



## Coffee Shop Application

### Regular Exam [Spring Fundamentals]

The **Coffee Shop** Application is here to help with the normal work of the cafe. The application records all orders that are not yet ready and reminds the employees of them by visualizing a picture of the category, the product's name and its price. It also calculates the approximate time that will be required for the execution of the orders. As an additional functionality, statistics are kept, on which employees can process how many orders are placed at that moment. At this stage, the functionality of the application is small but later it can be expanded upon.

There are several requirements you must follow in the implementation:

#### 1. Database Requirements

The **Database** of the **Coffee Shop** application needs to support **3 entities**:

##### User

- **Id** – Accepts **UUID-String** or **Long** values
- **Username**
  - The **length** of the **values** should be **between 5 and 20** characters long (both numbers are **INCLUSIVE**)
  - The values should be **unique** in the database
- **First name**
  - The **length** of the **values** should be **between 5 and 20** characters long (both numbers are **INCLUSIVE**)
- **Last name**
  - The **length** of the **values** should be **between 5 and 20** characters long (both numbers are **INCLUSIVE**)
- **Password**
  - The **length** of the **values** should be **at least 3** characters long (**INCLUSIVE**)
- **Email**
  - The values should contain a '@' symbol)
  - The values should be **unique** in the database

##### Order

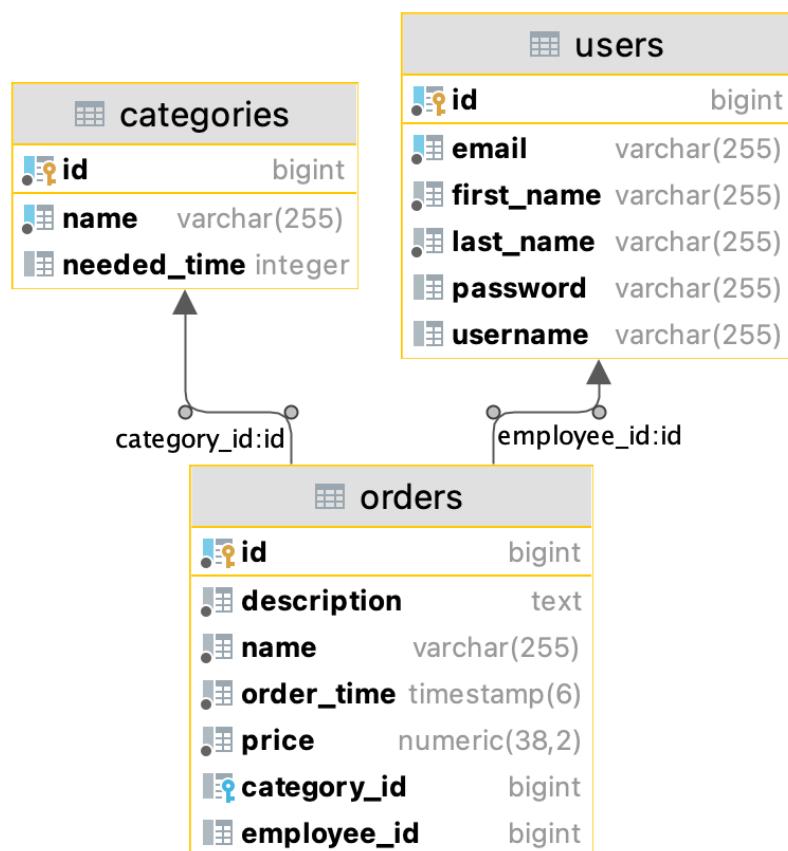
- **Id** – Accepts **UUID-String** or **Long** values
- **Name**
  - The **length** of the **values** should be **between 3 and 20** characters long (both numbers are **INCLUSIVE**)
- **Price**
  - The values must be a **positive** numbers
- **Order Time**
  - **Date and Time** values, that **cannot** be in the **future**
- **Category**
  - One product **has one** category and one category **can have many** products
- **Description**
  - The **length** of the **values** should be **at least 5** characters long
  - The description is a **long text field**
- **Employee**
  - A user that **adds this order** to the **Coffee Shop** application

## Category

- **Id** – Accepts **UUID-String** or **Long** values
- **Style name**
  - The values should be **unique** in the database
  - an option between (**COFFEE**, **CAKE**, **DRINK** and **OTHER**)
- **Needed Time**
  - The approximate **time in minutes** needed for the **preparation** of a product of the **specified category**
    - The needed time for a **Drink** is **1 min**
    - The needed time for a **Coffee** is **2 min**
    - The needed time for a **Other** is **5 min**
    - The needed time for a **Cake** is **10 min**

**Nullable/Empty/Blank values are not allowed unless explicitly mentioned.** Implement the entities with the **correct data types** and implement the **repositories** for them.

ER Diagram:



## 2. Initialize categories

- Implement a method that checks (when app started) if the database does not have any category and initialize them
  - You are free to do this in some different ways
  - You can skip the description if you want

### 3. Page Requirements

#### Index Page (logged out user)

Home Login Register

# Welcome to SoftUni Coffee Shop

**Coffee**

In our cafe you can find all kinds of coffee drinks that you can imagine. Short Italian coffee or huge American coffee, a hot drink to keep you warm or an iced frappe to keep you cool. The choice is yours. :)

**Cake**

To fully enjoy our excellent coffee, you will surely need a bite of our sweet products. Here comes the sweet part of our menu. Whether you choose Rainbow cake with Happy lion cake or prefer Unicorn, the choice is entirely yours, but the important thing is to relax and fully enjoy this moment. :)

**Drink**

If you feel that you need a little extra freshness to our excellent coffee, you can browse our menu of drinks. There you will find a huge variety of drinks, starting from natural juices and carbonated drinks, going through non-alcoholic and light alcoholic cocktails, to stronger drinks. :)

©SoftUni Spring Team 2021. All rights reserved.

#### Login Page (logged out user)

Home Login Register

# Login

Enter valid username and password

Username

Password

Login

©SoftUni Spring Team 2021. All rights reserved.

#### Login Page validations

Username

Username length must be between 5 and 20 characters.

Password

Password length must be more than 3 characters.

## Wrong username and password combination.

nasko

Password

Coffee Shop

## Registration Page (logged out user)

Home Login Register

# Register

Username

First name

Last name

Email

Password

Confirm Password

Register

©SoftUni Spring Team 2021. All rights reserved.

## Registration Page validations

- Note: it is not necessary to show message for different passwords, just not save the user and redirect again to the register page.

Username  1 Username length must be between 5 and 20 characters.

First name  1

Last name  1 Last name length must be between 5 and 20 characters.

Email  1 Enter valid email address.

Password  Password Password length must be more than 3 characters.

Confirm Password  Confirm Password Password length must be more than 3 characters.

Register

## Navigation (Guest user)

- Note: can access only to **Index**, **Login**, **Register** pages.

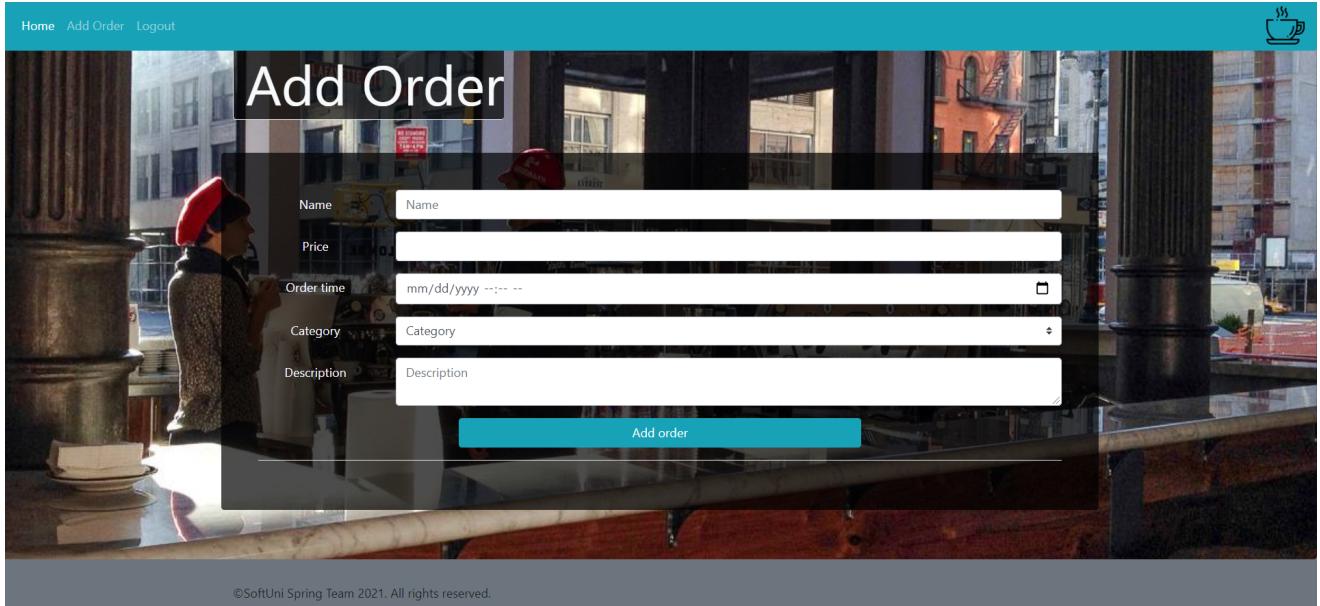


## Navigation (Registered user)

- Note: can access only to **Home** and **Add Order** pages and **Logout** functionality.



## Add Order



Home Add Order Logout ☕

# Add Order

Name

Price

Order time  mm/dd/yyyy --:-- --

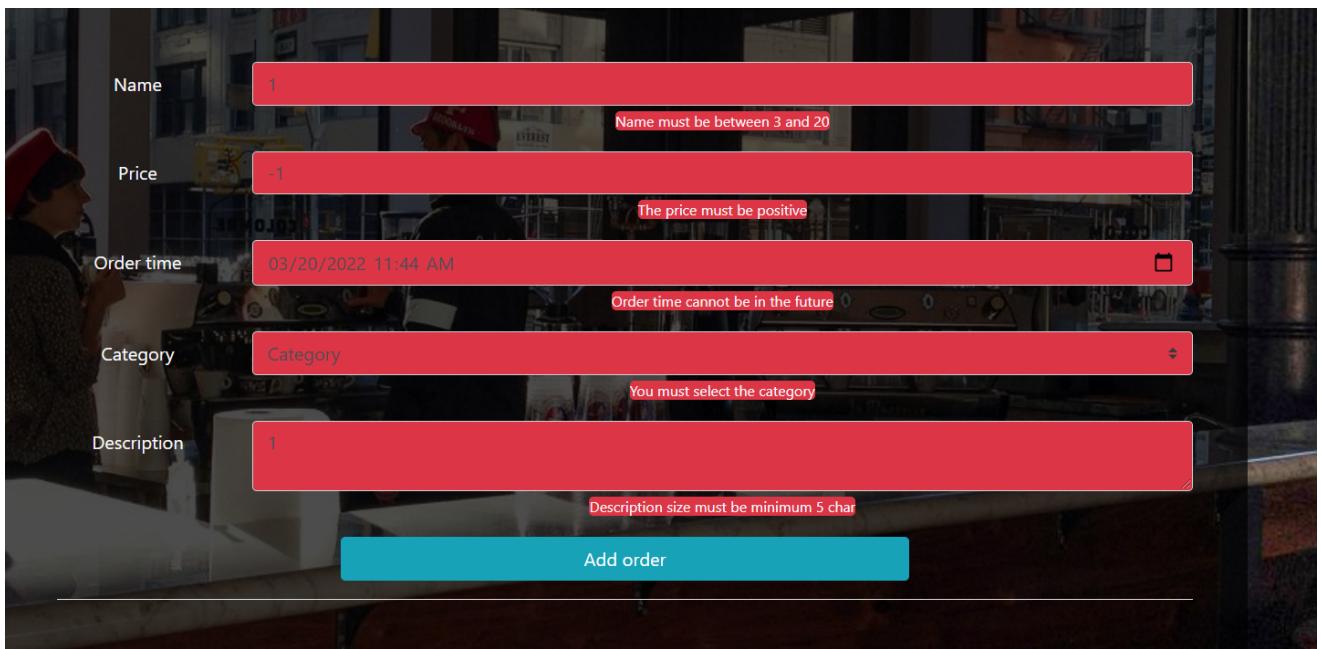
Category

Description

**Add order**

©SoftUni Spring Team 2021. All rights reserved.

## Add Offer Validation



Name  1 Name must be between 3 and 20

Price  -1 The price must be positive

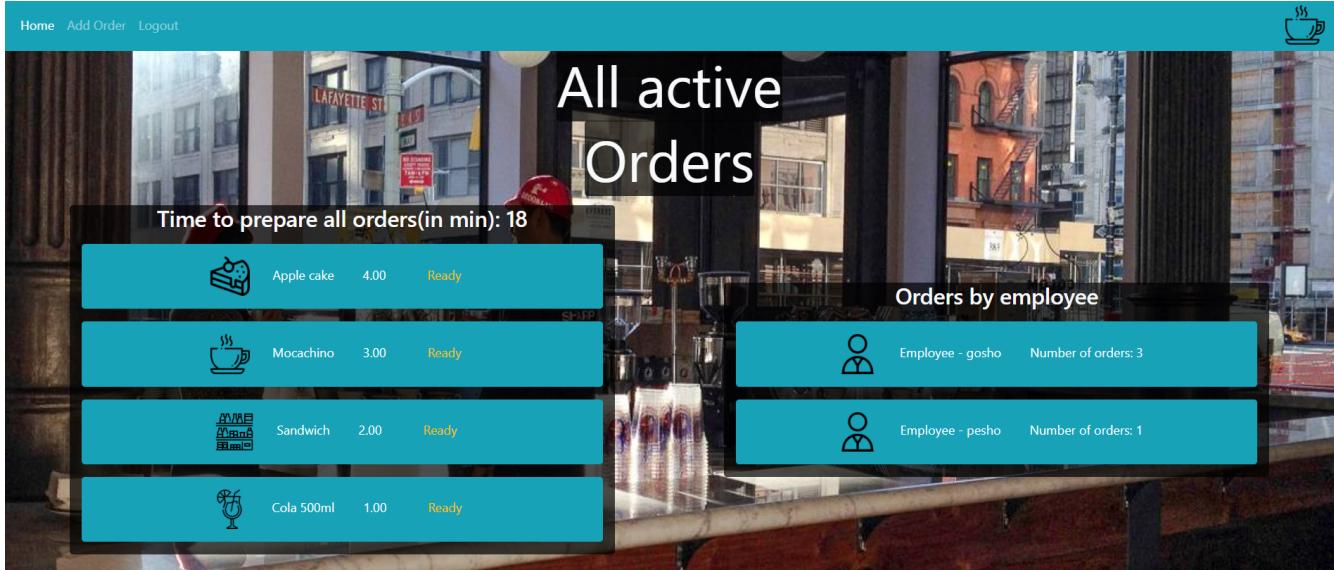
Order time  03/20/2022 11:44 AM Order time cannot be in the future

Category  Category You must select the category

Description  1 Description size must be minimum 5 char

**Add order**

## Home Page (with orders)



The templates have been given to you in the application skeleton, so make sure you implement the pages correctly.

### NOTES:

- The templates should look **EXACTLY** as shown above.
- The templates do **NOT require additional CSS** for you to write. Only **bootstrap** and the **given CSS** are enough.

## 4. Functional Requirements

The **Functionality Requirements** describe the functionality that the **Application** must support.

The **application** should provide **Guest** (not logged in) users with the functionality to **login**, **register** and **view the Index** page.

The application should provide Users (logged in) with the functionality to **logout**, **add an Order**, **view all Orders in DB (Home page)** and **Ready a single one** from orders.

In **Coffee Shop Application**, the navbar should redirect to **appropriate URL depending** on that if the user is logged in.

The **application** should provide **functionality** for **adding orders** with **category** and **employee**. Also can **Ready** them and remove from the database.

**Ready** button remove the selected order from the database and again **redirect** to the **home** page.

The first **header** on the **home** page shows the **total needed time to prepare all orders in minutes (integer number)** in the database at this moment.

**Order by Employee** tab show **every** registered employee and his **unready orders** at this moment.

- The orders are ordered by their order time in ascending order
- The picture in front of every order depends of order's category.
- The employees are ordered by their count of orders in descending order.

The application should store its data in a **MySQL** database or **PostgreSQL** database.

## 5. Security Requirements

The Security Requirements are mainly access requirements. Configurations about which users can access specific functionalities and pages.

- **Guest** (not logged in) users can access:
  - **Index** page;
  - **Login** page;
  - **Register** page.
- **Users** (logged in) can access:
  - **Home** page;
  - **Add Order** page;
  - **Logout** functionality.