Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ

# Розробка клієнт-серверних застосувань на Java

Звіт з курсового проєкту

Підготували студенти

2-го року навчання

Факультету інформатики

Спеціальністі «Інженерія програмного забезпечення»

Ксьондзик Максим

Старовойт Аліна

Горуля Катерина

Викладач: Бабич Трохим

Київ 2020

**Зміст:**

[1. Постановка і аналіз задачі: 2](file:///C:\Users\mmmek\Downloads\Telegram%20Desktop\Звіт.docx#_Toc43406195)

[2.    Короткий опис системи, в якій реалізований проект. 3](file:///C:\Users\mmmek\Downloads\Telegram%20Desktop\Звіт.docx#_Toc43406196)

[3. Розподіл ролей в середині команди: 4](file:///C:\Users\mmmek\Downloads\Telegram%20Desktop\Звіт.docx#_Toc43406197)

[4. Визначення класів 5](file:///C:\Users\mmmek\Downloads\Telegram%20Desktop\Звіт.docx#_Toc43406198)

[5. Опис GUI (графічний інтерфейс користувача) 8](file:///C:\Users\mmmek\Downloads\Telegram%20Desktop\Звіт.docx#_Toc43406199)

[6. Інструкція користувача 10](file:///C:\Users\mmmek\Downloads\Telegram%20Desktop\Звіт.docx#_Toc43406200)

[7.    Висновок. 15](file:///C:\Users\mmmek\Downloads\Telegram%20Desktop\Звіт.docx#_Toc43406203)

[8.  Додаток. Вихідні тексти програми з коментарями. 16](file:///C:\Users\mmmek\Downloads\Telegram%20Desktop\Звіт.docx#_Toc43406204)

1. **Постановка і аналіз задачі**

В рамках курсового проєкту данного курсу потрібно було реалізувати автоматизоване робоче місце.

Необхідно автоматизувати роботу невеликого підприємства по роботі з складом.

Існує декілька груп товарів (наприклад: Продовольчі, непродовольчі...). В кожній групі товарів існують конкретні товари (наприклад: борошно, гречка ...). У кожного товару є наступні властивості - назва, опис, виробник, кількість на складі, ціна за одиницю. Група товарів містить наступні властивості - назва, опис.

Реалізувати:

1. Реалізувати графічний інтерфейс користувача
2. Клієнт та сервер
3. Збереження даних в БД на сервері.
4. Назва товару - унікальна (не може зустрічатися більше в жодній групі товарів).
5. Назва групи товарів - унікальна.
6. Реалізувати додавання/редагування/видалення групи товарів - при видаленні групи товарів, видаляти і всі товари.
7. Реалізувати додавання/редагування/видалення товару в групу товарів (мається на увазі назва, опис, виробник, ціна за одиницю).
8. Реалізувати інтерфейс додавання товару (прийшло на склад крупи гречаної - 10 штук), інтерфейс списання товару (продали крупи гречаної - 5 шт.)
9. Пошук товару.
10. Вивід статистичних даних: вивід всіх товарів з інформацією по складу, вивід усіх товарів по групі товарів з інформацією, загальна вартість товару на складі (кількість \* на ціну), загальна вартість товарів в групі товарів.

Робота мала бути організована по мережі. Також задачею стало максимально уникнути блокувань на стороні сервера.Дані передають по мережі в зашифрованому вигляді.

1. **Короткий опис системи, в якій реалізований проєкт**

Курсовий проєкт – це клієнт-серверне застосування. Нашою командою було реалізовано два варіанти клієнтської частини.

В першому варіанті ми додавали front end написаний за допомогою HTML, CSS, JS. Для цього ми сторювали 2 сервери. Перший сервер спілкується з фронтом як з клієнтом, обробляє запити з JS та відповідає на них по HTTP протоколу. Інформацію (відповідь) він отримує від сполучення з другим сервером, в цьому випадку сервер1 є клієнтом, а сервер2 приймає та обробляє запити. Вони спілкуються по TCP протоколу.

Також був реалізований додатковий обмежений інтерфейс за допомогою бібліотеки Java swing. Він передає запити на сервер2 напряму, без участі сервер1.

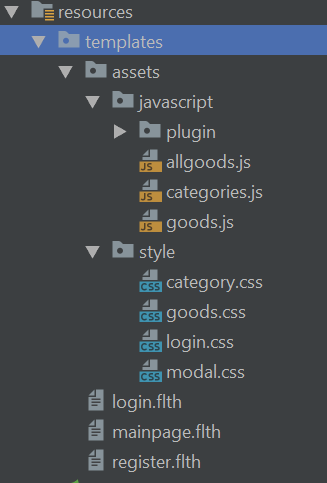
Майже всі запити - це робота з базою даних, тож над нею теж була проведена плідна робота із метою оптимізації пошуку, зручності зберігання та використання даних.

1. **Розподіл ролей всередині команди**

* Front-end веб – Катя
* Swing interface – Макс
* Back-end - всі

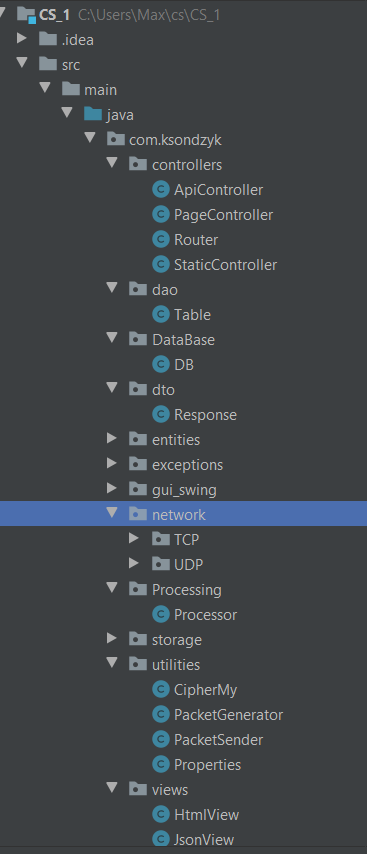
1. **Визначення класів**
   1. **Front end**

Весь проєкт базується на 3 основних сторінках, які рендеряться запитом на сервер. Цe login.flth, register.flth, mainpage.flth. До них підключені стилі і файли js наступним чином:



Categories.js відповідає за роботу з категоріями, це і реалізаціях всіх методів і перемальовка DOM дерева. Аналогічно з goods.js та allgoods.js

* 1. **Back end**

****

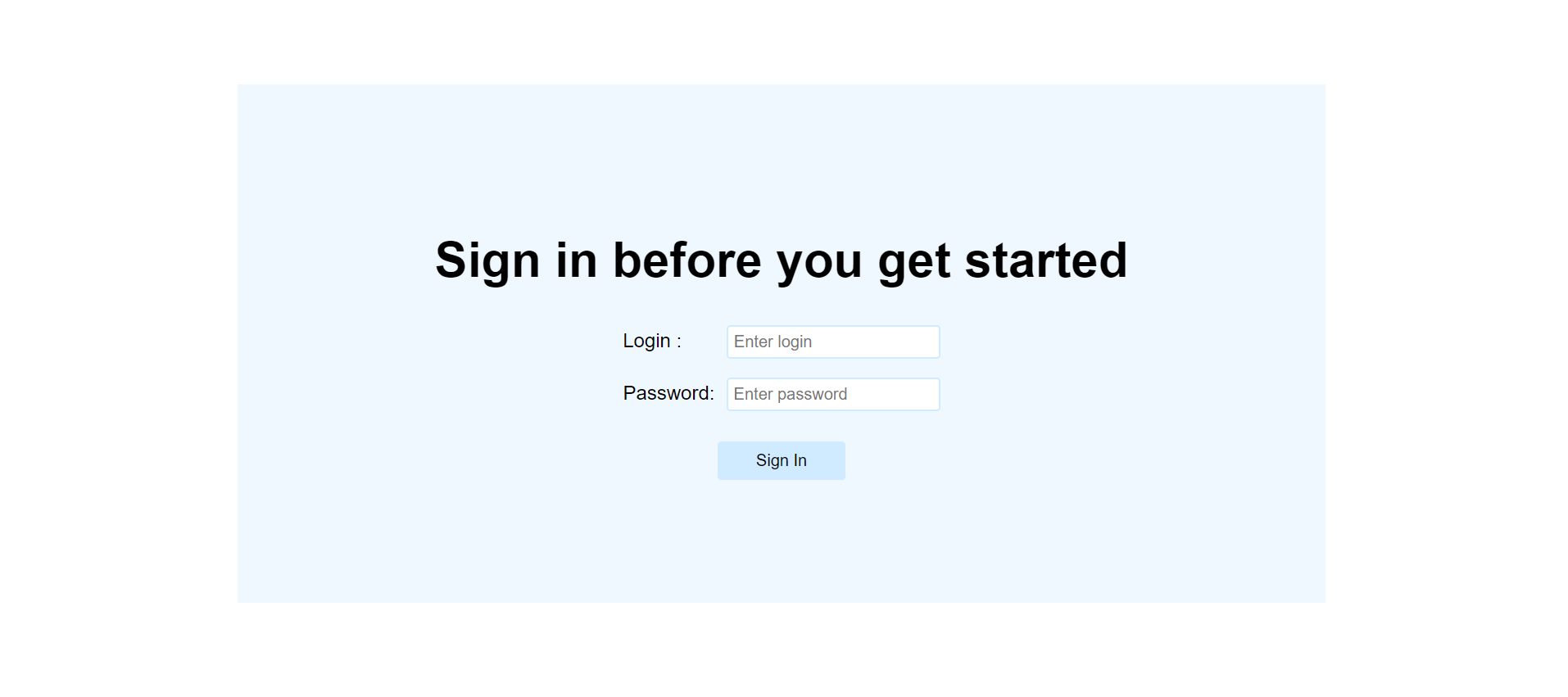
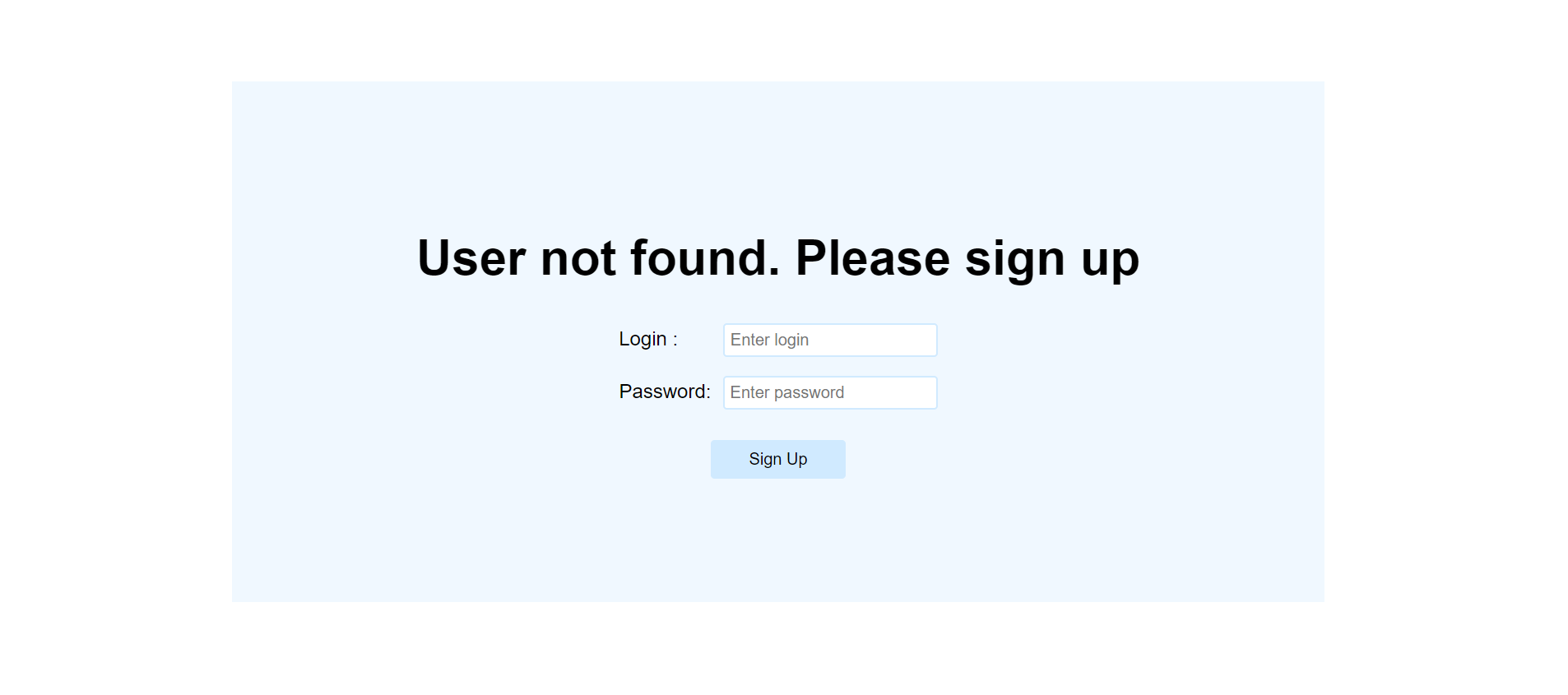
**Зображення, що містить стіл, малювання

Автоматично згенерований опис**

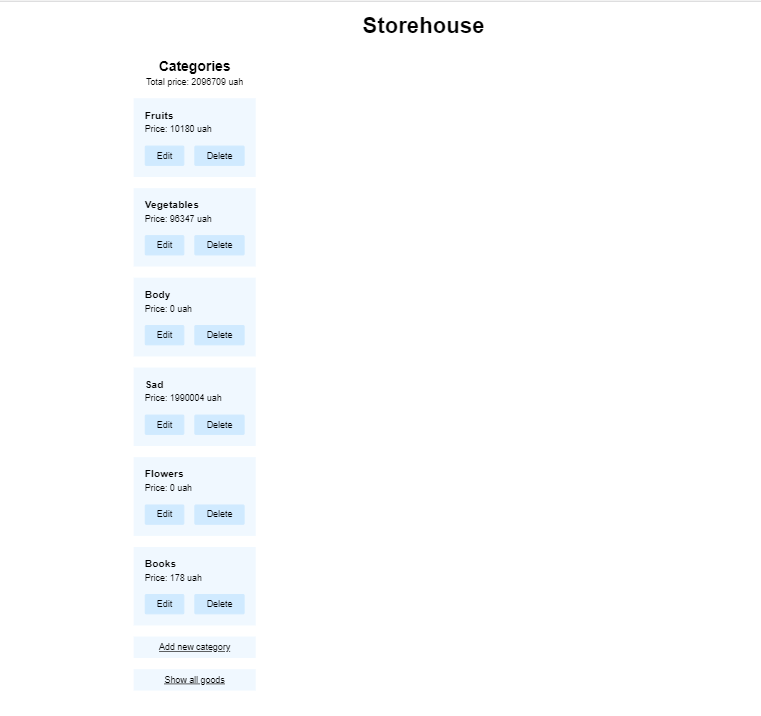
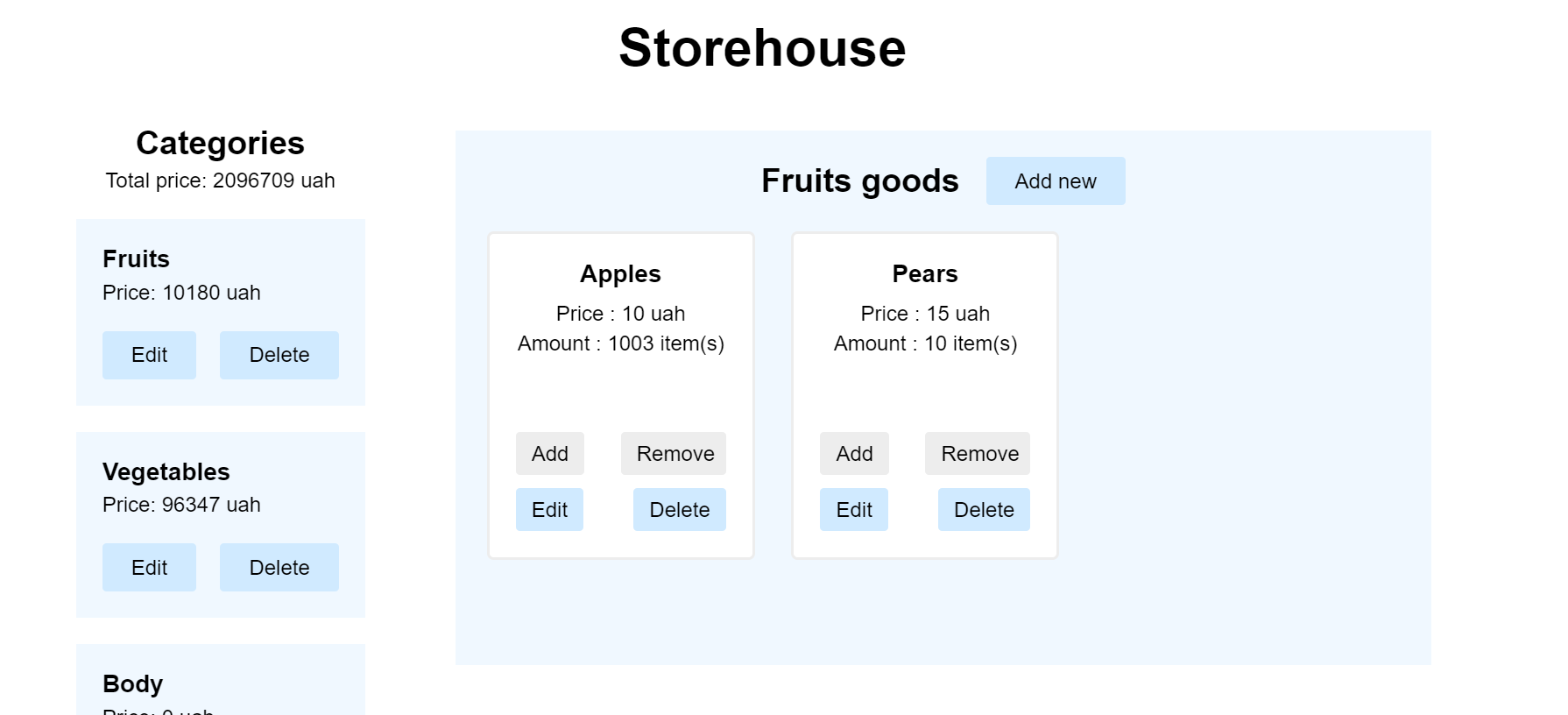
Архітектура застосунку

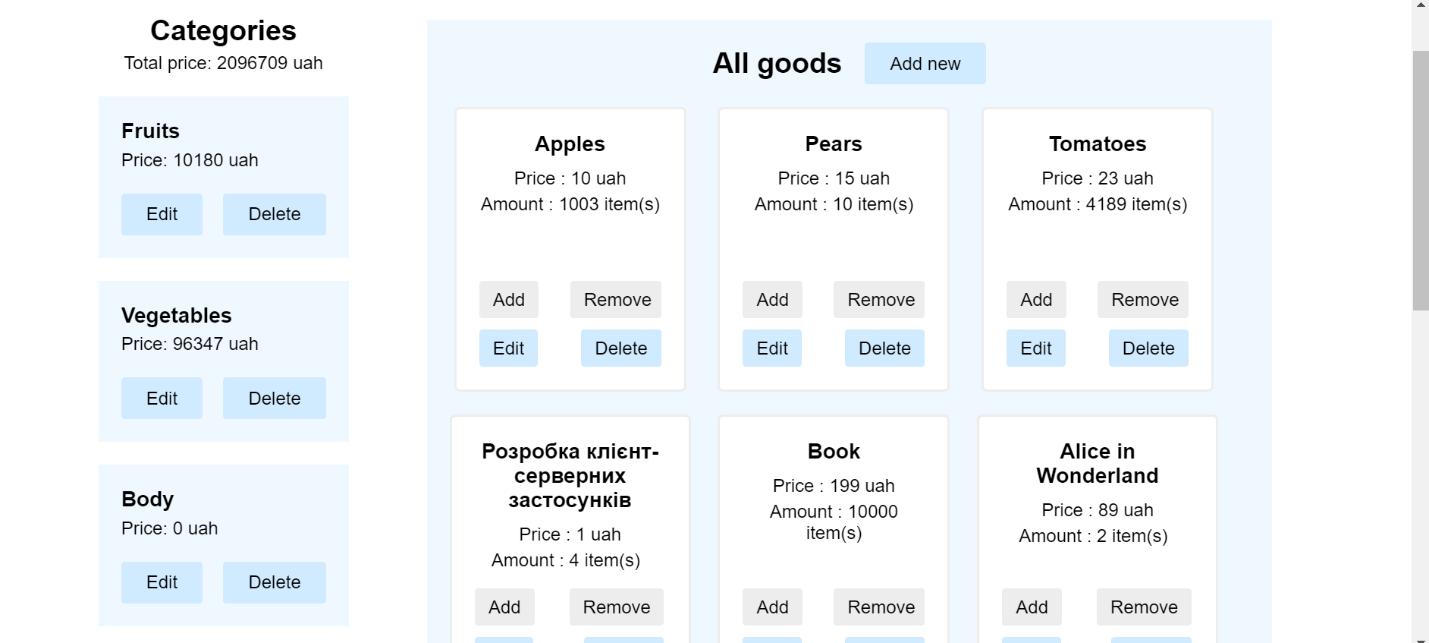
Ми використовуємо 2 сервери - HTTP сервер приймає запити безпосередньо від веб-клієнта. Router(interface httpHandler) відсилає запит на один з контролерів – PageController(відправляє сторінки), Static(відправляє статичні файли) або APIController(у форматі json надсилає інформацію з бази даних). Аби отримати будь-які дані контроллери надсилають повідомлення до внутрішнього серверу, котрий, в свою чергу, звертається до бази даних через dao Table.

1. **Опис GUI**

Користувач заходить на сайт і бачить перед собою вікно авторизації. Вводить логін і пароль. Якщо такий юзер вже існує, відкривається головна сторінка складу. Якщо такого юзеру немає – сторінка реєстрації.

На головній сторінці можна переглянути які є категорії. Побачити загальну вартість всіх товарів на складі, та окремо по кожній категорії. Також можна додавати/змінювати/видаляти категорію.

По розділам кожної категорії можна подивитись товари. Там також доступні такі методи, як додавання кількості на склад, списання, видалення всього товару, зміни, додавання нового товару.

Окремо є вікно, де можна переглянути всі товари та виконати пошук за назвою. В цьому вікні також працюють всі методи по роботі з товарами, які були зазначені раніше.



1. **Висновок**

Розробка клієнт серверних застосувань на Java є безумовно найбільш насиченим та одним з найкорисніших предметів за весь час навчання. Завдяки викладачу курс став ще й дуже зрозумілим, хоча матеріалу багато, а часу мало.

Дуже цікаво було збирати проєкт по частинам впродовж всього триместру і в результаті отримати все як у фільмах. Також сподобалось бачити графічний результат роботи. Працюючи на справжньому сайті отримувати не просто “response” від сервера, а красиво картинку з інформацією.

Новою для нас стала робота з базами даних. Складно було зрозуміти, як має виглядати правильно організована робота з бд, що таке дто та як цим користуватися. Деякі проблеми стало видно вже на фінальному етапі (як от те, що delete group не видаляє продуктів групи) та доводилося перероблювати запити до бд вручну.

Робота та взаємодія одразу 2 серверів – те, що було важко уявити на початку. Доводилося по кілька разів переписувати класи які вже були «готові» бо при підключені виникали нові проблеми та помилки.

Ще тим випробуванням стало підключення front end на js. Для валідного рендеру сторінок довелося прописувати багато функцій та запитів, а для зчитування файлів певного розширення цілих класів-контролерів. Це все іноді вганяло нас в паніку, проте проєкт на виході викликає посмішку та задоволення.

1. **Додаток. Вихідні тексти програми**

Посилання на гіт хаб : https://github.com/maxksondzyk/CS\_1/tree/develop\_dz5