# Practical project **“ProductMe”** description

**Team of developers:**

- **Krišjānis Mārtiņš Alliks, ka21042 - (Business logic development, MVC programming);**
- **Valters Strods, vs21161 - (UX, MVC programming);**
- **Domeniks Ulmanis, du21010 (UX, Database creation, MVC programming);**

**Development Environment:**

The system will be developed using Java 11 and SpringBoot 2.6.4. For data storage we are planning to use PostgreSQL. VCS is going to be Git and it will be stored on GitHub.

**Motivation:**

The key to any successful business of these modern times is – effectiveness and optimization. This can be achieved by asking potential business owners this fundamental question - how do we accomplish more by using fewer resources? In this day and age of IT the question is no longer so hard to answer – entrust simple tasks to machines, namely computers, thereby sparing yourself the number of human resources needed to run your business.

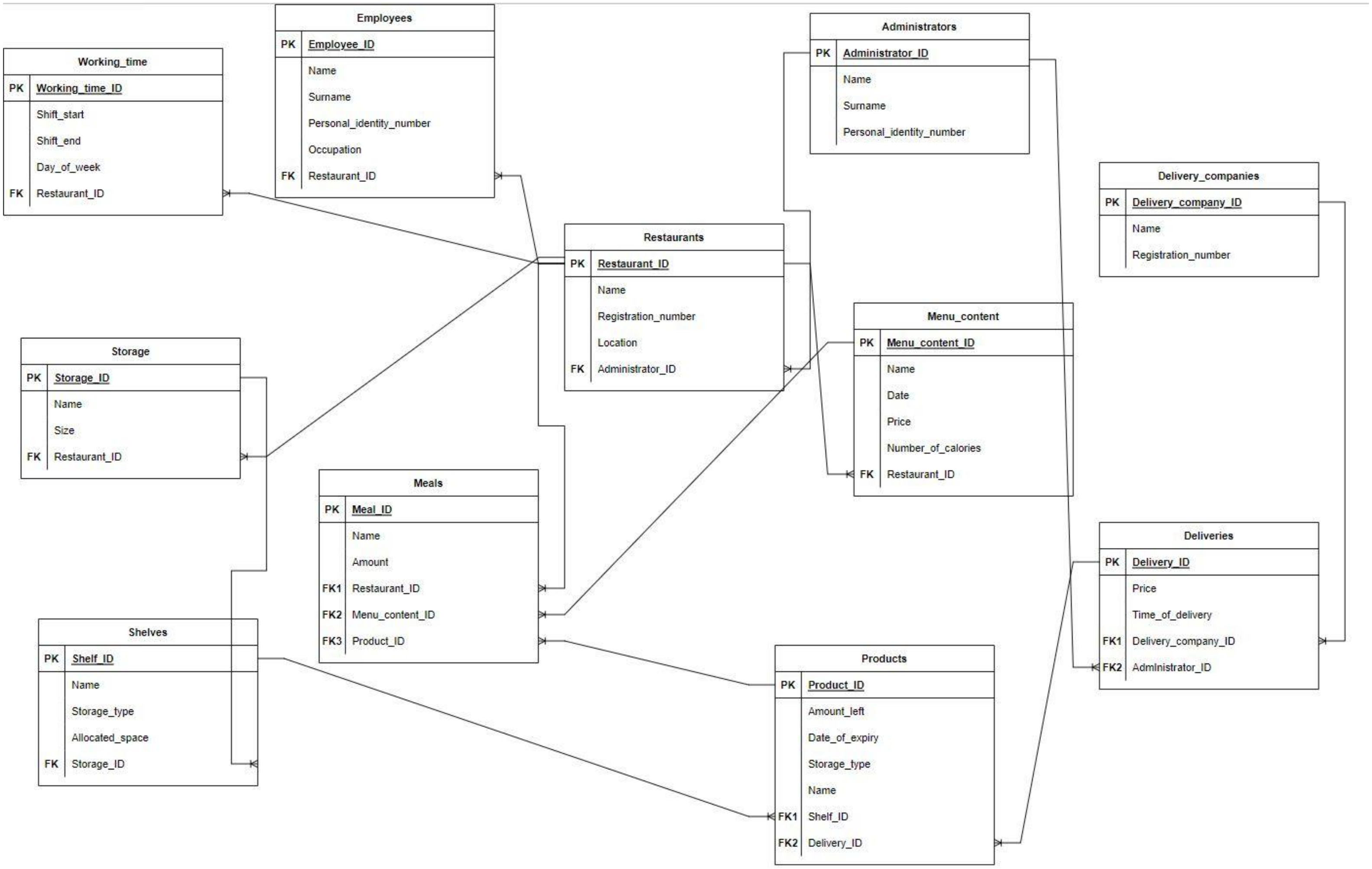
For this project we have decided to focus our idea on the food industry, specifically restaurants, since we believe this to be a rapidly developing industry. Since the pandemic the number of people choosing to order food from restaurants by delivery has skyrocketed. Now more than ever restaurants that provide such services are in need of workload optimization. We propose to create a management system which would help owners keep track or their supply and demand needs

**Main functionality:**

Any restaurant has its food storage chamber, usually a large, refrigerated unit. In this unit we have organized shelves each containing a specific product, e.g., carrots, potatoes, beef, lamb etc. Each shelf contains a finite amount of afro mentioned product. Naturally each of these items has its own expiration date which is important to the restaurant since it wants to provide its customers with the freshest produce. To keep track of all this information we would create a database specifically catered to the needs of restaurants. Each shelf would be indexed so its contents could be reflected in the database. To keep track of the expiration date each product would require having sub shelves so you can divide the produce according to its freshness. The amount of sub shelves would equal the amount of produce deliveries per a week. Such a database would allow the restaurant to easily see the amount of any product they have in storage, its location, its expiration date. Further all this data can be processed for accounting needs. You would be able to see how long it takes for a certain amount of a product to be fully consumed and from this information calculate the optimal amount of it you always need to have in storage. Many other accounting operations could be carried out from the database information such as calculation of expenses, etc.

**Data Registry:**

There will be a lot of databases, to register all of the data such as Users, Storages, Products, etc. Data usage model can be viewed - 1. image.



1. Image - database relations

**MVC:**

System will be designed with Model-View-Controller pattern.

Models:

- Restaurant
- RestaurantWorkTime
- Storage
- Shelf
- Product
- Delivery
- DeliveryCompany
- Worker
- Administrator
- RestaurantMenu
- MealIngredients

Views:

- Login view;
- Register view;
- List of all administrators/workers;
- List of delivery companies;
- List of products and operations available with those products;
- New order view;
- Edit user information view;
- View of adding/deleting/editing delivery companies;
- View of adding/deleting/editing restaurants;
- View of adding/deleting/editing storages;
- View of adding/deleting/editing shelves;
- View of adding/deleting/editing workers;
- View of adding/deleting/editing restaurant menu;
- View of adding/deleting/editing products in a recipe of a menu.

Controllers:

- LoginController - authentication and password operations;
- RestaurantController - create, delete, list, update restaurants;
- RestaurantWorkTimeController - create, delete, list, update working time;

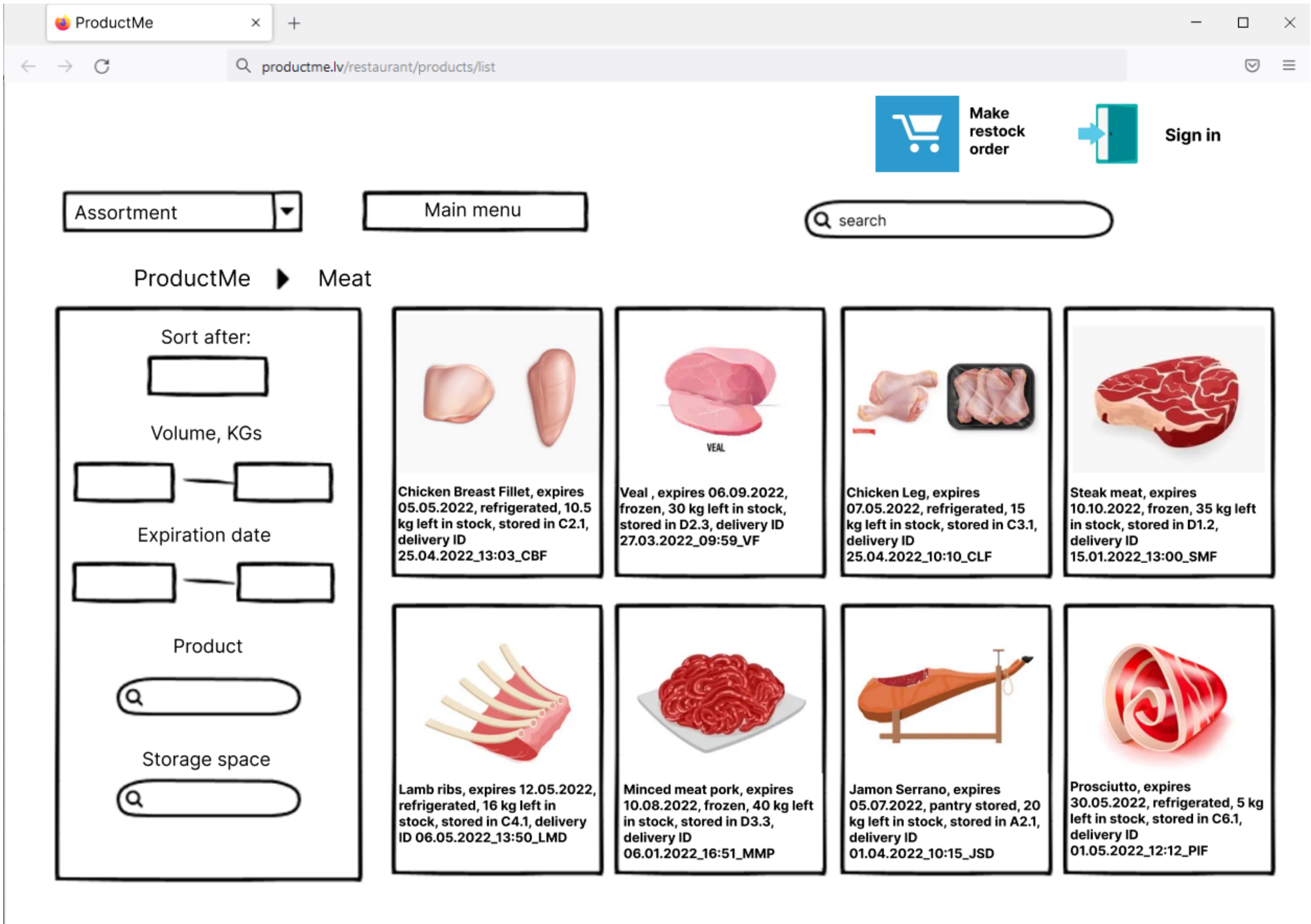
- StorageController - create, delete, list, update storages;
- ShelfController - create, delete, list, update shelves;
- ProductController - create, delete, list, update products;
- DeliveryController - create, delete, list, update deliveries;
- DeliveryCompanyController - create, delete, list, update delivery companies;
- WorkerController - create, delete, list, update workers;
- AdministratorController - create, delete, list, update administrators;
- RestaurantMenuController - create, delete, list, update menus;
- MealIngredientsController - create, delete, list, update meal ingredients;

**User Roles:**

Worker - can see inventory
Administrator - full control, access to everything, can manage products

**User Authentication:**
For user authentication we will use accounts with username and password, registered by administrator and/or registered.

**System Interface:**



Chicken Breast Fillet, expires 05.05.2022, refrigerated, 10.5 kg left in stock, stored in C2.1, delivery ID 25.04.2022_13:03_CBF

Veal , expires 06.09.2022, frozen, 30 kg left in stock, stored in D2.3, delivery ID 27.03.2022_09:59_VF

Chicken Leg, expires 07.05.2022, refrigerated, 15 kg left in stock, stored in C3.1, delivery ID 25.04.2022_10:10_CLF

Steak meat, expires 10.10.2022, frozen, 35 kg left in stock, stored in D1.2, delivery ID 15.01.2022_13:00_SMF

Lamb ribs, expires 12.05.2022, refrigerated, 16 kg left in stock, stored in C4.1, delivery ID 06.05.2022_13:50_LMD

Minced meat pork, expires 10.08.2022, frozen, 40 kg left in stock, stored in D3.3, delivery ID 06.01.2022_16:51_MMP

Jamon Serrano, expires 05.07.2022, pantry stored, 20 kg left in stock, stored in A2.1, delivery ID 01.04.2022_10:15_JSD

Prosciutto, expires 30.05.2022, refrigerated, 5 kg left in stock, stored in C6.1, delivery ID 01.05.2022_12:12_PIF