Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут ім. Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

**Курсова робота**

з дисципліни «Інженерія програмного забезпечення»

на тему: «Склад: електронний облік товарів»

Виконав:

студент 2 курсу ФІОТ

групи ІО-24

Довгань М. С.

Керівник:

старший викладач

кафедри ОТ

Васильєва М. Д.

Київ - 2024

**ЗМІСТ**

ВСТУП………………………………………………………………………… 3

ТЕХНІЧНЕ ЗАВДАННЯ……………………………………………………... 5

1. ОГЛЯД MVC……………………………………………………………….. 6

1.1 Загальна характеристика…………………………………………... 6

1.2 Функції компонентів MVC………………………………………... 6

2. ПРОЄКТУВАННЯ ПРОГРАМНОГО ДОДАТКУ………………………... 8

2.1. Технології розробки………………………………………………..8

2.2. Можливі прецеденти програмного додатку……………………..14

3. РОЗРОБКА ПРОГРАМНОГО ДОДАТКУ

4. ТЕСТУВАННЯ ПРОГРАМНОГО ДОДАТКУ…………………………

4.1. Загальні можливості користувача………………

4.2. Сторінка «Категорії»…………………………

4.3 Сторінка «Товари»………………………………………………... 10 4.4 Сторінка «Постачальники»………………………………………. 12

ВИСНОВКИ………………………………………………………………..

ПЕРЕЛІК ПОСИЛАНЬ…….

ДОДАТКИ…………..

**ВСТУП**

Сучасний світ невпинно розвивається, і, хоч інформаційні технології (ІТ), з’явилися відносно нещодавно, вони вже стали невід’ємною частиною життя мільйонів, або, навіть, мільярдів людей. Вони проникають у всі сфери діяльності значно спрощуючи та оптимізуючи велику кількість процесів. Не виключенням із цього є й один із таких напрямків - ведення обліку товарів на складах.

На протязі великого часу традиційними методами обліку для людей дотепер були паперові документи, які мають цілий ряд значних недоліків, серед яких втрата - вони можуть легко загубитися або пошкодитися, доступність та мобільність - документи є доступними лише в місці їх зберігання, витрати на обробку та зберігання - для великої кількості документів необхідно чимало простору задля їх належного зберігання, також необхідно враховувати вартість паперу, офісних канцтоварів, інструментів для друку та обробки, також, немало важливою проблемою є безпека та конфіденційність цих паперів - їх фізична копія властива до крадіжки та несанкціонованого доступу та, у подальшому, витоку конфіденційної інформації, і, можливо, найголовніший мінус звичайних документів - їх обмежена можливість пошуку та організації, у разі великих обсягів документів це може бути дуже часо- та ресурсозатратним процесом, можна навести ще велику кількість інших негативних аспектів роботи з паперовими документами. Електронне діловодство - тобто, у нашому випадку, облік товарів на складі дозволяє значно підвищити ефективність та точність роботи, забезпечуючи швидкий доступ до актуальних даних та інформації, дуже спрощуючи процес інвентаризації.

Об’єктом розробки даної курсової роботи є створення універсального складу для ведення електронного обліку товарів, який може використовуватися майже для будь-яких категорій даної сфери та має інтуїтивно зрозумілий графічний інтерфейс користувача (GUI). Розроблене програмне забезпечення дозволить автоматизувати та оптимізувати процес обліку товарів, а також зручний доступ до необхідної інформації в режимі реального часу.

Крім того, дана курсова робота слугуватиме для перевірки та закріплення знань, отриманих під час вивчення дисципліни «Інженерія програмного забезпечення». Вона дозволить застосувати теоретичні та практичні навички у розробці програмного забезпечення, а також, можливо, змусить вивчити деякі нові технології та підходи.

Таким чином, створення складу для електронного обліку товарів є актуальною на цей день задачею, яка дозволить значно підвищити ефективність роботи та отримати практичний досвід у розробці програмного забезпечення.

**ТЕХНІЧНЕ ЗАВДАННЯ**

Склад: електронний облік товарів

**Функціональні вимоги до програмного забезпечення:**

1. Управління категоріями:

1.1. Можливість додавати категорію.

1.2. Можливість видаляти категорію.

1.3. Можливість змінити категорію.

1.4. Можливість перегляду категорії.

1.5. Можливість перегляду всіх категорій.

2. Управління товарами:

2.1. Можливість додавати товар у категорію.

2.2. Можливість видаляти товар з категорії.

2.3. Можливість змінювати дані товару.

2.4. Можливість змінювати кількість товару на складі.

2.5. Можливість переглянути дані конкретного товару.

2.6. Можливість переглянути список всіх товарів:

2.6.1. Можливість відсортувати список по назві.

2.6.2. Можливість відсортувати список по бренду.

2.6.3. Можливість відсортувати список по ціні.

3. Управління постачальниками:

3.1. Можливість додавати постачальників.

3.2. Можливість видаляти постачальників.

3.3. Можливість змінювати дані постачальників.

3.4. Можливість переглянути дані конкретного постачальника.

3.5. Можливість переглянути список всіх постачальників:

3.5.1. Можливість відсортувати список по імені.

3.5.2. Можливість відсортувати список по прізвищу.

4. Пошук:

4.1. Можливість пошуку по ключовому слову серед товарів.

4.2. Можливість пошуку по ключовому слову серед замовників.

**1. ОГЛЯД MVC**

1.1. Загальна характеристика.

MVC (model-view-controller (модель-вигляд-контролер)) - це архітектурний шаблон, який використовується під час проєктування та розробки програмного забезпечення.

Даний шаблон передбачає поділ системи на три взаємопов’язані частини - модель даних, вигляд та модуль керування. Застосовується для відокремлення даних від інтерфейсу користувача так, щоб зміни інтерфейсу користувача мінімально впливали на роботу з даними, а зміни в моделі даних могли здійснюватися без змін інтерфейсу користувача.

Основна мета цього шаблону - гнучкий дизайн програмного забезпечення, який повинен полегшувати подальші зміни чи розширення програм, а також надавати можливість повторного використання окремих компонентів програми. Використання даного шаблону у великих системах також сприяє впорядкованості їхньої структури та робить їх більш зрозумілими шляхом значного зменшення складності.

1.2. Функції компонентів MVC.

У рамках архітектурного шаблону модель–вигляд–контролер (MVC) програма поділяється на три окремі, але взаємопов'язані частини з розподілом функцій між компонентами. Модель (Model) - частина, яка відповідає за зберігання даних та їхню структуру. Вигляд (View) - частина, яка відповідальна за представлення цих даних користувачеві, тобто інтерфейс програми. Контролер (Controller) - частина, яка керує компонентами, отримує сигнали у вигляді реакції на дії користувача (зміна положення курсора миші, натискання кнопки, ввід даних в текстове поле тощо) і передає дані у модель.

Модель є центральним компонентом шаблону MVC та відображає поведінку застосунку, незалежну від інтерфейсу користувача. Модель стосується прямого керування даними, логікою та правилами застосунку.

Модель поділяється на два типи: активна та пасивна.

Активна модель - вигляд відстежує зміни в моделі та реагує на них.

Пасивна модель - вигляд оновлюється через контролер.

Вигляд може являти собою будь-яке представлення інформації, одержане на виході, наприклад графік чи діаграму. Одночасно можуть співіснувати кілька виглядів (представлень) однієї й тієї ж інформації, наприклад гістограма для керівництва компанії й таблиці для бухгалтерії.

Контролер одержує вхідні дані й перетворює їх на команди для моделі чи вигляду.

**2. ПРОЄКТУВАННЯ ПРОГРАМНОГО ДОДАТКУ**

2.1. Технології розробки.

Дану курсову роботу буде виконано (написано) на наступних мовах веб-розробки:

HTML (Hyper Text Markup Language) - мова розмітки, яка визначає структуру і семантичний зміст веб-сторінки.

CSS (Cascading Style Sheets) - мова стилів, що відповідає за візуальне оформлення і макет веб-сторінки.

PHP (Hypertext Preprocessor) - це серверна мова програмування, яка використовується для створення динамічних веб-сайтів та веб-додатків.

Front-end сторінки, який відповідає за візуалізацію та взаємодію на стороні клієнта (браузера), а також відображає інтерфейс для користувача написаний на HTML і CSS.

Back-end сторінки, який взаємодіє з базами даних та відповідає за обробку логіки додатку написаний на PHP.

2.2. Можливі прецеденти програмного додатку.

Прецедент (use case) являє собою послідовність дій, які виконує система під час взаємодії з актором (у нашому випадку користувачем) для досягнення певної мети. Виключні ситуації (винятки) в програмних додатках виникають через різні непередбачені обставини або помилки, які можуть статися під час виконання програми. Розробка таких прецедентів допомагає зрозуміти та зафіксувати вимоги до функціоналу системи з точки зору її майбутніх користувачів. Наступні описані діаграми прецедентів, створені за допомогою сайту [plantuml.com](http://plantuml.com), на яких можна наглядно побачити на прикладі такі ситуації, які можуть виникнути під час взаємодії актора (користувача) із розробленим нами програмним додатком (системою) будуть слугувати основою для подальшого проєктування системи.

На рис. 1.1 ми можемо бачити діаграму прецедентів на вкладці «Категорії» сайту. Тут може бути декілька виключних ситуацій. Перша виключна ситуація: коли користувач не заповнив назву категорії, яку хоче додати, тоді система видасть попередження «Заповніть це поле!» із вказівкою до поля «Назва категорії» для введення інформації про назву категорії. Тобто, користувач може додати нову категорію, без її опису, але не може додати порожню категорію з її описом. Друга виключна ситуація: при натисканні користувачем кнопки «Додати» виконується перевірка назви категорії для унікальності (неповторності) додавання з уже існуючими в базі даних - якщо ця перевірка проходить успішно, і з такою ж назвою категорії не існує, то вона успішно додається, в іншому ж випадку, видається повідомлення «Така категорія вже існує!», що, відповідно, не додає її до бази даних.

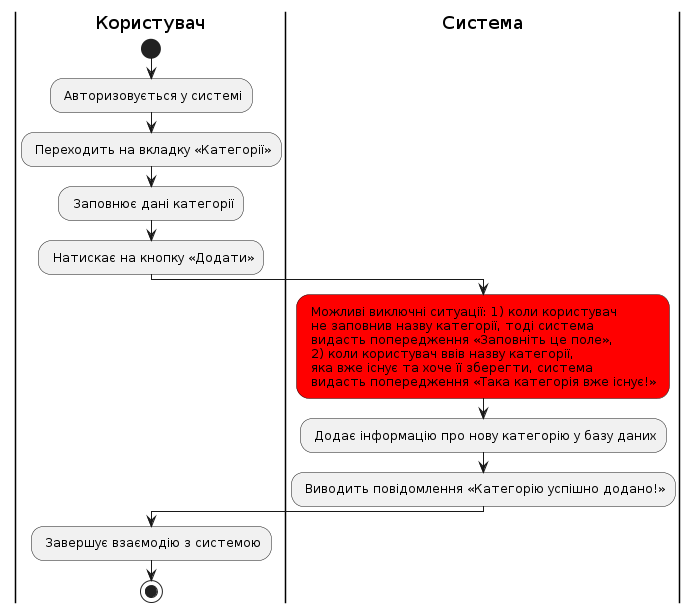


Рис. 1.1 Діаграма прецедентів на вкладці «Категорії».

На рис. 1.2 ми можемо побачити діаграму прецедентів на вкладці «Товари». Тут також можливі дві виключні ситуації. Перша виключна ситуація: коли користувач не обрав щось із переліченого: категорію товару, назву, опис, бренд, тоді система видасть попередження «Заповніть це поле» із вказівкою на поле, яке користувачу необхідно заповнити. Друга виключна ситуація: коли користувач указав кількість на складі або ціну від’ємним числом (0 >), у такому випадку система попередить користувача повідомленням «Значення має бути більшим або дорівнювати 0» із, відповідно, вказівкою на поле для введення числа.

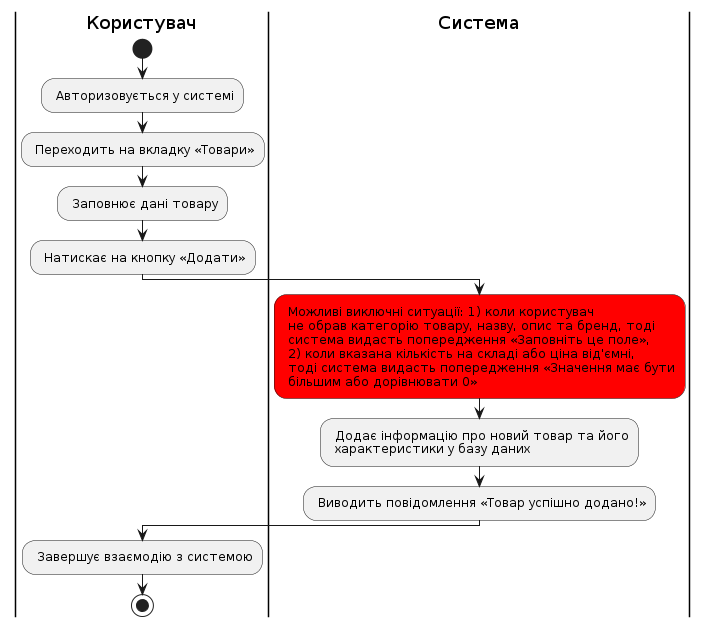


Рис. 1.2 Діаграма прецедентів на вкладці «Товари».

На рис. 1.3 ми можемо побачити діаграму прецедентів на вкладці «Постачальники». З діаграми видно, що у нас можливі дві виключні ситуації. Перша виключна ситуація: користувач не ввів дані про ім’я, прізвище, адресу при заповненні інформації про постачальника, тоді система видасть повідомлення «Заповніть це поле» із посиланням на незаповнене поле. Друга виключна ситуація: користувач ввів неправильний формат номеру, тоді система попередить його повідомленням «Введіть номер у форматі +380ХХХХХХХХХ!».

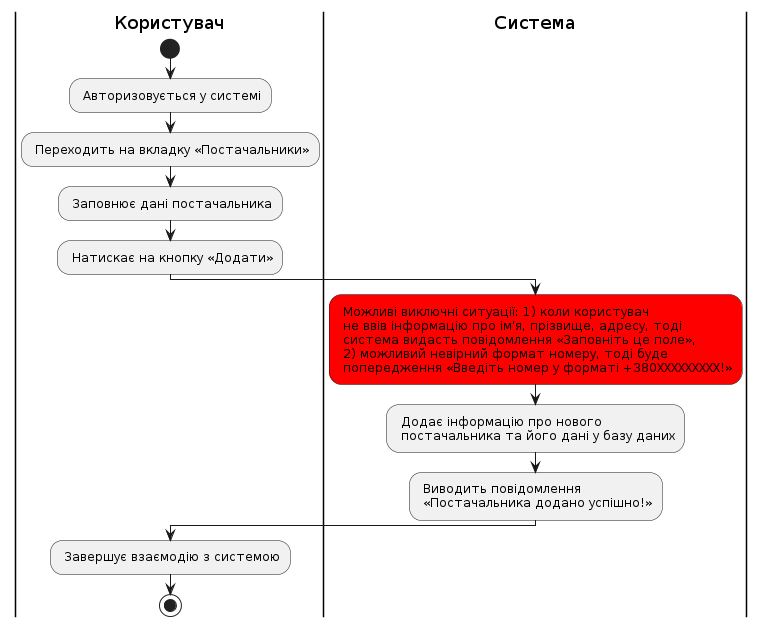


Рис. 1.3 Діаграма прецедентів на вкладці «Постачальники».

На рис. 1.4 ми можемо побачити діаграму прецедентів на вкладці «Пошук». Тут можлива лише одна виключна ситуація, коли користувач не ввів ключове слово для пошуку, тоді система виведе повідомлення біля комірки для вводу ключового слова із повідомленням «Заповніть це поле». Коли ж він вводить ключове слово, яке міститься в назві або описі товару, або в імені, прізвищі чи адресі постачальника, система приймає ці дані та проводить пошук по своїй базі даних, якщо введена користувачем інформація існує, та співпадає з якимось із постачальників або товаром, система його виводить, якщо ж ні - повідомляє про це повідомленням «Нічого не знайдено».



Рис. 1.4 Діаграма прецедентів на вкладці «Пошук».

**3. РОЗРОБКА ПРОГРАМНОГО ДОДАТКУ**

Для розробки програмного додатку було використано одразу декілька інструментів та технологій:

3.1. Visual Studio Code:

Visual Studio Code - це популярне середовище розробки, яке підтримує різні мови програмування та має широкий спектр розширень для зручної роботи з кодом. VS Code має велику кількість розширень, які дозволяють додавати нові функції та інтеграції з іншими інструментами, вони включають в себе підтримку різних мов програмування, інструментів для відлагодження, систем контролю версій та багато іншого. VS Code має вбудовану підтримку Git, що дозволяє легко робити коміти, створювати гілки та вирішувати конфлікти безпосередньо в середовищі розробки. Він забезпечує зручність спільної роботи над проектом та відслідковування історії змін. Інструменти для відлагодження дозволяють ефективно знаходити та виправляти помилки, переглядати змінні та виконувати код по кроках, що, у деяких ситуаціях, дуже виручає розробників.

3.2. XAMPP:

XAMPP - це повнофункціональний пакет програмного забезпечення із відкритим вихідним кодом, який дозволяє створювати локальне веб-серверне середовище для розробки, тестування та налагодження веб-додатків. Він включає в себе такі компоненти, як Apache, MySQL, PHP та деякі інші, що робить його ідеальним вибором для розробників.

Вся інформація, яку вводить та зберігає користувач зберігається у створеній базі даних, яка є критично важливою для забезпечення надійного, безпечного та ефективного зберігання та управління даними, необхідними для належного функціонування сучасного веб-сайту.

У нашому випадку, при запуску веб-інтерфейсу користувача (Control Panel) XAMPP, який дозволяє легко запускати, зупиняти та керувати компонентами, ми активуємо модулі локального веб-сервера Apache, що дозволяє розробникам запускати та тестувати веб-сайти та додатки безпосередньо на своїх комп’ютерах, без необхідності підключатися до віддаленого серверу та СУБД - систему управління базами даних phpMyAdmin, веб-інтерфейс для MySQL.

3.3. phpMyAdmin:

phpMyAdmin - це веб-застосунок з відкритим вихідним кодом, призначений для адміністрування MySQL за допомогою веб-інтерфейсу. Він дозволяє розробникам і адміністраторам баз даних легко ними керувати, виконуючи різні операції, такі як створення, редагування та видалення таблиць, виконання SQL-запитів та імпорт/експорт даних. phpMyAdmin надає зручний і інтуїтивно зрозумілий графічний інтерфейс користувача, який спрощує управління базами даних. Це дозволяє користувачам виконувати складні SQL-запити та операції без необхідності використання командного рядка. Він забезпечує швидкий доступ до всіх функцій MySQL, включаючи створення та редагування таблиць, індексів, користувачів та прав доступу. phpMyAdmin дозволяє керувати користувачами та їхніми правами доступу до баз даних, що забезпечує високий рівень безпеки та контролю, і взагалі є зрозумілим у використанні, навіть, у моєму випадку, для початківців.

**4. ТЕСТУВАННЯ ПРОГРАМНОГО ДОДАТКУ**

4.1. Загальні можливості користувача.

Загалом, можливостями користувача є управління категоріями (додавати категорію, видаляти, змінювати, переглядати), управління товарами (додавання товару, видалення, зміна даних, кількості, перегляд, сортування за назвою, брендом, ціною), управління постачальниками (додавання постачальників, видалення, зміна даних та контактів, перегляд, сортування за іменем, прізвищем) та пошук (по ключовому слову серед товарів та замовників).

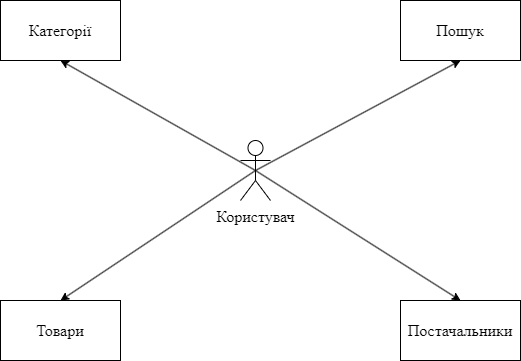


Рис. 4.1 Загальні можливості користувача

4.2. Сторінка «Категорії».

Під час переходу на сторінку «Категорії» користувач бачить перед собою сторінку, умовно розділену на дві частини - «Додати нову категорію» та «Список всіх категорій» - заповнивши усі необхідні поля («Назва категорії» та «Опис категорії») користувач тисне кнопку додати, і, щойно створена ним категорія додається до лівої частини - списку всіх категорій. У даному списку присутні всі категорії, які були створені та додаті до неї з їх назвою та описом, які легко змінюються - необхідно заповнити верхнє поле - нову назву та нижнє поле - новий опис категорії і натиснути на зелений олівець, щоб внесені зміни були збережені та внесені до бази даних. Видаляються категорії так само легко - варто всього лише натиснути на червону кнопку кошику, яка видаляє обрану користувачем категорію разом із її описом.

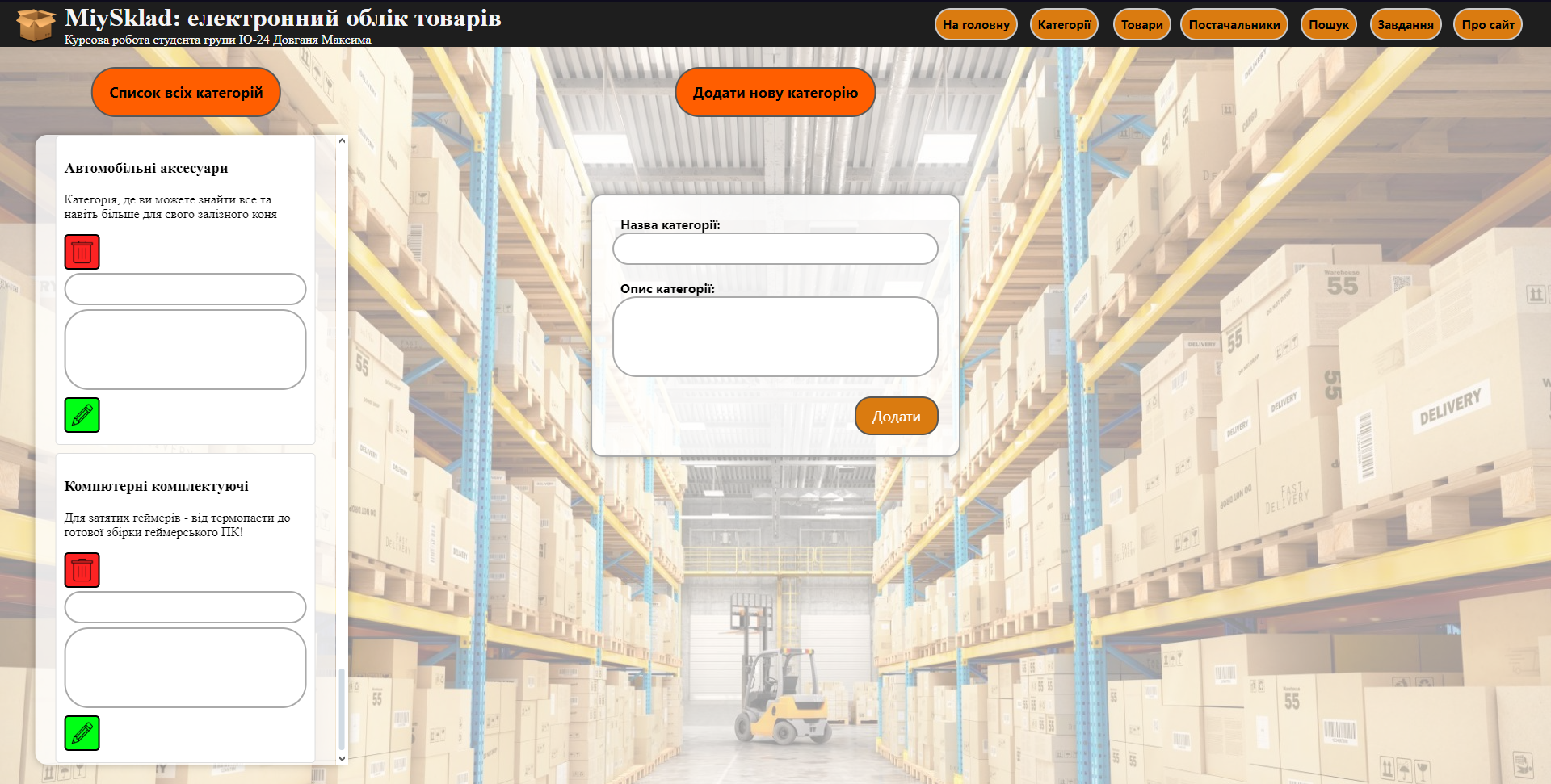


Рис. 4.2 Сторінка «Категорії»

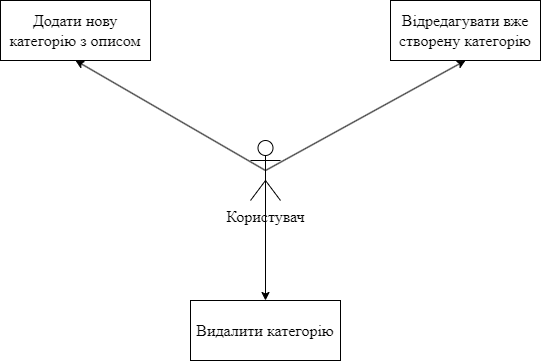


Рис. 4.3 Можливості користувача на сторінці «Категорії»

4.3. Сторінка «Товари».

Після переходу користувача до сторінки «Товари» він бачить перед собою схожу картину - все та ж умовно поділена на дві частини сторінка, але вже з іншим функціоналом - тут користувач може додати товар до заздалегідь створеної категорії - алгоритм дій наступний: спочатку обирається категорія, до якої ми хочемо додати товар у випадному списку - списку з усіх створених категорій, потім заповнюється назва товару, опис, кількість одиниць на складі, бренд товару та його ціна. Після натискання кнопки «Додати» він автоматично додається до бази даних і з’являється у лівій частині сторінки - де ми можемо видалити його, натиснувши на знайомий червоний кошик, відредагувати, попередньо заповнивши всі дані та натиснувши на зелений олівець, відсортувати список усіх товарів за назвою, за допомогою помаранчевих стрілок, відсортувати за брендом - клік на синю кнопку «R» та відокремити товари за їх вартістю - за допомогою фіолетового значка долара.

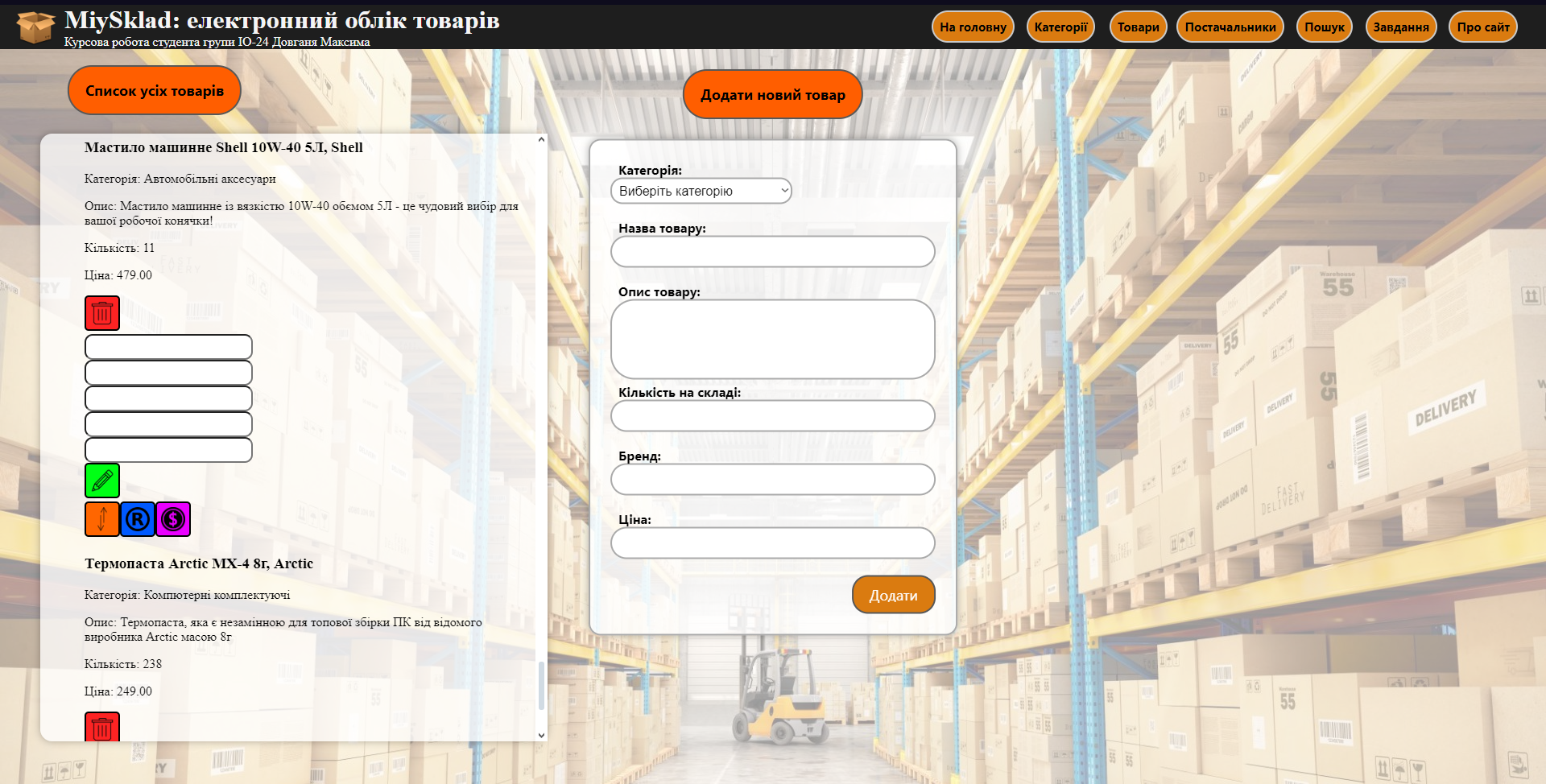


Рис. 4.4 Сторінка «Товари»

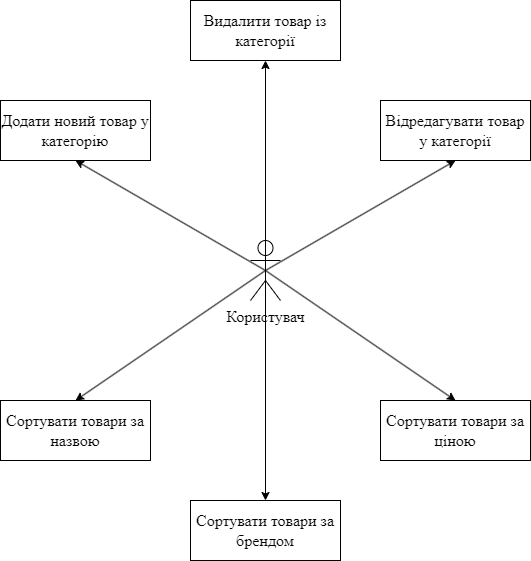


Рис. 4.5 Можливості користувача на сторінці «Товари»

4.4. Сторінка «Постачальники».

При переході на сторінку «Постачальники» ми бачимо меню для додавання нової людини - це ми може зробити, заповнивши його дані: ім’я, прізвище, адреса, номер телефону, після чого натиснути кнопку «Додати» і він потрапить до бази даних, що ми можемо бачити його візуально, у списку всіх постачальників. В цьому списку ми можемо видалити постачальника - червона кнопка кошика, відредагувати його дані, перед цим заповнивши всі поля та клікнути на зелений олівець, також ми можемо відсортувати всіх людей за ім’ям - для цього варто натиснути на синю кнопку, а помаранчева кнопка, у свою чергу, сортує їх за прізвищем.

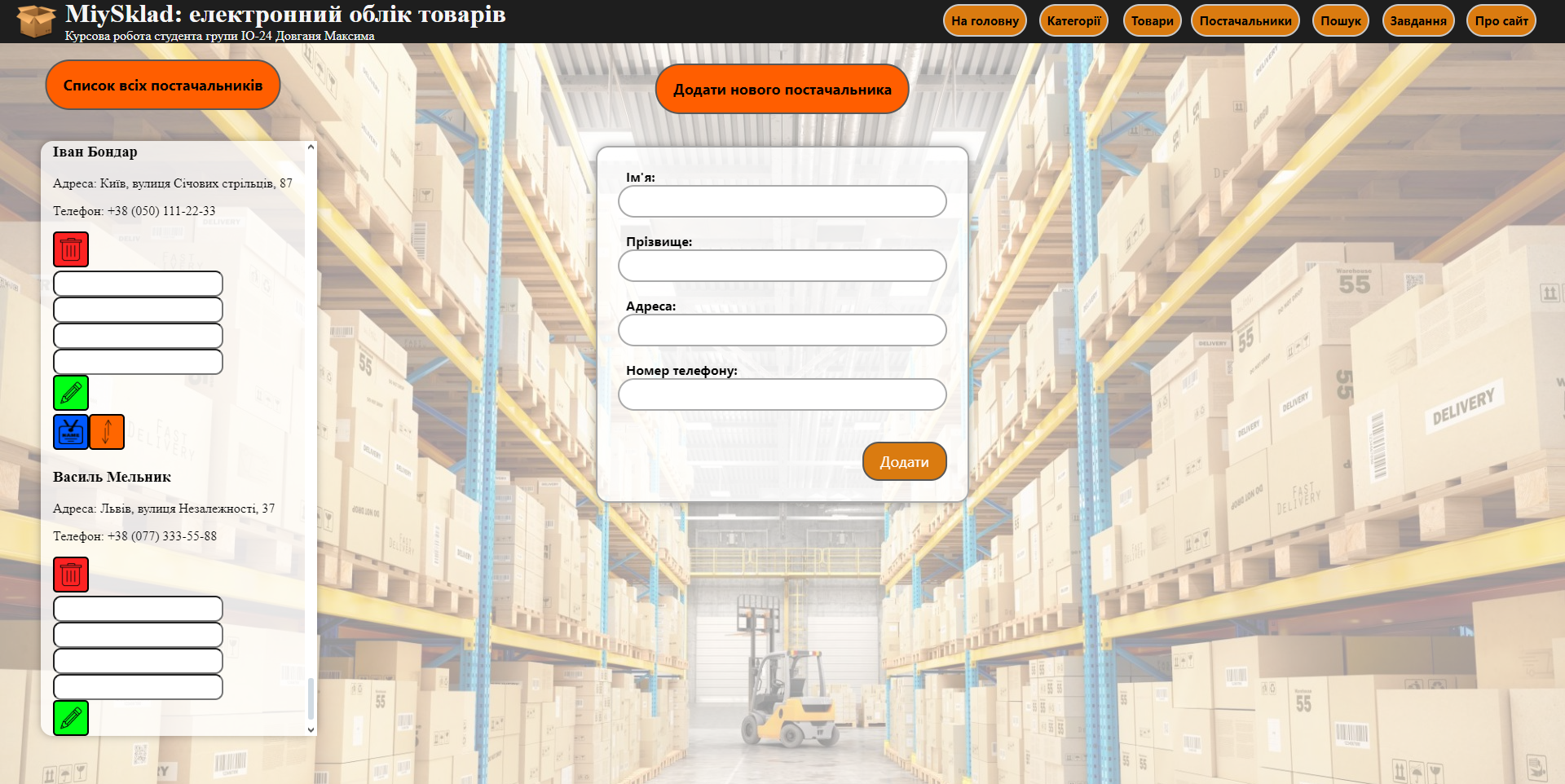


Рис. 4.6 Сторінка «Постачальники»

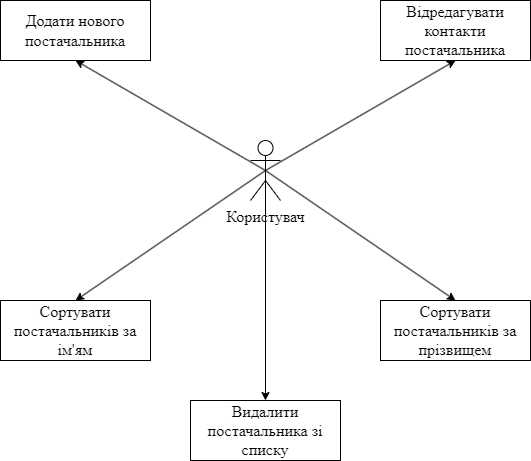


Рис. 4.7 Сторінка «Постачальники»

**ВИСНОВКИ**

У результаті виконання курсової роботи мною був створений склад, в якому ведеться електронний облік товарів. У наш час електронний облік стає дедалі поширенішим, але навіть зараз у деяких компаніях та корпораціях облік ведеться звичним способом, за допомогою документів, які є непрактичними. З метою вирішення цієї проблеми мною і була обрана саме ця тема для курсової роботи.

Створений мною проєкт дозволяє звичайному користувачеві розібратися в ньому, завдяки user-friendly графічному дизайну та інтуїтивності. Електронний облік у такому виді дозволяє економити велику кількість часу, сил та нервів працівникам, які відповідальні за облік будь-яких видів товару, особливо у значних кількостях.

Розроблений проєкт дозволяє своєму користувачеві створювати, редагувати та видаляти категорії продуктів і товарів. Товари ж, у свою чергу, можна додати до вже створених категорій, у яких їх можна перейменовувати, змінювати бренд, опис, кількість та ціну, а також проводити сортування за назвою, брендом та ціною, що значно полегшує орієнтацію в них. Також присутня можливість додавання постачальників, із їхнім іменем, прізвищем, адресою та номером телефону, та їх сортування. І, наостанок, реалізовано функцію пошуку, як по товарам, так і постачальникам, користувачу достатньо всього лише обрати бажану категорію, в якій він хоче виконати пошук та ввести ключове слово, яке міститься в шуканому товарі, або постачальнику.

Даний електронний облік товарів може бути використаний майже у будь-якій сфері діяльності, де необхідно вести облік. Для більшої гнучкості його можна оптимізувати та додати нові функції під побажання кінцевого користувача / замовника, для найкращого досвіду використання.

**ПЕРЕЛІК ПОСИЛАНЬ**

1. Інженерія програмного забезпечення: методичні вказівки до виконання лабораторних робіт для студентів спеціальності 123 «Комп’ютерна інженерія» Ч. І. Структурні шаблони. / Уклад.: А. О. Болдак, О. Н. Абу Усбах. - К.: НТУУ «КПІ», 2011. - 40 с.
2. Об’єктно-орієнтоване програмування: конспект лекцій для студентів напряму підготовки «Комп’ютерні науки» всіх форм навчання / Ю. Е. Парфьонов, В. М. Федорченко, М. Ю. Лосєв, О. В. Щербаков. – Харків : Вид. ХНЕУ, 2010. – 312 с. (Укр. мов.) стр. 3
3. Бублик В.В. Б90 Об’єктно-орієнтоване програмування: [Підручник] / В.В. Бублик. – К.: ІТ-книга, 2015. – 624 с.: іл. ISBN 978-966- 97182-1-1, стр. 10-22
4. Бандура, В. В. Архітектура та проектування програмного забезпечення конспект лекцій / В. В. Бандура, Р. І. Храбатин. - Івано-Франківськ: ІФНТУНГ, 2012. – 4-10 с.