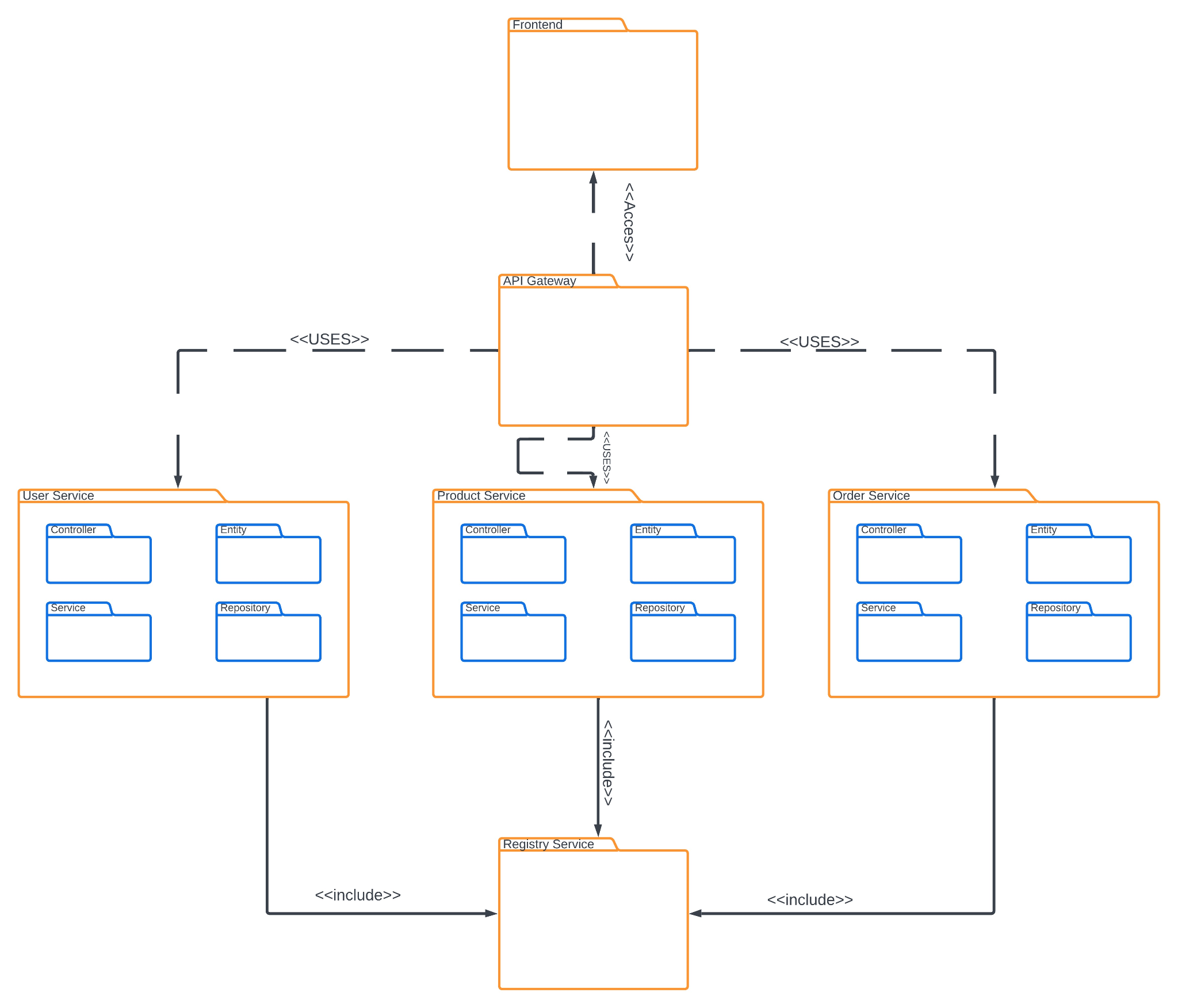
## References

### Project References

| # | Document Identifier | Document Title |
| --- | --- | --- |
| [SDP] | AVEC-SDP | AVEC Software Development Plan |
| [SRS] | AVEC-SRS | AVEC Software Requirements Specification |
| [STP] | AVEC-STP | AVEC Software Test Plan |

# Software Architecture overview

Package Diagram;



# Software design description

## GUI

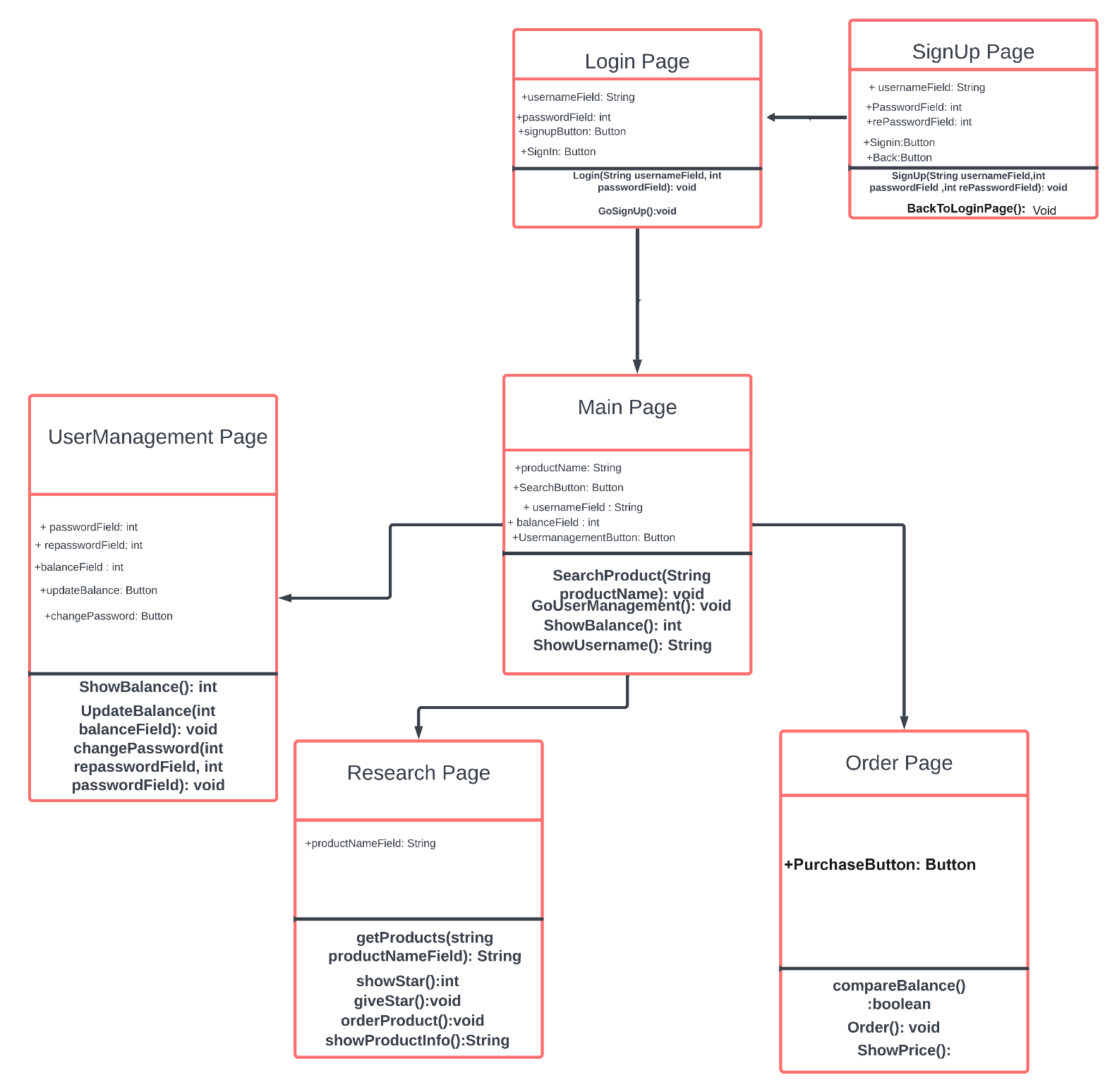
### Component interfaces

### On the login page, there are two username and password boxes that need to be filled. Below these, there are two clickable boxes, one for login and the other for sign up. There are three boxes in Sign up. One username box and two password boxes. When the user logs in, a new page opens. There are 2 boxes at the top of this page: product research, user management and on top-right corner there will be user’s name and user’s ballance . If the user wants to research a product, user can click on the product research box and search by typing the product he wants. If user finds the product user is looking for, he clicks on the product he wants and is directed to the window related to that product. If user wishes, user can give the product a star rating from this window or click buy the product. If user clicks on buy, a new window appears and the price and discounted price of the product he wants to buy and a clickable buy button appear. If there is sufficient balance, the product has been ordered message appears on the screen, otherwise an error message is encountered. In the user management section, there are 3 clickable boxes: password change box, product add box, budget add box. If you click on the password change box, 2 fillable boxes and one clickable box will appear. If these two boxes match and the box is ticked, a success message will appear on the screen. Otherwise, an error message is entered. If the product adding box is clicked, there will be three fillable boxes for the product name, product price and product description, and when these are filled, the message that the product has been added successfully will be displayed. If the balance upload box is clicked, the user's balance and a box that can be filled will appear on the screen. If the user enters a positive number here, a notification will be received that the balance has been updated by that amount.

AVEC dont have any interface classes because of being web-based applications. AVEC uses HTML CSS for frontend and Javascript for GUI interactions.

### Component design description

This structure is in the Frontend Package which is shown above Section 2.



### Workflows and algorithms

### 

### Software requirements mapping

SRS-AVEC-001.1,SRS-AVEC-003.1,SRS-AVEC-006.1,SRS-AVEC-007.1,SRS-AVEC-009.1

## Model

### Component interfaces

3 services will be used in the system: User Service, Product Service and Order Service. These micro services will be connected to the UI with API Gateway.

The User Service will manage the information about the user and their balance.

The Product Service will manage the information about the product and also will check the stock of the product.

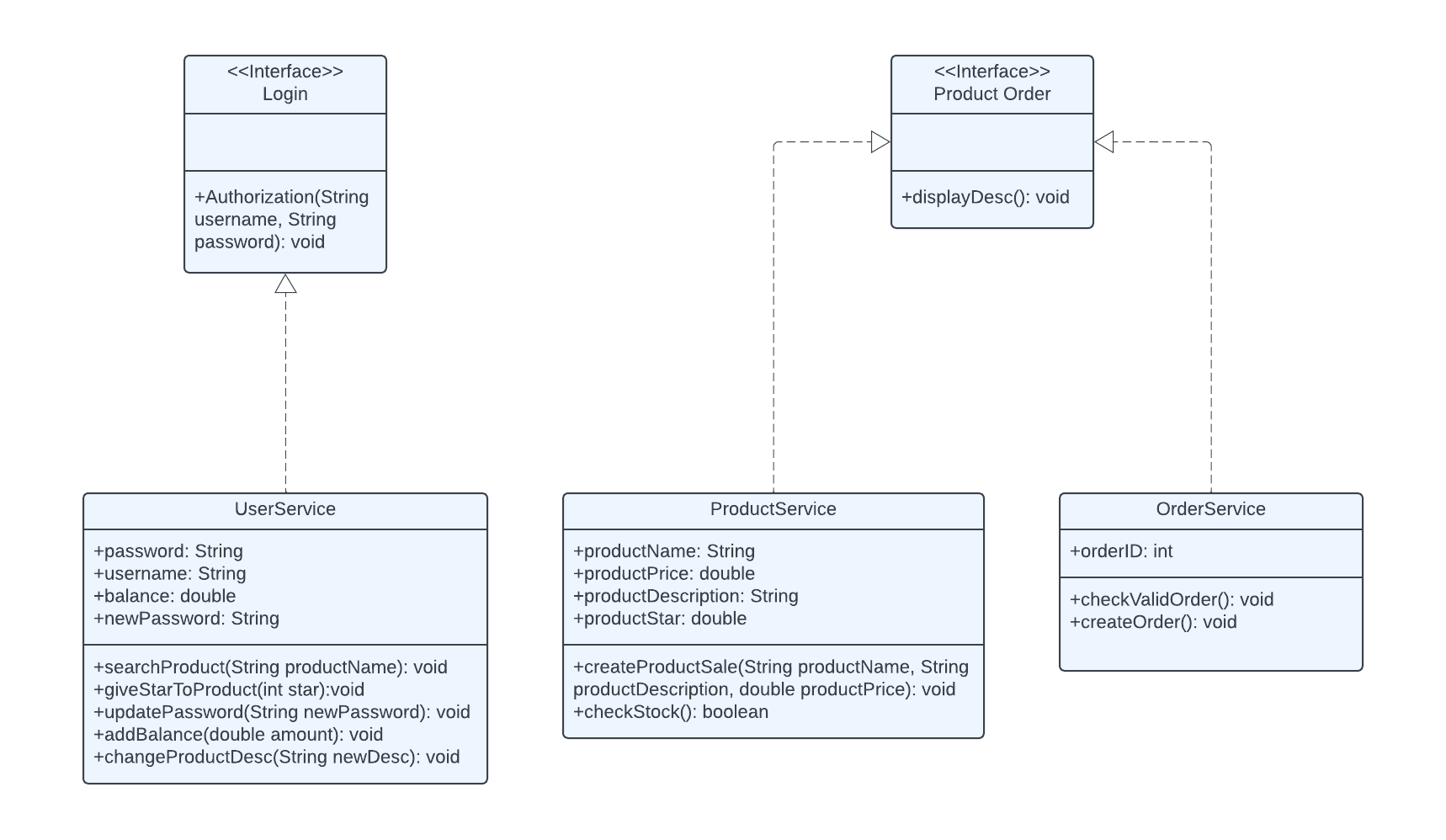
The Order Service will create the desired order if the certain qualifications are matched. The qualifications are:

1. The balance must be more or equal to the product price that is desired to be bought.

1. The product must have enough stock in order to be bought.



### Component design description

The Login interface is implemented by UserService and has 1 method which is Authorization which takes 2 String parameters, username and password respectively.

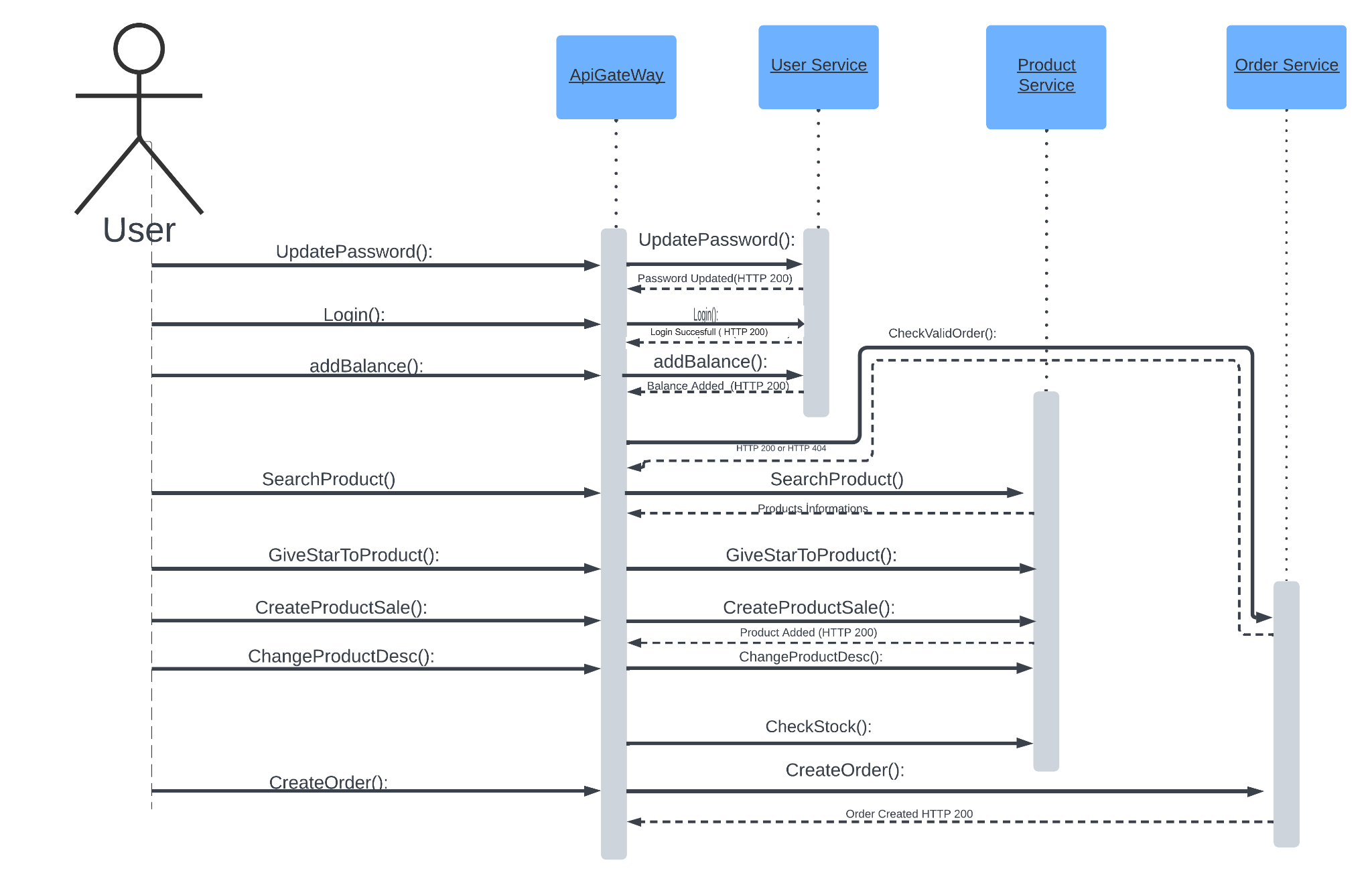
The UserService class has 4 attributes: password, username, balance and newPassword. It also has 5 methods: searchProduct, giveStarToProduct, updatePassword, addBalance and changeProductDesc.

The Product Order interface is implemented by the ProductService and OrderService class and has 1 method which is displayDesc.

The Product Service class has 4 attributes: productName, productPrice, productDescription and productStar. It also has 2 methods: createProductSale and checkStock.

The Order Service class 1 attribute: orderID. It also has 2 methods: checkValidOrder and createOrder.

### Workflows and algorithms



### Software requirements mapping

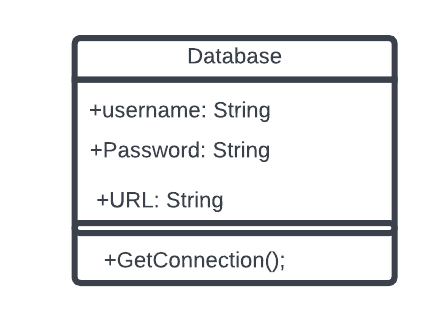
**-SRS-AVEC-001**,**-SRS-AVEC-002**,**-SRS-AVEC-003**,**-SRS-AVEC-004**,**-SRS-AVEC-005**,**-SRS-AVEC-006**,**-SRS-AVEC-007**,**-SRS-AVEC-008**,**-SRS-AVEC-009**

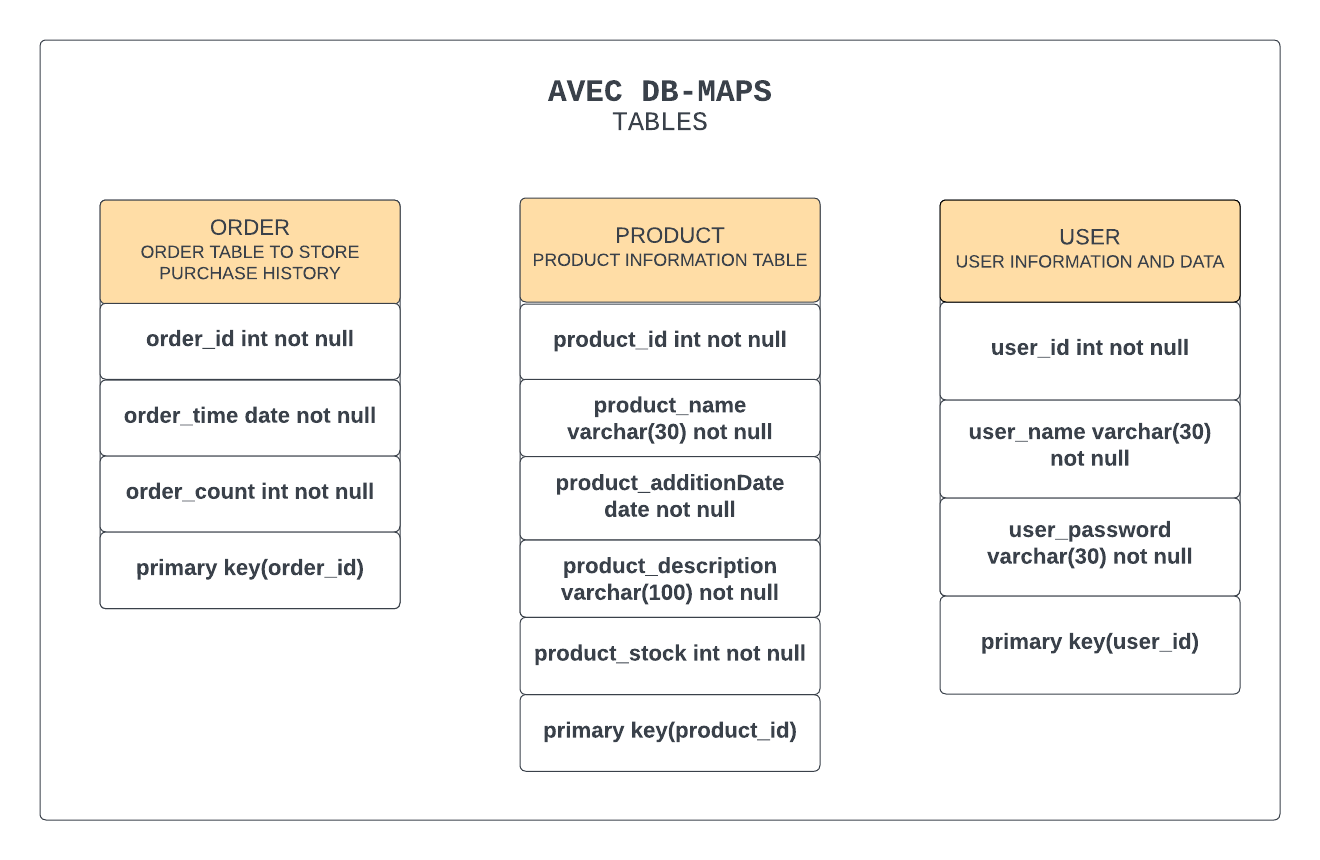
## Database

### Component interfaces

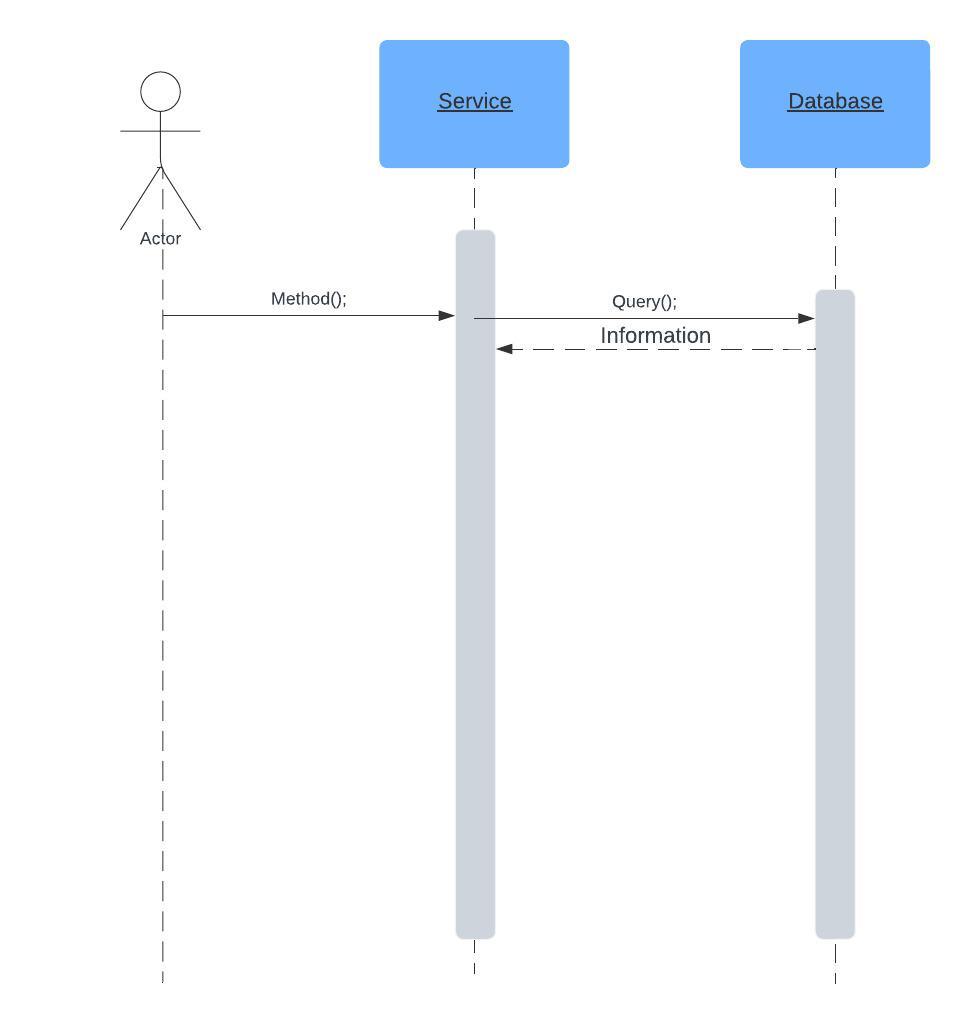
AVEC will be use Spring Data JDBC and Hibernate to communicate with Database which is MySQL

* + 1. **Component design description**





### Workflows and algorithms



### Software requirements mapping

SRS-AVEC-001.1, SRS-AVEC-002.1,SRS-AVEC-003.1,SRS-AVEC-004.1,SRS-AVEC-005.1,SRS-AVEC-007.1,SRS-AVEC-009.1,

# COTS Identification

COTS (commercial of the shelf) libraries used in AVEC are the following:

* Spring Data JDBC

https://spring.io/projects/spring-data

* Spring Cloud

https://spring.io/projects/spring-cloud

* Spring Web

https://spring.io/projects/spring-cloud

* Hibernate

https://hibernate.org/orm/releases/

* MySql

https://dev.mysql.com/downloads/workbench/

* Java

https://www.java.com/tr/download/

* JDK

https://www.oracle.com/java/technologies/downloads/