Interior Design Website

Project documentation

Student:

Chereji Iulia

**Group:**

30434

**Contents**

[I Project specification](#_heading=h.gjdgxs) **3**

[1.1 Domain Model Diagram](#_heading=h.30j0zll) 3

[II Use-Case model](#_heading=h.1fob9te) **3**

[2.1 Users and stakeholders](#_heading=h.f2irg1azp7rm) 4

[2.2 Use-Case identification](#_heading=h.3znysh7) 4

[2.3 UML Use-Case diagram](#_heading=h.2et92p0) 4

[III Architectural design](#_heading=h.tyjcwt) **5**

[3.1 Conceptual architecture](#_heading=h.3dy6vkm) 5

[3.2 Package diagram](#_heading=h.1t3h5sf) 5

[3.3 Class diagram](#_heading=h.4d34og8) 5

[3.4 Database (E-R/Data model) diagram](#_heading=h.2s8eyo1) 6

[3.5 Sequence diagram](#_heading=h.17dp8vu) 6

[3.6 Activity diagram](#_heading=h.3rdcrjn) 6

[IV Supplementary specifications](#_heading=h.26in1rg) **6**

[4.1 Non-functional requirements](#_heading=h.lnxbz9) 6

[4.2 Design constraints](#_heading=h.35nkun2) 6

[V Testing](#_heading=h.1ksv4uv) **7**

[5.1 Testing methods/frameworks](#_heading=h.44sinio) 7

[5.2 Future improvements](#_heading=h.2jxsxqh) 7

[VI Bibliography](#_heading=h.z337ya) **7**

# 

# I Project specification

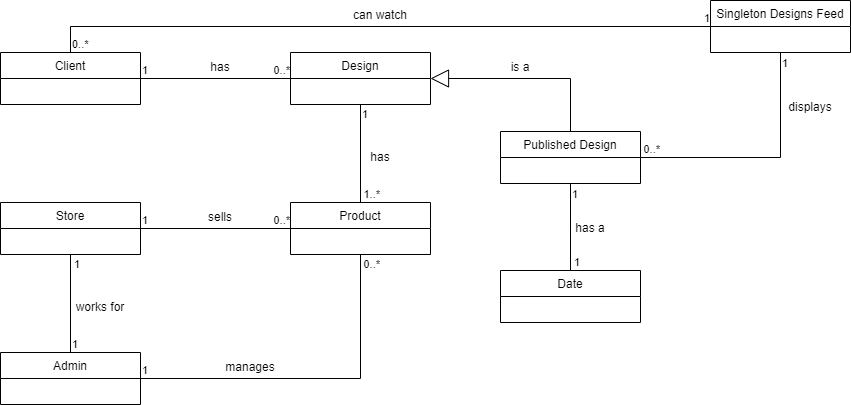
A client-server application that allows users to create interior designs for their home with products available in Romania. They can choose multiple items to add to their design and at the end they can download a shopping list or make their design public so other people can see and use it.

The application has two types of users: clients and admins.

The clients can log in and register on the website (optional), start a design, add or delete products from the design, search for products by store, type, keywords, finish a design, download the shopping list for a design, make a design public (by default they are private) and see their designs.

The admins can log into the website, they work for a specific store and their accounts are made for them. They can add, update or delete products of the store for which they work.

## 1.1 Domain Model Diagram

**

# II Use-Case model

The use cases have as main goals identifying the user on the website and performing CRUD operations. The admins can create, request, update and delete products and the clients can create, request, update and delete designs and only request products.

## 2.1 Users and stakeholders

Users: admins that manage the products, users that did not log in and can just watch the designs and the products, users that logged in that can create and post designs.

Stakeholders: application manager, representatives of the stores that are included in the application, other stores’ managers that might want to sign up to include their products in the application.

## 2.2 Use-Case identification

Use case name: Log in

Level: User-Goal

Main actor: Both client and admin

Main success scenario:

1. The user clicks the log in button
2. He enters the correct credentials
3. He is identified and taken to the main page for users/admins

Extension:

1. The user clicks the log in button
2. He enters incorrect credentials
3. The application displays a message

Use case name: Create a design

Level: User-Goal

Main actor: Logged in client

Main success scenario:

1. The users clicks the create design button
2. He adds one or more products to the design
3. He finishes the design and chooses to make it public or private
4. He downloads a shopping list for the design

Extension:

1. The user clicks the create design button
2. He is not logged in so he is redirected to the log in / register page

Use case name: Update product

Level: User-Goal

Main actor: Admin

Main success scenario:

1. The admin searches for a product from their shop’s list
2. He clicks the update button
3. He modifies the data and clicks save
4. The product is updated in the application database

Extension:

1. The admin chooses a product to update
2. He enters invalid data and clicks save
3. The application displays an error message and the product is not updated

Use case name: Search for products

Level: User-Goal

Main actor: Both clients and admins

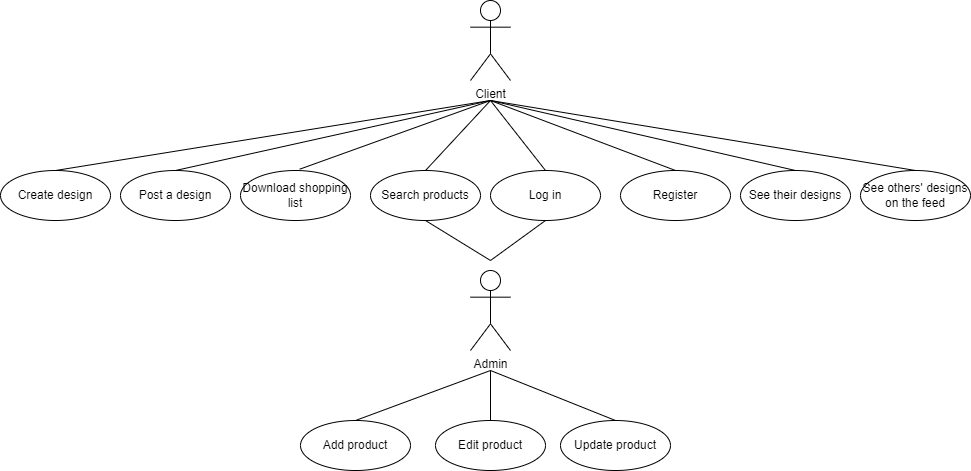
Main success scenario:

1. The user enters data in the search fields
2. He clicks the search button
3. The products that matched the criteria are displayed

Extension:

1. The user clicks search without filling any field
2. The application displays a message and no search is performed

## 2.3 UML Use-Case diagram



# 

# III Architectural design

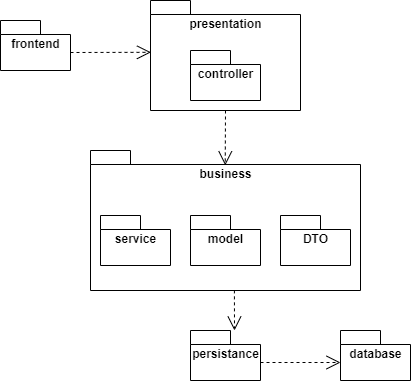
## 3.1 Conceptual architecture

The application is a web application. It implements the Client-Server architecture. This architectural design is a distributed application structure that partitions tasks between the providers of a resource or service, called servers, and service requesters, called clients. Clients and servers communicate over a computer network. A client does not share any of its resources, but requests a server's content or service function. The website uses a relational database stored into the server.

The Layered Architectural Pattern will be used since a client request should go through many stages of validations and verifications before accessing the resource.

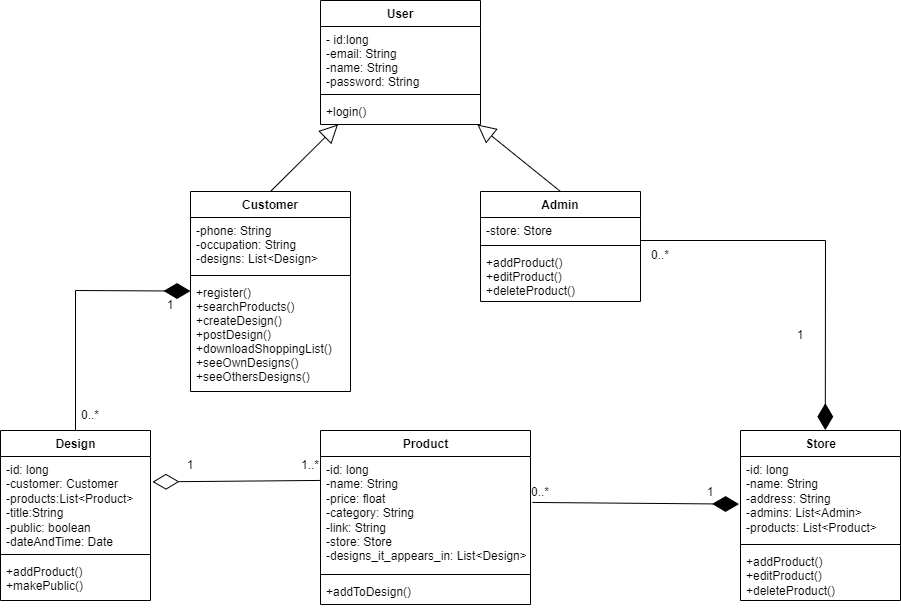
## 

## 3.2 Package diagram

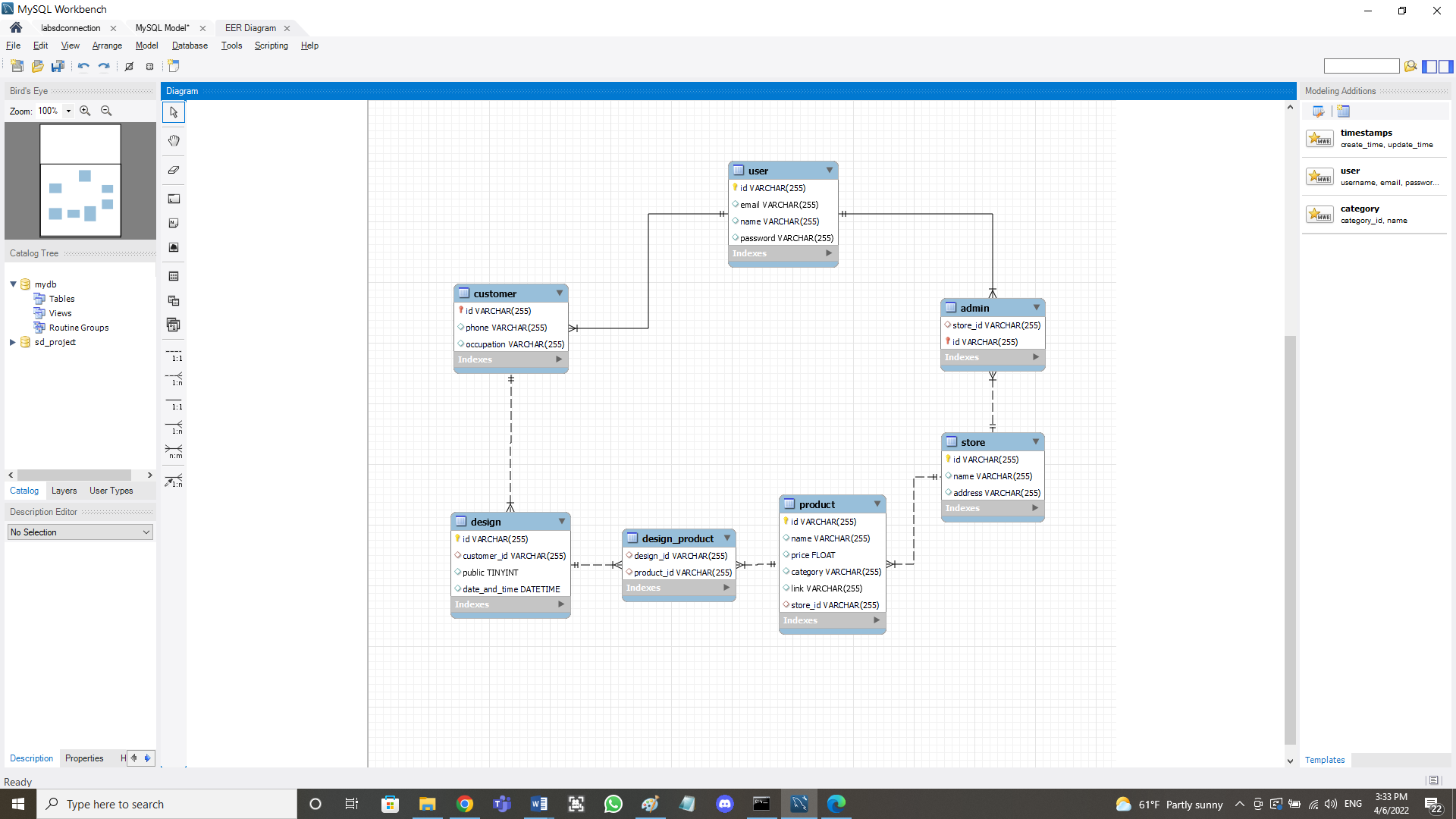
**

## 

## 3.3 Class diagram



## 3.4 Database (E-R/Data model) diagram



## 

## 3.5 Sequence diagram

## 

## 3.6 Activity diagram

# IV Supplementary specifications

*< Se va scrie o mica introducere./>*

## 4.1 Non-functional requirements

4.1.1 Availability

Availability is the ratio of time a system or component is functional to the total time it is required or expected to function. The website must be available every day of the week because users might try to access it at all times.

4.1.2 Security

The security of the application is important because no user should be able to access another user’s account without the credentials or to access a page that he should not be seeing by writing the appropriate URL. The application will implement the hashing of passwords and it will check what is the user that is logged in when a page is accessed and if it that user should have access to it.

4.1.3 Scalability

A system is described as scalable, if it will remain effective when there is a significant increase in the number of resources and the number of users. The application should be scalable because it should support new users as they create accounts, new designs as they are created and new stores with all their products when they sign a contract with the website owner.

4.1.4 Performance

The response time should be as small as possible so users don’t have to wait too long for their requests to receive a response.

## 4.2 Design constraints

4.2.1 Languages

The application will be written using Java and SQL for the backend and Javascript, HTML and CSS for the frontend.

4.2.3 Frameworks

The frameworks that will be used are Spring, React, JPA and Hibernate, and the development tool is IntelliJ IDEA.

4.2.4 Security measurements

The passwords will be stored using a hash function and the user rights will be checked when accessing a page.

4.2.5 Database specification

The application will use a relational database managed using MySQL Workbench 8.0.

4.2.6 Browsers

The application mush be able to run on any computer which has any browser from Microsoft Edge to Google Chrome.

# V Testing

*< Se va discuta la laborator./>*

## 5.1 Testing methods/frameworks

## 5.2 Future improvements

# VI Bibliography