



## Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту  
(повна назва)Кафедра Інформатики  
(повна назва)Рівень вищої освіти перший (бакалаврський)Спеціальність 122 Комп'ютерні науки  
(код і повна назва)Тип програми освітньо-професійнаОсвітня програма Інформатика  
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

«\_\_\_\_» \_\_\_\_\_ 2023 р.

**ЗАВДАННЯ**  
НА КВАЛІФІКАЦІЙНУ РОБОТУстудентові Гавришевій Камілі Амівні  
(прізвище, ім'я, по батькові)1. Тема роботи Розробка програмного застосунку для ведення обліку з продажу товару

затверджена наказом університету від 15 травня 2023 року № 474 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 29 травня 2023 р.

3. Вихідні дані до роботи науково-методична та науково-технічна література, матеріали конференцій, дані інтернет-мережі, документація мови програмування, бібліотек та фреймворків.

4. Перелік питань, що потрібно опрацювати в роботі \_\_\_\_\_

1. Огляд програм для ведення обліку з продажу товару.2. Вимоги до складського застосунку.3. Огляд технологічних рішень для розроблення ПЗ.4. Моделювання програмного застосунку для ведення обліку з продажу товару.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Огляд програм для ведення обліку з продажу товару, актуальність проблеми збору, обробки та зберігання даних, постановка задачі, моделювання розроблення програмного застосунку, програмна реалізація вебзастосунку, тестові зображення.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Консультант з дотримання діючих стандартів та норм	Доцент Творошенко І.С.		

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	10.04.2023	
2	Аналіз завдання, підбір літератури	11.04.23-13.04.23	
3	Аналіз літератури з досліджуваної проблеми	14.04.23-17.04.23	
4	Огляд технологічних рішень	18.04.23-23.04.23	
5	Розроблення моделі застосунку	24.04.23-29.04.23	
6	Програмна реалізація	30.04.23-10.05.23	
7	Оформлення пояснювальної записки	11.05.23-30.05.23	
8	Перевірка на плагіат	27.05.23	
9	Рецензування	28.05.23	
10	Підготовка презентації та доповіді	29.05.23-30.05.23	
11	Занесення роботи в електронний архів	31.05.23	
12	Попередній захист кваліфікаційної роботи	07.06.23	

Дата видачі завдання 10 квітня 2023 р.

Студент \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_ проф. Машталір С. В.  
(підпис) (посада, прізвище, ініціали)

## РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 70 с., 2 табл., 29 рис., 1 дод., 37 джерел.

СТАТИСТИКА ПРОДАЖУ ТОВАРУ, ОБЛІК, БАЗА ДАНИХ, ФІНАНСОВА ЗВІТНІСТЬ, JAVA, INTELLIJ IDEA, POSTGRESQL, SPRING FRAMEWORK.

Об'єкт роботи – це вебзастосунок для ведення обліку одиниць товару для продажу, які будуть мати електронне представлення для операцій над ними.

Метою роботи є розробка програмного застосунку для ефективного процесу ведення обліку з продажу товару, зокрема одягу, збереження даних про заміри, матеріали, кольору та інших параметрів.

Використано методи проєктування інформаційних систем, методи проєктування баз даних відповідно до реляційної моделі. Розроблено алгоритм збереження даних товару до бази даних та перегляду статистики відповідного товару, реалізовано програмні рішення за для перегляду даних одиниць товару із бази даних.

У результаті роботи розроблено програмний застосунок системи для ведення обліку з продажу товару, який надає функції реєстрації, зміни та видалення даних товару, а також перегляду статистичної інформації.

PRODUCT SALES STATISTICS, ACCOUNTING, DATABASE, FINANCIAL REPORTING, JAVA, INTELLIJ IDEA, POSTGRESQL, SPRING FRAMEWORK.

The object of work is the web-application with functionality of manipulating units of goods for sale, which have a data base representation for operations on them.

The purpose of the work is the development of a software application for the effective process of keeping records on the sale of goods, in particular clothes, saving data on measurements, materials, colors and other parameters.

Information systems design methods, database design methods according to the relational model are used. An algorithm for saving product data to the database and viewing the statistics of the corresponding product has been developed, and software solutions have been implemented for viewing product unit data from the database.

As a result of the work, a software application of the product sales accounting system was developed, which provides the functions of registering, changing and deleting product data, as well as viewing statistical information.

## ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів .....	6
Вступ.....	7
1 Проблематика збору, обробки та зберігання даних .....	8
1.1 Вплив комп'ютеризації на сферу заберігання та маніпулювання інформацією.....	8
1.2 Огляд програм для ведення обліку з продажу товару .....	11
1.3 Вимоги до складського застосунку.....	15
1.4 Постановка задачі .....	17
2 Програмне обґрунтування обраних технологій.....	19
2.1 Перша модель програмного рішення.....	19
2.1.1 Модель програми на базі технології Java Swing.....	20
2.1.2 Модель бази даних на сервері PostgreSQL.....	25
2.2 Друга модель програмного рішення .....	28
2.2.1 Модель програми на базі технологій Web та Java як back-end .....	28
2.2.2 Модель бази даних на сервері phpMyAdmin.....	32
2.3 Порівняння технологій та обґрунтування вибору моделі .....	35
3 Програмна реалізація вебзастосунку .....	36
3.1 Структура розробки вебзастосунку .....	36
3.2 Розробка Бази Даних у PostgreSQL.....	40
3.3 Розробка серверної частини.....	45
3.4 Інструкція користувача .....	55
3.5 Тестування розробленої моделі.....	59
Висновки .....	64
Перелік джерел посилання .....	65
Додаток А Тестові зображення.....	69

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,  
СКОРОЧЕНЬ І ТЕРМІНІВ**

ІС – інформаційна система

ПЗ – програмний застосунок

БД – база даних

СР – середовище розробки

ОЗ – область знань

ПрО – предметна область

ІТ – інформаційні технології

2D – двовимірний

## ВСТУП

Важливою складовою ефективною роботи є ведення обліку товарів та їх продажу. В сучасному світі, де віртуальний простір став невід'ємною частиною бізнесу, майже кожна компанія має власний інтернет-магазин або електронний каталог для представлення своєї продукції. Завдання зберігання та організації електронного каталогу товарів та ведення обліку їх продажу на віртуальному просторі стає дедалі актуальнішим.

Одним зі способів вирішення цієї проблеми є розробка програмного забезпечення, яке б допомагало управляти процесом продажу товару та контролювати наявність товару на складі. У рамках кваліфікаційної роботи було розроблено програмний застосунок для ведення обліку товару в електронному вигляді з метою полегшення контролю за процесом продажу.

В рамках даної кваліфікаційної роботи було розроблено та змодельовано програмний застосунок для ведення обліку товару та їх продажу, який надає можливість реєстрації, зміни та видалення даних товару, а також перегляд статистичної інформації про наявність товару на складі. Також було розроблено модель застосунку та програмну реалізацію мовою Java з інтерфейсом користувача, базою даних.

Для збереження даних про товар та їх продаж було розроблено алгоритм збереження даних до бази даних та перегляду статистики відповідного товару. В результаті було створено програмні рішення для перегляду даних одиниць товару із бази даних. Даний програмний застосунок системи для ведення обліку надає зручний та ефективний інструмент для управління продажами товарів та їх наявністю на складі.

В процесі виконання кваліфікаційної роботи було проведено аналіз сучасних програмних рішень для ведення обліку товарів та продажу їх. Були визначені основні проблеми та недоліки існуючих програм, що були використані в якості джерела інформації та порівняння з розробленим програмним застосунком.

# 1 ПРОБЛЕМАТИКА ЗБОРУ, ОБРОБКИ ТА ЗБЕРІГАННЯ ДАНИХ

## 1.1 Вплив комп'ютеризації на сферу зберігання та маніпулювання інформацією

Комп'ютеризація і цифрова революція охопили всі сфери життя людини і відчутно змінили світ. Новітні технології, які виникають і розвиваються швидкими темпами, дозволяють людям отримувати інформацію, виконувати завдання і спілкуватися за допомогою комп'ютерів, смартфонів і інших пристроїв [1–3].

Однією з галузей, які найбільше відчули вплив комп'ютеризації, є бізнес. Із ростом обчислювальних потужностей комп'ютерів, відбулися кардинальні зміни в системі ведення ділової діяльності. З'явилися програми для обліку, управління і контролю за бізнесом, що зробили його більш ефективним та прибутковим. Крім того, з'явилися інтернет-магазини, що стали більш доступними та зручними для користувачів.

Зараз електронні продажі є не тільки зручними, але і дуже перспективними напрямками бізнесу. Онлайн-магазини пропонують широкий асортимент товарів та послуг, швидку та зручну доставку і різні форми оплати. Електронна комерція зростає зі швидкістю світла, тому що все більше людей віддають перевагу онлайн-шопінгу. Більшість магазинів має свої вебсайти, де можна знайти велику кількість інформації про продукти та послуги.

Одним з головних переваг електронної комерції є можливість миттєвого обміну інформацією між продавцем і покупцем. За допомогою електронних листів, онлайн-чатів і телефонів, клієнти можуть спілкуватися з продавцями, задавати питання, отримувати консультації та отримувати відповіді на свої запитання в режимі реального часу. Це значно покращує комунікацію між продавцем і покупцем, що може призвести до збільшення продажів та задоволеності клієнтів.

Окрім того, електронна комерція дозволяє продавцям ефективніше вести маркетингову кампанію та рекламу своїх товарів. За допомогою електронних листів, рекламних банерів, контекстної реклами та соціальних медіаканалів, продавці можуть швидко і легко донести інформацію про свої товари до цільової аудиторії. Крім того, електронна комерція дозволяє збирати та аналізувати велику кількість даних про клієнтів та їхню поведінку на сайті, що дозволяє зробити більш точну та ефективну маркетингову кампанію [4].

Ще однією важливою перевагою електронної комерції є можливість забезпечення більш широкого асортименту товарів та послуг, що є доступними для покупців з будь-якої точки світу. Великі електронні маркетплейси, такі як Amazon, eBay, AliExpress тощо, дозволяють продавцям з різних країн зареєструватися на їхніх сайтах та продавати свої товари в різних куточках світу [5]. Це дозволяє покупцям мати доступ до більш широкого асортименту товарів, а продавцям збільшити свій ринок збуту.

Однак, електронна комерція також має свої недоліки та виклики. Одним з найбільших викликів є безпека і конфіденційність інформації, яку передають покупці та продавці під час здійснення транзакцій. Інформація про кредитні картки, адреси доставки та інші особисті дані можуть бути скомпрометовані та використані для крадіжок, атак хакерів та інших злочинів.

Ще одним викликом є зростання конкуренції на ринку електронної комерції. З появою нових і нових маркетплейсів та інтернет-магазинів, продавці повинні змагатися за увагу покупців та намагатися пропонувати найкращі ціни та послуги. Це може призвести до зниження маржинальності та витрат на рекламу та маркетинг.

Крім того, електронна комерція може мати вплив на традиційну роздрібну торгівлю та впливати на зайнятість. Багато традиційних крамниць та магазинів можуть бути змушені закрити свої двері, оскільки більшість покупців надає перевагу онлайн-шопінгу. Це може призвести до зменшення робочих місць у роздрібній торгівлі та зростання безробіття.

Усі ці виклики та недоліки можуть бути розв'язані за допомогою тотальної автоматизації та використанням сучасних технологій. Наприклад, захист інформації може бути покращений за допомогою шифрування та інших методів кібербезпеки. Крім того, збільшення конкуренції може спонукати продавців до розробки та впровадження нових інновацій та покращення якості товарів та послуг [6, 7].

Тотальна автоматизація може також розв'язати виклики, пов'язані зі збільшенням обсягів продажів та виробництва. Застосування інформаційних технологій може допомогти автоматизувати багато рутинних операцій, що зменшить час та витрати на їх виконання. Наприклад, програми для автоматизації відправки замовлень, обліку товарів та управління складом можуть значно полегшити роботу продавців та знизити кількість помилок.

Ще одним важливим викликом електронної комерції є недостатня довіра споживачів до інтернет-магазинів. Це пов'язано з ризиком шахрайства, порушенням конфіденційності даних та іншими проблемами. Однак, забезпеченням безпеки та довіри можна зайнятися за допомогою використання спеціальних сертифікатів та протоколів безпеки, збору відгуків та рейтингів від покупців, а також наданням гарантій та повернень товару.

Також важливим викликом є забезпечення доставки товарів у встановлений термін та з дотриманням якості. Однак, використання сучасних технологій може допомогти вирішити ці проблеми. Наприклад, за допомогою спеціальних програм для відстеження доставки товарів, покупці можуть слідкувати за їхнім рухом та бути в курсі процесу доставки. Крім того, розробка нових технологій доставки, таких як дрони та роботи-кур'єри, може значно покращити швидкість та якість доставки.

Отже, можна зробити висновок, що електронна комерція має багато переваг та може стати справжньою революцією в сучасному світі бізнесу. Вона дозволяє покупцям з усього світу зручно та ефективно здійснювати покупки, а продавцям збільшувати свій ринок збуту та заробіток. Завдяки електронній комерції бізнес-процеси можуть бути значно оптимізовані та

автоматизовані, що сприяє підвищенню продуктивності та ефективності роботи компаній.

Однак, електронна комерція також має свої недоліки та виклики, які можуть бути розв'язані за допомогою тотальної автоматизації та використанням сучасних технологій [8, 9]. Наприклад, кібербезпека, проблеми з доставкою та поверненням товарів, розвиток конкуренції та нестабільність ринку.

У цілому, електронна комерція є важливим елементом в сучасному бізнесі та продовжує активно розвиватися, пропонуючи нові та інноваційні рішення для продавців та покупців. Зростання використання смартфонів та інших мобільних пристроїв стимулює розвиток мобільної електронної комерції, яка є одним з найбільш перспективних напрямів розвитку цієї галузі.

Таким чином, відправною точкою для розвитку електронної комерції є постійний пошук інновацій та використання сучасних технологій для підвищення ефективності та якості бізнес-процесів. Це дозволяє компаніям зберігати конкурентні переваги, розширювати свій ринок збуту та забезпечувати задоволення покупців.

## 1.2 Огляд програм для ведення обліку з продажу товару

Облік продажу товарів є важливим аспектом бізнесу, оскільки він дозволяє відслідковувати кількість проданих одиниць товарів, контролювати запаси на складі та зберігати інформацію про товари. Із зростанням попиту на автоматизовані рішення, на ринку з'являється все більше програм для ведення обліку з продажу товару.

Однією з найпопулярніших програм для ведення обліку є QUICKBOOKS яку можна побачити на рисунку 1.1. Вона дозволяє відслідковувати продажі, контролювати запаси на складі та створювати звіти про продажі. Також QUICKBOOKS має можливість налаштування рівнів

доступу користувачів, що є важливим аспектом для підприємств з багатьма співробітниками. Однак, QUICKBOOKS не дозволяє зберігати інформацію про матеріали та заміри, що зробило її непідходящою для певних типів бізнесу.

DATE	TYPE	CUSTOMER	DUE DATE	BALANCE	TOTAL	STATUS	ACTION
07/01/2013	Invoice	Revell Hair Salon	07/31/2013	\$1,300.00	\$1,300.00	Overdue	Receive payment
08/09/2013	Invoice	Shelly E. King	09/08/2013	\$450.00	\$450.00	Overdue	Receive payment
08/12/2013	Estimate	Scott's Auto Repair	09/22/2013	\$0.00	\$1,300.00	Pending	Start invoice
08/17/2013	Sales Receipt	Bill's Sporting Goods		\$0.00	\$1,200.00	Paid	Print
08/20/2013	Sales Receipt	Sandy Chu		\$0.00	\$2,625.00	Paid	Print
08/23/2013	Invoice	Deborah Garrett	09/22/2013	\$450.00	\$450.00	Open	Receive payment

Рисунок 1.1 – Загальний вигляд програмного застосунку QUICKBOOKS.  
Інтерфейс

Іншою популярною програмою є Херо. Вона дозволяє відслідковувати продажі, контролювати запаси на складі та створювати звіти про продажі. Херо також має можливість створення накладних та рахунків-фактур, що є важливим для бізнесу, який потребує такої функціональності. Однак, Херо не дозволяє зберігати додаткову інформацію про товари, таку як заміри та матеріали, що може бути необхідним для певних видів бізнесу.

У разі потреби програми для ведення обліку з продажу товарів, яка буде більш спеціалізованою під одяг, вам може знадобитися більш настроюваний застосунок. Такий застосунок повинен мати можливість зберігати заміри та матеріали, а також іншу інформацію, яка потрібна для ведення обліку продажу одягу. Зазвичай, загальноприйняті програми для ведення обліку продажу товарів, такі як QUICKBOOKS та Херо, не мають таких функцій, які були би націлені специфічно на продаж одягу.

У загальному розумінні програми для ведення обліку з продажу товару допомагають бізнесам зберігати інформацію про товар, таку як назва товару, ціна, кількість одиниць, дата продажу та інші атрибути. Вони також дозволяють відслідковувати наявність товару на складі та переглядати звіти про продажі.

QUICKBOOKS та Xero це дві досить популярні програми для ведення обліку з продажу товару, які зазвичай використовуються маленькими бізнесами та фрілансерами. QUICKBOOKS є вебзастосунком, який можна використовувати в будь-якому браузері, а Xero – це програмне забезпечення, яке потрібно встановлювати на ваш комп'ютер, його інтерфейс можна побачити на рисунку 1.2.

The screenshot displays the Xero interface for a business bank account. At the top, there's a navigation bar with 'Plumbing Pipes' and 'Christian Newman'. Below it, the account details for 'Business Bank Account' are shown, including the ASB logo, statement balance of 21,116.67, and balance in Xero of 120,018.56. A 'Reconciliation Report' is available for 29 Aug 2016. The main section shows a list of transactions with columns for Date, Description, Reference, Payment Ref, Spent, Received, Balance, Bank Transaction Source, and Status. The 'Balance' column is highlighted with a red dashed box.

Date	Description	Reference	Payment Ref	Spent	Received	Balance	Bank Transaction Source	Status
15 Nov 2016	Bayside Club	Cheque 123456			100,000.00	120,018.56		Unreconciled
6 Sep 2016	Bunnings			281.67		20,018.56		Unreconciled
6 Sep 2016	Calix	Fuel for EFT456		74.14		20,300.23		Unreconciled
5 Sep 2016	Calix			50.00		20,374.37		Unreconciled
2 Sep 2016	Calix			50.00		20,424.37		Unreconciled
29 Aug 2016	Walter Whyte				1,652.00	20,474.37	Imported	Reconciled
29 Aug 2016	Calix			50.00		18,822.37		Unreconciled
29 Aug 2016	Jayne White	Refund		100.00		18,872.37		Unreconciled
26 Aug 2016	Jimbo James				275.00	18,972.37	Imported	Reconciled

Рисунок 1.2 – Загальний вигляд програмного застосунку Xero. Інтерфейс застосунку

QUICKBOOKS має приємний інтерфейс та дозволяє вести облік продажів з декількох магазинів в одному місці. Він також має зручні засоби для створення рахунків-фактур та звітів про продажі. Однак, QUICKBOOKS не має спеціальних функцій для продажу одягу, таких як зберігання замірів або матеріалів.

З іншого боку Xero має більше функцій, ніж QUICKBOOKS, і дозволяє вести облік товарів, які мають різні атрибути, такі як кольори, розміри та матеріали. Він також має зручні інструменти для створення рахунків-фактур та звітів про продажі. Однак, так само, як і QUICKBOOKS, Xero не є спеціалізованим під продаж одягу, тому він може не мати певних функцій, які були б корисні для вашого бізнесу.

Тому, якщо ви не знаходите потрібної програми для ведення обліку з продажу товарів, що задовольняє ваші потреби, ви можете розглянути можливість розробки власного застосунку. Розробка власної програми може бути більш часовозатратною та ресурсозатратною, але вона може надати вам можливість створити програму, яка відповідає саме вашим потребам.

При розробці власної програми важливо враховувати всі функції та можливості, які вам потрібні. Наприклад, ви можете додати функції для збереження замірів, матеріалів та інших характеристик, які важливі для вас. Також, вам може бути потрібно додати функції для обліку продажів, складу, рахунків-фактур та звітів.

Залежно від вашого досвіду та вмінь, ви можете розглянути можливість розробки програми самостійно або звернутися до професійних розробників програмного забезпечення. Професійні розробники можуть надати вам консультації щодо функцій, дизайну та інших аспектів розробки програми, а також забезпечити вас якісним програмним продуктом.

У будь-якому випадку, важливо визначити ваші потреби та зробити дослідження на ринку програм для ведення обліку з продажу товарів, щоб знайти програму, яка найкраще відповідає вашим потребам. І, якщо ніяка програма не відповідає вашим потребам, ви можете розглянути можливість розробки власної програми.

У якості можливої функції може виступати наступна можливість, яку треба враховувати при виборі програми для ведення обліку з продажу товарів, це можливість інтеграції з іншими системами. Наприклад, якщо ви використовуєте інтернет-магазин, то вам може знадобитися програма, яка

може інтегруватися з вашою системою електронної комерції і автоматично оновлювати запаси товарів на складі.

Іншою важливою функцією є можливість відслідковувати рух товарів по ланцюжку постачання. Якщо ви купуєте товари від кількох постачальників і продаватимете їх в різних магазинах, то вам може знадобитися програма, яка дозволить відстежувати, які товари були закуплені від якого постачальника, і які з них були продані в якому магазині.

У випадку коли неможливо знайти потрібний функціонал серед стандартизованих під загальні потреби застосунків, який відповідає вашим потребам, є можливість розглянути можливість розробки власної програми для ведення обліку з продажу товарів. Це може бути досить складним та коштовним процесом, тому варто врахувати всі витрати та можливості, перш ніж розпочинати розробку.

Загалом, облік продажу товарів є важливою частиною будь-якого бізнесу, і програми для ведення обліку з продажу товарів можуть допомогти зробити процес більш ефективним та зручним. Важливо визначити ваші потреби та знайти програму, яка найкраще відповідає вашим потребам, або розглянути можливість розробки власної програми, якщо не знайдете на ринку підходящої програми.

### 1.3 Вимоги до складського застосунку

Програмний застосунок для ведення обліку з продажу одиниць товару має задовільняти різноманітні потреби для забезпечення ефективної роботи компанії.

Перш за все, застосунок повинен бути негроміздким і не завантажувати систему зайвими операціями. Оскільки продаж одягу передбачає значну кількість операцій з каталогом товарів, пошуком і додаванням нових

елементів, імпортом даних та іншими подібними діями, програма повинна бути оптимізована та швидко працювати.

Далі, програмний застосунок має бути гарно організованим і логічним для користувачів. Для цього важливо розробити зручний інтерфейс з можливістю швидкого доступу до необхідних функцій. Під час проєктування програми також варто враховувати потреби різних користувачів, щоб зробити її максимально доступною для всіх працівників компанії.

Крім того, застосунок повинен бути добре спланованим. Наприклад, важливо передбачити можливість відстеження різних параметрів товарів, таких як кольори, розміри, склад, ціни та інші. Кожен з цих параметрів повинен бути добре структурованим і зручним для користувача.

Нарешті, програмний застосунок повинен бути розроблений з уклоном під продаж одягу. Одним з головних вимог до такого застосунку є можливість ведення детальної інформації про кожен одиницю одягу, включаючи його опис, розмір, колір, матеріал, ціну та інші параметри. Також корисним може бути функція відстеження історії замовлень, щоб зручно контролювати рух товару від постачальника до клієнта.

Окрім цього, програма повинна мати можливість створення звітів про продажі, прибуток, запаси товарів та інші показники. Це дозволить компанії контролювати фінансову сторону бізнесу і приймати рішення на основі аналізу даних.

Не менш важливим є забезпечення безпеки даних користувачів. Програмний застосунок повинен мати надійний механізм захисту конфіденційної інформації та бути здатним зберігати резервні копії даних для запобігання їх втраті.

Застосунок для ведення обліку з продажу одягу повинен також мати можливість вести облік товарів за сезонами. Одяг може бути розроблений для різних сезонів року, тому важливо мати можливість легко знайти та переглянути, який товар підходить для кожного сезону. З цією функцією

можна легко спланувати закупівлю товарів з урахуванням сезонності та попиту.

Ще одним важливим аспектом є можливість вести облік за типами товарів, такими як одяг, взуття та аксесуари. Це дозволить спростити процес замовлення та доставки товарів клієнтам та відстеження наявності товарів на складі. Також важливо мати можливість швидко та легко переглянути, які типи товарів продаються найбільше та які можуть бути збільшені в продажах.

Крім того, програма повинна забезпечувати точність та швидкість роботи. Це дуже важливо для успішного ведення бізнесу, оскільки помилки в обліку можуть призвести до втрати прибутку та клієнтів. Застосунок повинен бути добре організованим та забезпечувати легкий доступ до необхідної інформації, такої як інформація про клієнтів, замовлення та наявність товарів на складі.

Узагальнюючи, для ефективного ведення обліку з продажу одиниць товару програмний застосунок повинен бути негроміздким, гарно організованим, добре спланованим, з уклоном під продаж одягу та мати можливість створення звітів та захисту даних. Це дозволить компанії ефективно управляти своїм бізнесом та досягати більшого успіху на ринку.

#### 1.4 Постановка задачі

Актуальність проблеми полягає в тому, що у сучасних умовах на ринку продажу товарів дедалі більше уваги приділяється електронній комерції. Відповідно, зростає потреба у зручних та функціональних програмах для ведення обліку продажів товарів. Зокрема, для компаній, що займаються продажем одягу, важливо мати програму, яка зможе відстежувати детальну інформацію про кожен одиницю товару та забезпечувати швидкий та точний облік продажів.

Об'єкт роботи – це вебзастосунок для ведення обліку одиниць товару для продажу, які будуть мати електронне представлення для операцій над ними.

Метою роботи є розробка програмного застосунку для ефективного процесу ведення обліку з продажу товару, зокрема одягу, збереження даних про заміри, матеріали, кольору та інших параметрів.

Для досягнення мети необхідно вирішити такі завдання:

- розробити зручний та простий інтерфейс користувача, який буде забезпечувати швидкий доступ до функцій оперування даними одиниць товару та статистики;

- розробити алгоритм завантаження та збереження даних товарів із бази даних, яка буде забезпечувати користувачам швидкий та точний пошук необхідних даних;

- розробити систему оперування даними купівель товарів, яка дозволить відслідковувати статистику продажів;

- оптимізувати роботу платформи та забезпечити її стабільність та безперебійність роботи.

Щоб досягти поставлених цілей, необхідно проаналізувати потреби та бажання користувачів платформи, визначити найважливіші завдання та обрати технології та інструменти для їх розв'язання. Також потрібно враховувати сучасні тенденції та напрямки розвитку вебтехнологій, щоб забезпечити актуальність та конкурентоспроможність платформи.

## 2 ПРОГРАМНЕ ОБГРУНТУВАННЯ ОБРАНИХ ТЕХНОЛОГІЙ

### 2.1 Перша модель програмного рішення.

У рамках даної кваліфікаційної роботи метою є розробка програмного рішення для оптимізації роботи з обліком товарів і їх продажу у сфері продажу одягу. Для досягнення цієї мети, планується створити віконний програмний застосунок на Java з інтерфейсом користувача та базою даних [10, 11].

Віконні програмні застосунки на Java є дуже популярними у сфері розробки програмного забезпечення. Для створення таких застосунків можна використовувати Java Swing або Java FX [12]. Хоча Java FX була запущена у 2008 році як заміна Java Swing, проте зараз вона вважається застарілою технологією. Java Swing є стабільною, зручною та має більш широкий спектр функціональних можливостей.

Для зберігання та організації даних в програмному рішенні, планується використовувати базу даних PostgreSQL. PostgreSQL є однією з найпопулярніших та найпотужніших відкритих баз даних, що має високу надійність та можливості для масштабування. Вона підтримує SQL-стандарт, а також має додаткові можливості, такі як JSON-підтримка та робота з геоданими.

Використання бази даних PostgreSQL з Java Swing дозволяє забезпечити ефективну та швидку роботу програмного застосунку, який може працювати зі значною кількістю даних [13–15]. База даних PostgreSQL має високу швидкість та здатність до роботи з великими об'ємами даних, що є важливим у сфері продажу одягу, де облік і управління даними про товари є важливим.

Недоліком використання Java Swing може бути те, що це технологія, яка не є найсучаснішою та не має найкращої графічної якості порівняно з іншими сучасними технологіями, такими як JavaFX або Java Web Start. Однак, з іншого боку, це компенсується його високою стабільністю та надійністю. Крім того, Java Swing має велику кількість бібліотек та інструментів, які дозволяють

розробникам ефективно створювати інтерфейс користувача та забезпечувати високу продуктивність програмного забезпечення.

Таким чином, використання стеку технологій Java Swing та PostgreSQL у розробці програмного забезпечення для обліку та продажу товарів в сфері продажу одягу є досить оптимальним та ефективним підходом. За допомогою цих технологій можна створити програмне рішення з високою швидкістю та продуктивністю, яке забезпечить надійний та ефективний облік товарів та продажів. Водночас, слід враховувати певні недоліки використання цих технологій, але вони можуть бути компенсовані їх перевагами та можливостями, які вони надають.

### 2.1.1 Модель програми на базі технології Java Swing

Метою роботи є розробка програмного рішення для оптимізації роботи з обліком товарів і їх продажу у сфері продажу одягу. Для вирішення цього завдання підходить розробка віконного програмного застосунку на Java з інтерфейсом користувача та базою даних [16].

Java є однією з найпопулярніших мов програмування, що використовується для розробки різноманітних програмних застосунків [17]. Віконні програмні застосунки на Java можна писати з використанням Java Swing або Java FX. Детальний огляд відмінностей, переваг та недоліків даних технологій наведено у таблиці 2.1.

Java Swing – це бібліотека класів для створення віконних програмних застосунків на мові Java. Java Swing був розроблений з метою заміни старішої бібліотеки класів AWT (Abstract Window Toolkit), що не мала достатньої кількості функцій та була повільною. Java Swing був розроблений компанією Sun Microsystems (згодом поглинутою компанією Oracle Corporation) [18].

Таблиця 2.1 – Характеристики та параметри бібліотек

<b>Основа порівняння між Java Swing і Java FX</b>	<b>Java Swing</b>	<b>Java FX</b>
Компоненти	Swing має низку компонентів.	Менше компонентів порівняно зі застарілими API
Інтерфейс користувача	Стандартні компоненти інтерфейсу можна розробити за допомогою Swing	Розширені компоненти графічного інтерфейсу можна створювати з вдосконаленим зовнішнім виглядом
Розвиток	API Swing використовуються для написання компонентів інтерфейсу користувача	Сценарії JavaFX і швидка розробка інтерфейсу користувача за допомогою конструктора екрану
Функціональність	Немає нових функціональних можливостей у майбутньому	У JavaFX є багатий новий інструментарій, який, як очікується, буде розширюватися в майбутньому
Категорія	Повнофункціональна застаріла бібліотека інтерфейсу користувача	Підходить до багатофункціональних компонентів інтерфейсу користувача
Підтримка MVC	Підтримці MVC у всіх компонентах бракує узгодженості	Дружній з шаблоном MVC

Java Swing має внутрішню організацію, що базується на використанні різних класів та інтерфейсів. Кожен клас Swing відповідає за певну функцію або компонент вікна, наприклад, кнопка, текстове поле, меню тощо. Інтерфейси Swing визначають, як компоненти взаємодіють з користувачем, наприклад, реагують на кліки миші або натискання клавіш на клавіатурі.

Програмно дана технологія написана з використанням мови Java та заснований на парадигмі ООП [19]. Кожен компонент вікна є об'єктом, що наслідується від класу JComponent, який забезпечує базові методи та властивості, які необхідні для роботи з вікнами.

Java Swing є надійним та ефективним інструментом для розробки віконних програмних застосунків на мові програмування Java [20]. Вона була введена в 1997 році як основний інструмент для створення віконних застосунків у Java, і до цього дня залишається однією з найпопулярніших технологій для розробки десктопних програм [21].

Основною перевагою Java Swing є її здатність працювати на будь-яких операційних системах, тому що вона не залежить від графічного інтерфейсу ОС. Це означає, що програмний код написаний на Java Swing буде виглядати однаково на будь-якій операційній системі. Крім того, Java Swing має широкі можливості для налаштування вікон та інших елементів інтерфейсу, що дозволяє створювати досить складні інтерфейси з високим рівнем функціональності.

Дана бібліотека забезпечує високий рівень контролю над розміщенням елементів інтерфейсу та їх зовнішнім виглядом [22, 23]. У розробці інтерфейсу Swing використовуються такі елементи, як панель, кнопка, текстове поле, списки, таблиці та інші, які можна комбінувати та розміщувати за різними правилами. Крім того, Java Swing дозволяє здійснювати анімацію елементів інтерфейсу, що підвищує їх привабливість та інформативність.

Java Swing реалізована як набір бібліотек та класів, які підтримують відображення графічних елементів інтерфейсу. Ці бібліотеки реалізують різноманітні функції, такі як малювання елементів інтерфейсу, обробка подій, управління вікнами та інші.

Сервіс забезпечує широкі можливості для створення віконних програмних застосунків на Java. Основою для розробки віконних інтерфейсів є компоненти Swing, які включають в себе кнопки, поля вводу, таблиці, списки, діалогові вікна і багато іншого. Використання цих компонентів дозволяє швидко і просто створювати інтерактивний інтерфейс користувача.

Java Swing має багато переваг у порівнянні з іншими технологіями для розробки віконних програмних застосунків. Однією з переваг є його кросплатформовість, що дозволяє запускати програмні застосунки на будь-

якій операційній системі [24]. Крім того, Java Swing має велику кількість вбудованих компонентів і бібліотек, що спрощує розробку програмних застосунків. Java Swing також має зручний механізм подій, який дозволяє програмістам легко відслідковувати та реагувати на дії користувача. Зазначено блок-схему роботи програми на рисунку 2.1.

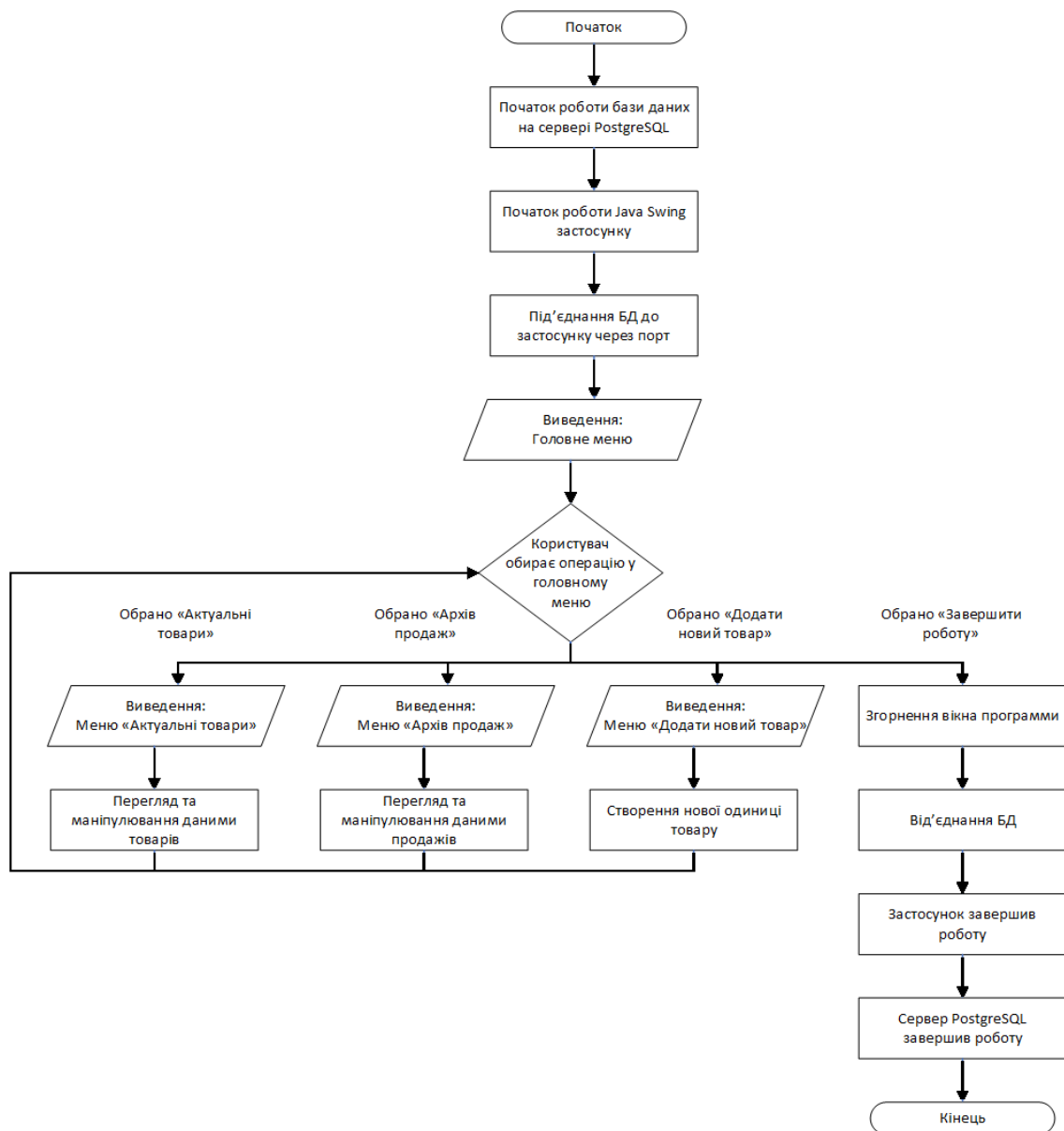


Рисунок 2.1 – Блок-схема роботи застосунку на основі Java Swing, PostgreSQL

У порівнянні з Java FX, Java Swing є більш стабільною технологією і має широкую підтримку серед інженерів, розробників та спільноти. Java FX, з

іншого боку, має новітні функції та можливості, але вона залишається менш популярною серед розробників. Також, для Java FX вимагається встановлення Java Runtime Environment 8 або вище, тоді як для Java Swing можна використовувати більш старі версії Java Runtime Environment.

Java Swing може бути програмно написаний за допомогою стандартних інструментів розробки, таких як NetBeans, Eclipse та IntelliJ IDEA, які забезпечують зручний інтерфейс для розробки та візуального редагування компонентів Swing. Крім того, є багато документації, підручників та ресурсів, що допомагають розробникам у вивченні та використанні Java Swing.

Написання програми на Java Swing вимагає розуміння таких концепцій, як контейнери, компоненти, менеджери розміщення та події. Контейнери – це об’єкти, які містять інші компоненти, такі як JFrame або JPanel. Компоненти – це елементи, які розміщуються в контейнерах, наприклад, JButton або JTable. Менеджери розміщення визначають, як компоненти розміщуються в контейнері, і включають такі типи, як BorderLayout та GridLayout. Події – це спосіб взаємодії з користувачем та програмою, такі як натискання кнопки або рух миші [25, 26].

Один з основних переваг Java Swing полягає у тому, що він є частиною стандартної бібліотеки Java. Це означає, що розробники можуть використовувати його без необхідності завантаження та встановлення окремого фреймворку. Крім того, Java Swing є портативним, тобто код, написаний на ньому, може запускатися на будь-якій платформі, що підтримує Java.

У загальному, Java Swing є надійним та потужним інструментом для розробки віконних програмних застосунків на Java. Він забезпечує зручний інтерфейс користувача та ефективну роботу з базою даних, що робить його чудовим вибором для написання програм для обліку та продажу товарів, зокрема в сфері продажу одягу. На відміну від старішої технології Java FX, Java Swing є потужним інструментом, що досі активно розвивається та підтримується [27].

Однак, Java Swing має свої недоліки. Наприклад, він не такий сучасний, як деякі інші інструменти розробки графічного інтерфейсу, тому може виглядати менш привабливо для розробників, які шукають новіші та більш інноваційні інструменти. Крім того, іноді можуть виникати проблеми з розміщенням компонентів на формі, що може бути незручно для користувачів.

Насправді Java Swing є дуже гнучким та масштабованим інструментом, що дозволяє розробляти досить складні програми з великою кількістю елементів інтерфейсу та обробкою великої кількості даних. Крім того, з підтримкою багатьох інструментів розробки, використання Java Swing може бути дуже ефективним та продуктивним для розробки програмних застосунків на Java [28, 29].

У цілому, вибір технології для розробки програмного забезпечення залежить від конкретних потреб і вимог проєкту, але Java Swing може бути відмінним вибором для написання віконних програмних застосунків, які працюють з великою кількістю даних, зокрема в сфері продажу одягу.

### 2.1.2 Модель бази даних на сервері PostgreSQL

Віконний застосунок на Java Swing та PostgreSQL базується на структурі бази даних, яка створена, організована та працює з деякими основними таблицями: Category, Sizes, Condition та Goods.

Таблиця «Category» використовується для зберігання категорій товарів і містить поле «CategoryName» типу TEXT в якості первинного ключа. В цій таблиці містяться дані про категорії товарів, наприклад «Верхній одяг», «Плаття», «Спідниці» тощо.

Таблиця «Sizes» зберігає інформацію про розміри товарів і має поле «SizeName» типу TEXT як первинний ключ. У цій таблиці містяться записи, які відповідають різним розмірам товарів, таким як «XXS», «XS», «S», «M», «L» тощо [30, 31].

Таблиця «Condition» використовується для зберігання інформації про стан товарів і містить поле «ConditionName» типу TEXT як первинний ключ. У цій таблиці містяться записи, які представляють різні стани товарів, наприклад «Новий», «Ідеальний», «Б/у», «З дефектом» тощо.

Таблиця «Goods» є головною таблицею, яка містить дані про товари. Вона включає такі поля:

- «goodID» унікальний ідентифікатор товару, який автоматично генерується (SERIAL PRIMARY KEY);
- «Price» ціна товару типу DECIMAL(10, 2), яка зберігає ціну товару з точністю до двох десяткових знаків;
- «PrimeCost» собівартість товару типу DECIMAL(10, 2), яка вказує на вартість виробництва товару з точністю до двох десяткових знаків;
- «Income» прибуток від продажу товару типу DECIMAL(10, 2), який розраховується відносно різниці між ціною і собівартістю товару з точністю до двох десяткових знаків;
- «Brand» назва бренду товару типу TEXT;
- «Category» поле з таблиці «Category», яке посилається на поле «CategoryName» таблиці «Category» і вказує на категорію товару;
- «Size» поле з таблиці «Sizes», яке посилається на поле «SizeName» таблиці «Sizes» і вказує на розмір товару;
- «Condition» поле з таблиці «Condition», яке посилається на поле «ConditionName» таблиці «Condition» і вказує на стан товару.

Така структура бази даних дозволяє зберігати і організовувати дані про товари, їх категорії, розміри та стан. Застосування бази даних PostgreSQL забезпечує надійну та ефективну роботу з даними. Запити можуть бути виконані швидко та точно завдяки оптимізованому рішенню для зберігання та операцій з великим обсягом даних.

Розроблений віконний застосунок на Java Swing забезпечує інтерфейс користувача для зручного взаємодії з базою даних. Користувач може переглядати, додавати, редагувати та видаляти дані про товари, категорії,

розміри та стан. Застосунок може забезпечити різноманітні функції, такі як пошук товарів за категорією, фільтрація за розміром, сортування за ціною тощо.

Використання комбінації Java Swing та PostgreSQL відкриває широкі можливості для створення потужного та надійного віконного застосунку, спрямованого на оптимізацію роботи з обліком товарів та їх продажем у сфері продажу одягу [32, 33]. Цей застосунок дозволяє виконувати різноманітні завдання, пов'язані з управлінням і моніторингом інвентаризації, збереженням даних про товари та клієнтів, а також ефективним веденням процесу продажу та створенням звітів і аналітики. Загальну схему бази даних можна побачити на рисунку 2.2.

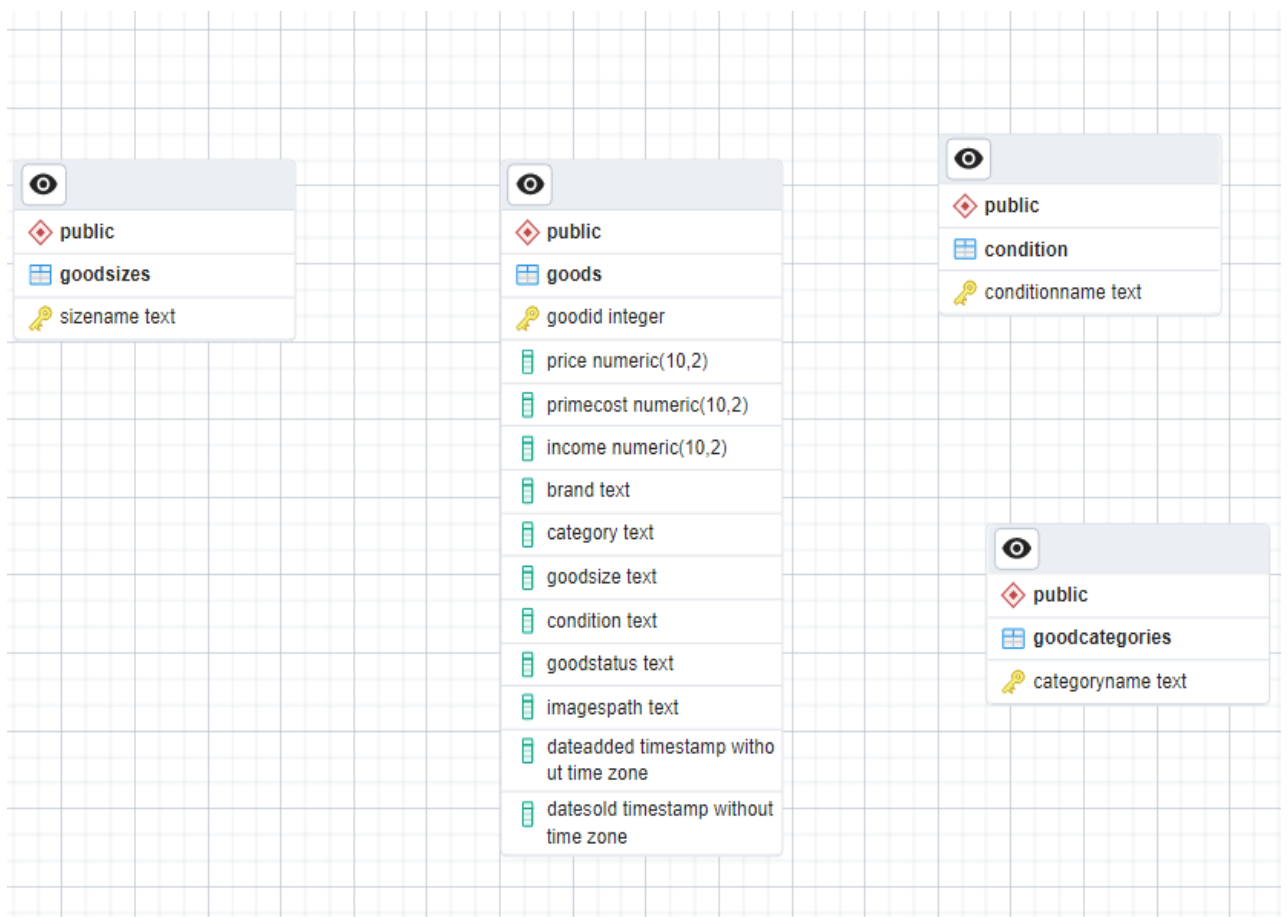


Рисунок 2.2 – Схема бази даних у сервісі PostgreSQL

## 2.2 Друга модель програмного рішення

### 2.2.1 Модель програми на базі технологій Web та Java як back-end

У цьому розділі кваліфікаційної роботи розглядається розробка вебзастосунку, в якому будуть використані HTML, CSS та Java для back-end розробки. Головною метою застосунку є надання можливості перегляду та маніпулювання даними про одиниці товару, зокрема одягу.

Застосунок буде включати HTML-сторінки, які будуть відповідати за дизайн та інтерфейс користувача. CSS використовуватиметься для стилізації сторінок та надання їм привабливого вигляду. Java виступатиме в ролі back-end, виконуючи обробку запитів користувача та взаємодію з базою даних.

Основні функції, які будуть реалізовані у застосунку, включатимуть:

Перегляд полів таблиці, що зберігає дані про одяг, такі як розмір, ціна, собівартість, дохід, стан, наявність та бренд. Користувач зможе переглядати ці дані у зручному форматі [34, 35].

Зміна даних у вже збережених товарах. Користувач матиме можливість редагувати поля та оновлювати інформацію про товари у базі даних.

Видалення або помітка товарів як проданих. Користувач зможе видаляти існуючі товари або встановлювати їх статус як «проданий», внаслідок чого вони будуть переміщені до архіву.

Створення нових записів про одиниці товару. Користувач зможе вводити параметри нових товарів та додавати їх до бази даних.

Вебзастосунок буде мати базову навігацію через головне меню та його пункти, які дозволять користувачеві переміщатися між різними операціями: перегляд, редагування, видалення та створення записів про одиниці товару.

При запуску вебзастосунку, користувач буде бачити перед собою головне меню, яке буде містити пункти навігації до різних операцій. Наприклад, в меню можуть бути такі пункти:

«Перегляд товарів»: При виборі цього пункту, користувач буде перенаправлений на сторінку, де будуть відображені всі наявні товари.

Користувач зможе переглядати інформацію про кожен товар, зокрема його розмір, ціну, собівартість, дохід, стан, наявність та бренд.

«Редагування товарів»: Цей пункт дозволить користувачу редагувати дані про вже збережені товари. При виборі цього пункту, користувачу буде показана сторінка зі списком товарів, і він зможе вибрати товар, який потрібно відредагувати. Користувач зможе змінювати будь-яке поле товару та зберігати зміни.

«Видалення товарів»: Цей пункт надасть можливість користувачу видаляти товари з бази даних. Користувачу буде показана сторінка зі списком товарів, і він зможе вибрати товари, які потрібно видалити. Після підтвердження видалення, товари будуть видалені з бази даних.

«Створення нового товару»: Цей пункт дозволить користувачу створювати нові записи про одиниці товару. При виборі цього пункту, користувачу буде відображена сторінка з формою, де він зможе ввести всі необхідні параметри нового товару, такі як розмір, ціна, собівартість, дохід, стан, наявність та бренд. Після заповнення форми, новий товар буде доданий до бази даних.

Вебзастосунок буде використовувати Java в якості back-end розробки для обробки запитів користувача та взаємодії з базою даних. Для досягнення цієї мети, можна використати Java-фреймворки, такі як Spring або JavaServer Faces (JSF), що дозволять створити потужний та ефективний back-end.

У якості back-end розробки для обробки запитів користувача та взаємодії з базою даних. Для досягнення цієї мети, можна використати Java-фреймворки, такі як Spring або JavaServer Faces (JSF), що дозволять створити потужний та ефективний back-end.

При обробці запитів користувача, вебзастосунок буде використовувати Java для взаємодії з базою даних. Запити, отримані від користувача через вебінтерфейс (HTML/CSS), будуть передаватися до back-end частини застосунку, де вони будуть оброблятися та виконуватися необхідні операції з базою даних.

Для забезпечення взаємодії з базою даних, можна використати Java Database Connectivity (JDBC) або вищерівняні інструменти, такі як Hibernate або MyBatis. Ці інструменти дозволяють легко взаємодіяти з базою даних, виконувати запити і отримувати результати для подальшого відображення на вебзастосунку. Зазначено блок-схему роботи програми на рисунку 2.3.

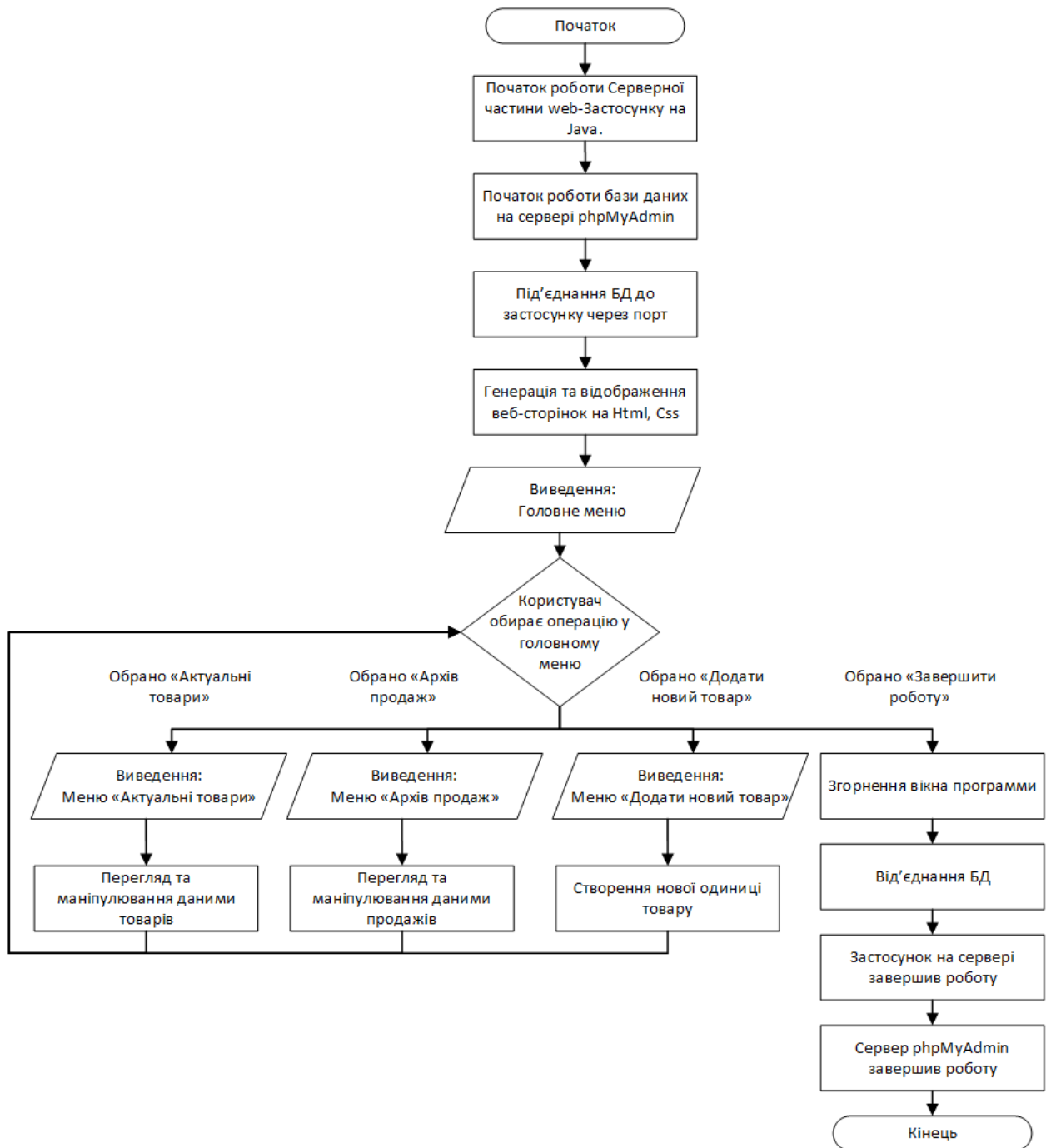


Рисунок 2.3 – Блок-схема роботи застосунку на основі Java backend, HTML, Css, та phpMyAdmin у якості БД

Таким чином, вебзастосунок буде складатися з трьох основних компонентів: HTML/CSS для вебінтерфейсу та відображення даних, Java для back-end розробки та обробки запитів користувача, а також база даних для зберігання і управління даними про одиниці товару.

Описова модель програми:

- користувач запускає вебзастосунок у своєму веббраузері;
- відображається головне меню з пунктами навігації;
- користувач обирає певний пункт меню, наприклад, «Перегляд товарів»;
- вебзастосунок виконує запит до back-end частини, реалізованої на Java;
- Java-код обробляє запит, отримує необхідні дані з бази даних і передає їх до вебінтерфейсу.

Вебінтерфейс використовує HTML та CSS для відображення сторінок та стилізації елементів. Користувач може переглядати дані про одиниці товару, такі як розмір, ціна, собівартість, дохід, стан, наявність та бренд. Ці дані витягуються з бази даних за допомогою Java-коду, який виконує запити до бази даних за допомогою JDBC [36].

Крім перегляду даних, користувач також може внести зміни в існуючі товари. Зміни вносяться через вебінтерфейс, і дані передаються до back-end частини застосунку для обробки та оновлення в базі даних. Користувач також може видаляти товари або помічати їх як продані, після чого вони переміщуються до архіву.

Надається можливість створювати нові записи про товари та вказувати їх параметри. Користувач може ввести необхідну інформацію через вебінтерфейс, і дані передаються до back-end частини для збереження в базі даних.

Застосунок буде мати гнучку структуру, що дозволяє додавати нові функціональні можливості та покращувати існуючі. Навігація в програмі буде реалізована через головне меню, де кожен пункт меню відповідає певній

операції або функціоналу. Користувач може вибрати пункт меню та переміщатися між різними операціями.

Всі ці компоненти – HTML, CSS, Java та база даних – працюватимуть разом для забезпечення зручного та ефективного вебзастосунок для перегляду, маніпулювання та управління даними одиниць товару.

### 2.2.1 Модель бази даних на сервері phpMyAdmin

У даному розділі описано процес розробки вебзастосунок з використанням HTML, CSS, Java та phpMyAdmin. Вебзастосунок надає можливість перегляду та маніпулювання даними про одиниці товару (одяг), такими як розмір, ціна, собівартість, дохід, стан, наявність та бренд. Для зберігання цих даних використовується база даних phpMyAdmin, яка дозволяє організувати структуру таблиць та виконувати операції з ними.

База даних у phpMyAdmin складається з чотирьох таблиць: Category (категорії), Sizes (розміри), Condition (стан) та Goods (товари). Кожна таблиця має свою структуру та поля, що відповідають відповідним сутностям.

Таблиця Category містить інформацію про категорії одягу. Кожен запис у цій таблиці має поле CategoryName як первинний ключ, що визначає назву категорії.

Таблиця Sizes зберігає розміри одягу. Кожен запис у цій таблиці має поле SizeName як первинний ключ, що визначає назву розміру.

Таблиця Condition містить інформацію про стан товарів. Кожен запис у цій таблиці має поле ConditionName як первинний ключ, що визначає назву стану.

Таблиця Goods зберігає деталі про одиниці товару. Кожен запис у цій таблиці має наступні поля: goodID (унікальний ідентифікатор товару), Price (ціна), PrimeCost (собівартість), Income (дохід), Brand (бренд), Category (посилання на категорію у таблиці Category), Size (посилання на розмір у

таблиці Sizes) та Condition (посилання на стан у таблиці Condition). Поле goodID є первинним ключем таблиці.

Організація бази даних в phpMyAdmin включає створення таблиць з необхідною структурою та правилами роботи з даними. Для цього використовуються мова SQL та функціональні можливості phpMyAdmin.

У розглянутому прикладі бази даних для вебзастосунку з одягом, спочатку створюються таблиці з відповідними полями та ключами. Таблиця Category має поле CategoryName як первинний ключ, що визначає назву категорії. Таблиця Sizes має поле SizeName як первинний ключ, визначаючи розмір. Таблиця Condition містить поле ConditionName як первинний ключ, визначаючи стан товару. Таблиця Goods містить поля для деталей про товар, такі як ціна, собівартість, дохід, бренд, а також посилання на категорію, розмір та стан, використовуючи зовнішні ключі для зв'язку з таблицями Category, Sizes та Condition відповідно.

Для додавання початкових даних до таблиць використовуються команди INSERT INTO. Наприклад, у таблицю Category можна додати записи про різні категорії одягу за допомогою INSERT INTO Category (CategoryName) VALUES ('Верхній одяг'), ('Плаття'), ('Спідниці') і так далі. Аналогічно, можна додати початкові дані до таблиць Sizes та Condition. Лістинг 2.1 містить SQL скрипти розгорнення Бази даних.

Лістинг 2.1 Створення таблиць застосунку:

```
CREATE TABLE Category (CategoryName TEXT PRIMARY KEY);  
INSERT INTO Category (CategoryName)  
VALUES ('Верхній одяг'), ('Плаття'), ('Спідниці'), ('Майки й  
футболки'), ('Сорочки та блузки'), ('Кофти'), ('Нижня білизна'),  
('Акcesуари'), ('Інші речі'), ('Спортивний одяг'), ('Костюми'),  
('Комбінезони'), ('Одяг для дому та сну'), ('Спецодяг'), ('Для вагітних'),  
('Взуття'), ('Штани та шорти');  
  
CREATE TABLE Sizes (SizeName TEXT PRIMARY KEY);
```

```

INSERT INTO Sizes (SizeName)
VALUES ('XXS'), ('XS'), ('S'), ('M'), ('L'), ('XL'), ('XXL'), ('3XL'), ('4XL'),
('5XL'), ('6XL'), ('7XL'), ('One size'), ('Інший');
CREATE TABLE Condition (ConditionName TEXT PRIMARY KEY);
INSERT INTO Condition (ConditionName)
VALUES ('Новий'), ('Ідеальний'), ('Б/у'), ('З дефектом'), ('З браком');
CREATE TABLE Goods (
    goodID SERIAL PRIMARY KEY,
    Price DECIMAL(10, 2),
    PrimeCost DECIMAL(10, 2),
    Income DECIMAL(10, 2),
    Brand TEXT,
    Category TEXT REFERENCES Category(CategoryName),
    Size TEXT REFERENCES Sizes(SizeName),
    Condition TEXT REFERENCES Condition(ConditionName)
);

```

Правила роботи з базою даних визначаються структурою таблиць та їх взаємозв'язками. Наприклад, у таблиці Goods поле Category є зовнішнім ключем, що посиляється на поле CategoryName у таблиці Category. Це дозволяє зв'язати дані про товари з відповідною категорією. Аналогічно, поля Size та Condition мають зовнішні ключі, що посиляються на відповідні таблиці.

Крім цього, ви можете виконувати складніші запити, такі як об'єднання таблиць, фільтрація даних за певними умовами, сортування тощо, використовуючи різні оператори та функції SQL.

Загалом, phpMyAdmin надає зручне інтерфейсне середовище для виконання різних операцій з базою даних, дозволяючи вам ефективно керувати та маніпулювати вашими даними.

### 2.3 Порівняння технологій та обґрунтування вибору моделі

У даному розділі описано розробку вебзастосунку для перегляду та маніпулювання даними одиниць товару з використанням технологій HTML, CSS, Java та phpMyAdmin. Застосунок надає зручні функції, що дозволяють користувачеві взаємодіяти з базою даних, переглядати та редагувати різні поля товарів.

Можливості зміни даних: Застосунок дозволяє користувачу змінювати дані вже збережених товарів. Наприклад, можна змінювати ціну, розмір, стан тощо. Крім того, користувач може видаляти існуючі товари або помічати їх як продані, після чого вони будуть переміщені до архіву.

Для реалізації цього застосунку були розглянуті два варіанти технологій: Java Swing з використанням бази даних PostgreSQL та HTML, CSS, Java з використанням phpMyAdmin.

PhpMyAdmin може не надавати такого рівня контролю та оптимізації бази даних, який можна було б отримати з використанням спеціалізованої системи управління базами даних, наприклад, PostgreSQL.

Java Swing надає широкий вибір готових компонентів та інструментів для побудови інтерфейсу користувача, що спрощує розробку. Це дозволяє створювати багатофункціональний та привабливий інтерфейс з великою гнучкістю та можливістю налаштовувати гнучкий функціонал навідміну від HTML і CSS які обмежені в плані готових рішень. Це може обмежити можливості розробки багатофункціонального та сучасного інтерфейсу.

У заключенні, вибір між Java Swing та PostgreSQL або HTML, CSS, Java та phpMyAdmin залежить від конкретних потреб розробників. Було обрано варіант Java Swing та PostgreSQL через його здатність надати широкий функціонал для реалізації бізнес-логіки, а також можливість використання потужної реляційної бази даних для ефективного зберігання і маніпулювання даними одиниць товару.

### 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБЗАСТОСУНКУ

#### 3.1 Структура розробки вебзастосунку

Розробка вебзастосунку є комплексним на багатоскладовим процесом та містить у собі множину різних задач від вибіру середовища розробки та технологічного стеку до побудови алгоритмів відпрацювання програми. Дана процедура здійснюється за наступною схемою:

Крок 1. Ставиться задача розробити вебзастосунок. Пишеться перелік вимог та проєктується функціонал застосунку. Пишеться бізнес-логіка. Розробляється дизайн програми.

Крок 2. Вибирається технологічний стек для розробки: мови програмування, бази даних, фреймворки, бібліотеки.

Крок 3. Вибирається середовище розробки яке задовільняє вимогам написання застосунку із технологіями із Крок 2.

Крок 4. Проєктується та пишеться БД у середовищі системи управління бази даних згідно з розробленою бізнес-логікою.

Крок 5. Розробляється програмна реалізація по плану розробленого дизайну, програма під'єднується до БД, та пишеться функціонал маніпулювання даними з БД.

Крок 6. Розробляється меню програми, та навігація програмою, відповідні до дизайну, з'являється інтерфейс користувача.

Крок 7. Проводяться завершальні міри по редагування дизайну й доведення застосунку до кінцевого вигляду.

Крок 8. Вебзастосунок тестується, та виправляються недоліки, які були знайдені у процесі тестування якості.

Крок 9. Вебзастосунок передається у експлуатацію.

Перед розробкою застосунку треба визначити які вікна потрібно спроектувати. План вікон проєкту можна побачити в таблиці 3.1.

Таблиця 3.1 – Меню застосунку та їх опис

<b>Меню</b>	<b>Опис</b>	<b>Функції</b>	<b>Навігація</b>
Головне меню	Дане меню забезпечує управління операціями застосунку.	Навігація застосунком. Кнопки переходу до інших меню.	З цього меню можна дістатися до «Актуальні товари», «Архів продаж», «Створення нової одиниці товару»
Актуальні товари	Дане меню забезпечує перегляд товарів, із статусом «В наявності».	Надає змогу переміщувати товар у «Продано», видаляти, редагувати товар та повертатись у головне меню.	З цього меню можна дістатися до «Головне меню», «Редагування одиниці товару»
Архів продаж	Дане меню забезпечує перегляд товарів, із статусом «Продано».	Надає змогу переміщувати товар у «В наявності», видаляти товар, обирати дати перегляду.	З цього меню можна дістатися до «Головне меню», «Вибір дат»
Створення нової одиниці товару	Дане меню забезпечує заповнення полів товару, та посилення запиту до БД на створення нового екземпляру.	Містить поля для занесення даних товару, та можливість обрати зображення. Є кнопка повернення до головного меню.	З цього меню можна дістатися до «Головне меню»
Редагування одиниці товару	Забезпечує виправлення полів товару, та посилення запиту до БД на оновлення даних екземпляру.	Містить поля для виправлення даних товару, та можливість обрати зображення. Є кнопка повернення до головного меню.	З цього меню можна дістатися до «Актуальні товари»
Вибір дат	Забезпечує вибір дат перегляду архіву.	Містить два поля вибору дат, які зберігаються та передаються у меню «Архів продаж».	З цього меню можна дістатися до «Архів продаж»

Програма, в якості інтерфейсу користувача, зазвичай включає в себе різні вікна та систему навігації між ними. Вікна представляють собою окремі графічні компоненти, які дозволяють користувачу взаємодіяти з різними функціями та даними програми. Ці вікна можуть бути розташовані на різних екранах або з'являтися у вигляді діалогових вікон, що виходять з-під основного вікна програми.

Система навігації дозволяє користувачеві переходити від одного вікна до іншого зручним та логічним способом. Це може бути досягнуто за допомогою кнопок, посилок або інших елементів інтерфейсу, що вказують на наявність інших вікон або розділів програми. Користувач може вільно переміщатись між вікнами, залежно від своїх потреб та завдань, що виконуються. На рисунку 3.1 можна побачити ієрархію вікон, та внутрішню організацію, переміщення між вікнами.

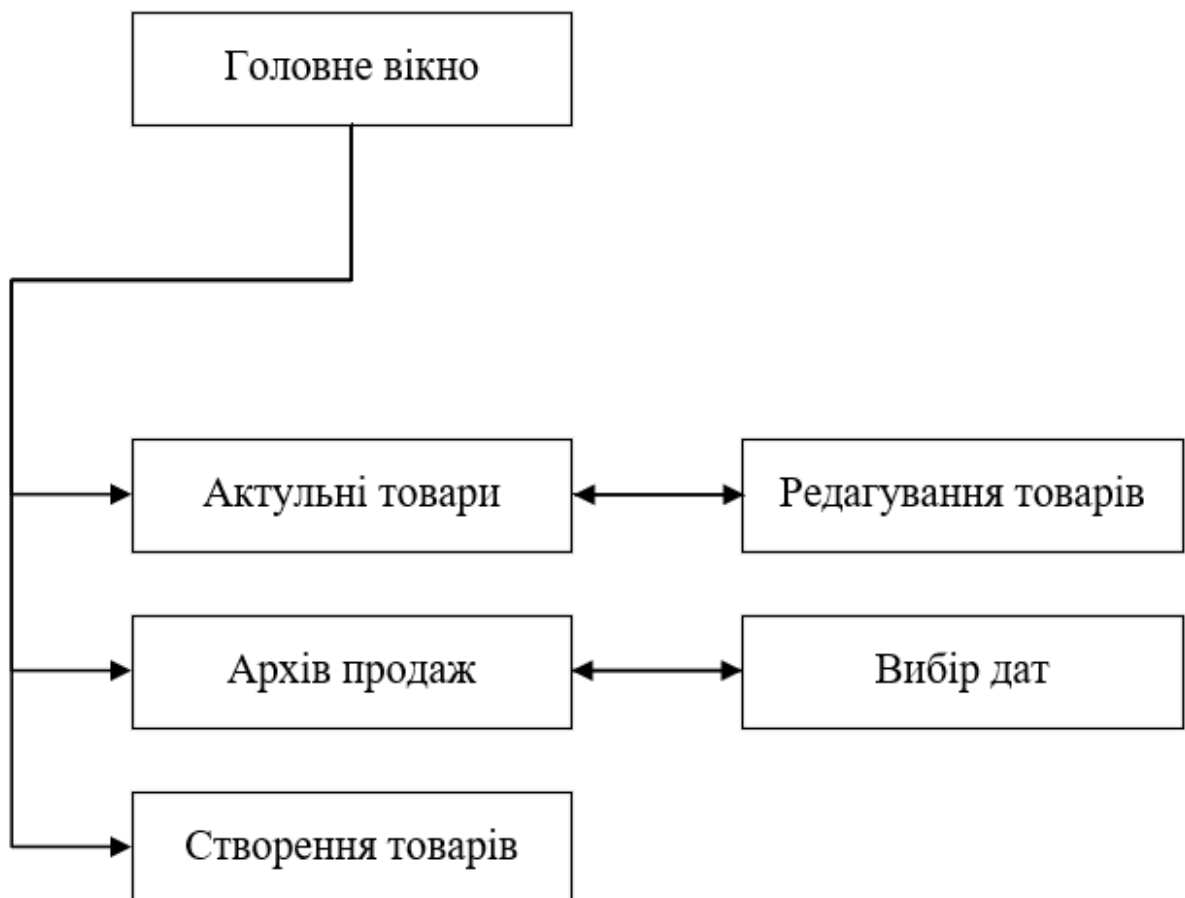


Рисунок 3.1 – Ієрархія робочих областей застосунка

Наявність декількох вікон у програмі дозволяє користувачу зручно переходити з одного функціонального блоку до іншого, зберігаючи контекст роботи і швидко поступаючи до необхідних даних чи інструментів. Крім того, це спрощує розподіл завдань та забезпечує більш ефективну організацію робочого процесу.

Загалом, наявність вікон та системи навігації в програмі робить її більш інтуїтивно зрозумілою та зручною для користувача, дозволяючи зручно переміщатись та взаємодіяти з різними функціями та даними програмного забезпечення.

В даному розділі описуються бізнес-правила, що лежать в основі застосунку для ведення обліку з продажу одиниць товару (одягу). Ці правила дозволяють забезпечити ефективну та точну систему обліку, що сприяє оптимізації управління запасами, моніторингу продажів та забезпеченню актуальності даних:

- кожен товар має унікальний ідентифікатор та повну інформацію про назву, опис, розмір, категорію та ціну;
- адміністратор магазину може додавати нові товари, редагувати або видаляти існуючі;
- забезпечується актуалізація даних про наявність товарів на складі;
- застосунок дозволяє реєструвати продажі товарів та зберігати повну інформацію про кількість проданих одиниць, дату продажу, клієнта та вартість;
- адміністратор має можливість відстежувати загальний обсяг продажів та проводити аналіз ефективності різних товарів чи категорій;
- система автоматично оновлює залишки товарів після реєстрації продажу;
- адміністратор має доступ до актуальної інформації про кількість товарів на складі;
- при досягненні певного рівня запасів, адміністратор отримує повідомлення про необхідність поповнення запасів;

- застосунок надає можливість генерувати звіти про продажі, запаси товарів, прибутковість та іншу статистику;
- адміністратор може аналізувати звіти та використовувати їх для прийняття управлінських рішень та стратегічного планування;
- застосунок забезпечує захист даних та обмежений доступ до функцій тільки для адміністратора магазину;
- здійснюється резервне копіювання даних для уникнення втрати важливої інформації;
- застосунок має можливість оновлюватись для виправлення помилок, вдосконалення функціональності та додавання нових можливостей.

Ці бізнес-правила визначають ефективну роботу застосунку для ведення обліку з продажу одиниць товару (одягу), забезпечуючи адміністратору магазину потужні інструменти для управління та контролю над процесами продажу та управління запасами.

### 3.2 Розробка Бази Даних у PostgreSQL

Розробка бази даних є важливою складовою процесу створення програмного забезпечення. У випадку використання PostgreSQL як системи управління базами даних (СУБД), розробка бази даних здійснюється з використанням його функціональності та можливостей. Основним завданням розробки бази даних є створення оптимальної структури, що відображає потреби та вимоги проєкту. Це включає визначення таблиць, полів, зв'язків між ними, а також правил і обмежень для забезпечення цілісності даних. Під час розробки бази даних у PostgreSQL слід враховувати такі аспекти, як ефективність запитів, надійність збереження даних та можливості резервного копіювання та відновлення. Розробка бази даних у PostgreSQL вимагає професійного підходу, досвіду у роботі з СУБД та знання відповідних концепцій і методів проєктування баз даних.

Під час розробки бази даних у PostgreSQL для даного проєкту були визначені наступні сутності: Category, Sizes, Condition і Goods.

Сутність Category використовується для зберігання категорій товарів. У таблиці Category є поле CategoryName, яке є основним ключем і містить назву категорії товару. Наприклад, категорії можуть бути «Верхній одяг», «Плаття», «Спідниці» та інші.

Сутність Sizes зберігає розміри товарів. У таблиці Sizes також є основний ключ SizeName, який містить розмір товару, наприклад, «XXS», «XS», «S», тощо.

Сутність Condition використовується для зберігання стану товарів. Таблиця Condition також має основний ключ ConditionName, який містить назву стану товару, наприклад, «Новий», «Ідеальний», «Б/у» тощо.

Сутність Goods представляє собою основну сутність, яка зберігає інформацію про товари. У таблиці Goods є такі поля: goodID (основний ключ, автоматично збільшуване значення), Price (ціна товару), PrimeCost (собівартість товару), Income (прибуток), Brand (бренд товару), Category (зовнішній ключ, посилається на поле CategoryName в таблиці Category), Size (зовнішній ключ, посилається на поле SizeName в таблиці Sizes), Condition (зовнішній ключ, посилається на поле ConditionName в таблиці Condition).

Ці сутності знаходяться у певних залежностях одна від одної. Наприклад, сутність Goods має зовнішні ключі, які посилаються на відповідні поля CategoryName, SizeName та ConditionName у таблицях Category, Sizes та Condition відповідно. Це дозволяє встановлювати зв'язки між товаром і його категорією, розміром та станом.

База даних була створена шляхом виконання SQL-скрипту, який включав створення таблиць Category, Sizes, Condition та Goods. Для кожної таблиці були визначені необхідні поля та їх типи даних. Після створення таблиць були виконані запити INSERT для додавання даних до таблиць Category, Sizes та Condition.

Організація бази даних полягає у використанні відповідних ключів та зв'язків між таблицями, що дозволяє забезпечити цілісність даних та ефективний доступ до інформації. Наприклад, зв'язок між сутністю Goods та сутностями Category, Sizes та Condition дозволяє здійснювати пошук товарів за категорією, розміром та станом.

Після створення та організації бази даних можна працювати з нею, виконуючи різні запити, такі як вибірка товарів за категорією, оновлення цін або додавання нових записів. За допомогою SQL-запитів можна отримувати потрібну інформацію з бази даних та здійснювати різноманітні операції з даними. Ці залежності між сутностями дозволяють встановлювати зв'язки і взаємозв'язки між даними в базі даних. Вони допомагають забезпечити цілісність даних та забезпечують ефективний доступ до інформації. Правильне визначення сутностей і їх взаємозв'язків у проєкті бази даних є ключовим кроком у розробці БД у PostgreSQL. Схему БД зображено на рисунку 3.2.

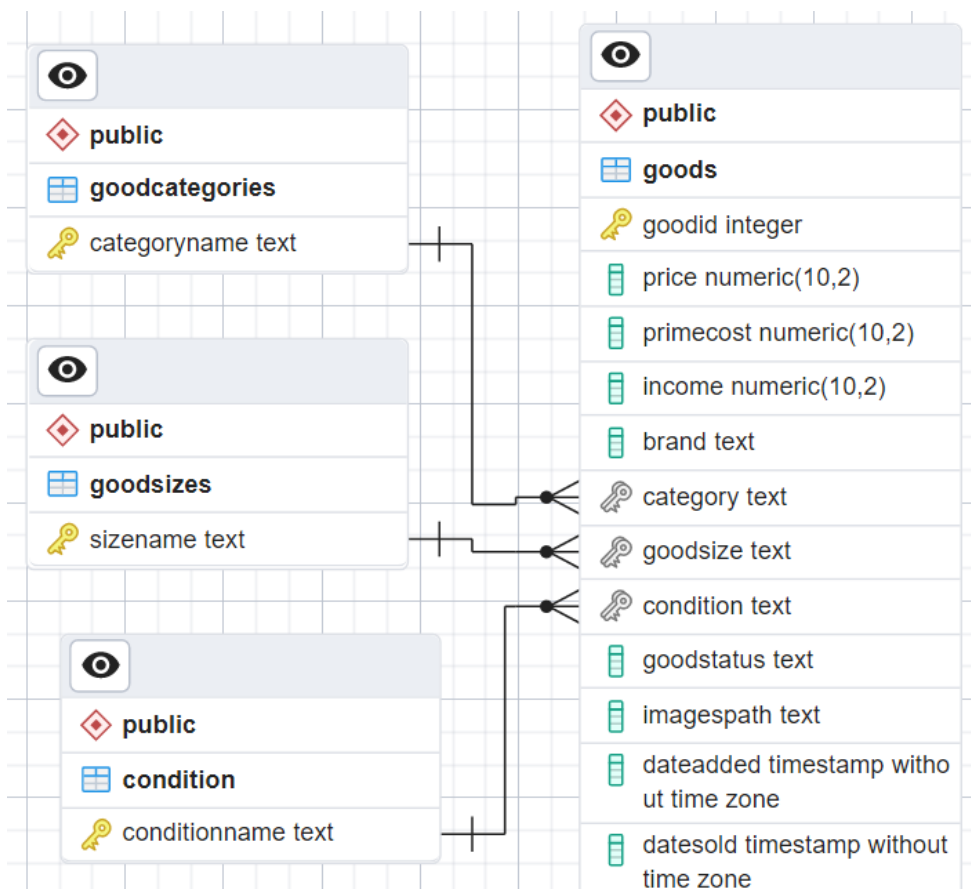


Рисунок 3.2 – Схема у СУБД PostgreSQL

Розробка бази даних у PostgreSQL включала створення кількох таблиць з відповідними полями, первинними ключами, зовнішніми ключами та зв'язками між ними.

Таблиця Category містить поле CategoryName, яке є первинним ключем і містить назву категорії товару. У цій таблиці зберігаються категорії товарів, такі як «Верхній одяг», «Плаття», «Спідниці» та інші.

Таблиця Sizes має поле SizeName, яке є первинним ключем і містить розмір товару. Наприклад, розміри можуть бути «XXS», «XS», «S», тощо.

Таблиця Condition використовується для зберігання стану товарів. У ній є поле ConditionName, яке є первинним ключем і містить назву стану товару, такі як «Новий», «Ідеальний», «Б/у» тощо.

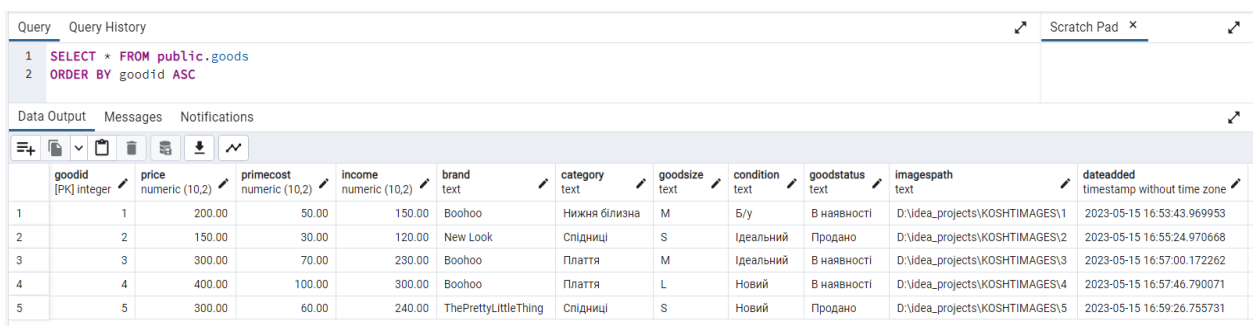
Основна таблиця Goods містить наступні поля: goodID (первинний ключ, з автоматично збільшуваним значенням), Price (ціна товару), PrimeCost (собівартість товару), Income (прибуток), Brand (бренд товару), Category (зовнішній ключ, посилається на поле CategoryName в таблиці Category), Size (зовнішній ключ, посилається на поле SizeName в таблиці Sizes), Condition (зовнішній ключ, посилається на поле ConditionName в таблиці Condition).

Таким чином, таблиця Goods має зв'язки з таблицями Category, Sizes та Condition, використовуючи зовнішні ключі. Ці зв'язки дозволяють забезпечити цілісність даних і здійснювати звернення до інших таблиць для отримання додаткової інформації. Типи даних, використані в таблицях, включають TEXT для рядків, SERIAL для автоматично збільшуваних значень та DECIMAL для числових значень з фіксованою точністю.

Таким чином, створена база даних у PostgreSQL представляє собою добре організовану та працюючу структуру, яка забезпечує ефективне зберігання та управління даними про товари. Таблиця Category зберігає інформацію про категорії товарів, що дозволяє згрупувати їх за спільними характеристиками. Таблиця Sizes містить розміри товарів, що дозволяє точно вказати доступні варіанти розмірів для кожного товару. Таблиця Condition містить дані про стан товарів, що дозволяє класифікувати їх за їхньою

фізичною умовою. Нарешті, таблиця Goods містить детальну інформацію про самі товари, включаючи ціну, собівартість, прибуток, бренд, а також посилання на категорію, розмір і стан товару.

Загалом, розроблена база даних у PostgreSQL забезпечує зручну інтеракцію з даними про категорії товарів, розміри, стани та самі товари. Вона дозволяє зберігати, організовувати та легко отримувати необхідну інформацію, сприяючи ефективному управлінню та аналізу товарного асортименту. На рисунку 3.3 можна побачити таблицю «Товари».



goodid [PK] integer	price numeric (10,2)	primecost numeric (10,2)	income numeric (10,2)	brand text	category text	goodsize text	condition text	goodstatus text	imagespath text	dateadded timestamp without time zone
1	200.00	50.00	150.00	Boohoo	Нижня білизна	M	Б/у	В наявності	D:\Idea_projects\KOSHTIMAGES\1	2023-05-15 16:53:43.969953
2	150.00	30.00	120.00	New Look	Спідниці	S	Ідеальний	Продано	D:\Idea_projects\KOSHTIMAGES\2	2023-05-15 16:55:24.970668
3	300.00	70.00	230.00	Boohoo	Плаття	M	Ідеальний	В наявності	D:\Idea_projects\KOSHTIMAGES\3	2023-05-15 16:57:00.172262
4	400.00	100.00	300.00	Boohoo	Плаття	L	Новий	В наявності	D:\Idea_projects\KOSHTIMAGES\4	2023-05-15 16:57:46.790071
5	300.00	60.00	240.00	ThePrettyLittleThing	Спідниці	S	Новий	Продано	D:\Idea_projects\KOSHTIMAGES\5	2023-05-15 16:59:26.755731

Рисунок 3.3 – Таблиця «Товари» у СУБД PostgreSQL

Структура бази даних, яку створили, дозволяє ефективно керувати даними про категорії товарів, розміри, стани та самі товари. Це важливо для забезпечення точності, доступності та надійності інформації, яка використовується в даному вебзастосунку. Можна легко отримувати, змінювати та видаляти дані з бази даних, що дозволяє динамічно змінювати та оновлювати даний вебзастосунок залежно від потреб користувачів.

Крім того, база даних дозволяє розробляти різноманітні запити та звіти, які допоможуть аналізувати та використовувати дані ефективним способом. Застосунок надає можливість здійснювати пошук товарів за певними критеріями, розраховувати фінансові показники, такі як прибуток, та аналізувати поведінку користувачів на основі даних з бази даних.

Загалом, розроблена база даних забезпечує потужний засіб для зберігання, організації та взаємодії з даними, необхідними для успішного функціонування та виконання завдань даного вебзастосунку.

### 3.3 Розробка серверної частини

У рамках розроблення даного застосунку було обрано IntelliJ IDEA в якості середовища розробки для даного проєкту. Такий вибір обґрунтований декількома причинами, і це середовище надає численні переваги, які сприятимуть успішному виконанню проєкту.

По-перше, IntelliJ IDEA є потужним та прогресивним інструментом для розробки програмного забезпечення на мові Java та інших мовах програмування. Воно має багатий набір функціональних можливостей, таких як автоматичне завершення коду, інтеграція з системами керування версіями, розширені можливості рефакторингу коду та візуальне налагодження. Це дозволяє розробникам ефективно працювати з кодом, збільшуючи продуктивність та якість розробки.

По-друге, IntelliJ IDEA надає велику кількість інтегрованих інструментів та плагінів, які спрощують процес розробки. Можна легко налаштувати робоче середовище згідно зі своїми потребами, використовуючи різноманітні інструменти для тестування, аналізу коду, контролю якості та багато іншого. Це дозволяє створювати високоякісний та надійний код, а також підтримувати його протягом усього життєвого циклу проєкту.

Крім того, IntelliJ IDEA має дружній та інтуїтивно зрозумілий інтерфейс, що сприяє зручному та комфортному робочому процесу. Воно надає можливість швидко переключатися між файлами, відстежувати зміни в коді та швидко знаходити помилки та проблеми. Також IntelliJ IDEA підтримує інтеграцію з іншими інструментами розробки, що полегшує спільну роботу команди та сприяє колективній розробці.

Узагалі, обрання IntelliJ IDEA допоможе у даному проєкті шляхом покращення продуктивності, зручності та якості розробки. Воно забезпечить потужні інструменти та зручне середовище для ефективної роботи над проєктом, що дозволить швидше досягти поставлених цілей та забезпечити високу якість даного програмного забезпечення.

Для розробки даного вебзастосунку було необхідно встановити та налаштувати JDK (Java Development Kit). JDK є набором інструментів, необхідних для розробки та виконання Java-програм.

Спочатку завантажили останню версію JDK з вебсайту Oracle або з іншого джерела, яке надає офіційні дистрибутиви JDK. Після завантаження встановлювального пакету виконали процес установки на даній розробницькій машині.

Після успішної установки JDK налаштували змінну середовища JAVA\_HOME для вказання шляху до встановленої JDK. Це дозволило системі знайти необхідні компоненти JDK під час компіляції та виконання програм.

Далі, перевірили, чи JDK правильно встановлений, відкривши командний рядок та введенням команди «java -version». Це дозволило переконатись, що JDK успішно інстальовано і готове до використання.

Після успішної установки та налаштування JDK могли розпочати розробку даного вебзастосунку з використанням Java. Використовували інтегроване середовище розробки, наприклад Eclipse або IntelliJ IDEA, для створення проєкту та написання коду. JDK забезпечував необхідні бібліотеки та інструменти для розробки вебзастосунків на Java.

В цілому, встановлення та налаштування JDK було необхідним кроком для успішної розробки даного вебзастосунку, оскільки воно забезпечувало середовище та інструменти для розробки та виконання Java-коду.

У рамках розроблення даного вебзастосунку було використано різні технології та інструменти для досягнення даних цілей. База даних PostgreSQL дозволила зберігати та керувати інформацією про товари та їх статуси. Використання JDBC драйвера дозволило здійснювати зв'язок між програмою та базою даних, виконувати SQL-запити та отримувати результати.

Даний Java Swing вебзастосунок надавав зручний та інтуїтивно зрозумілий інтерфейс для користувачів. Вікна та кнопки були розташовані з урахуванням зручності використання та ефективності роботи.

Архітектура класів даного застосунку розділена на чотири пакети та один головний клас під назвою «MainFrame». «MainFrame» відповідає за програмний код головного вікна програми і з нього можна викликати інші класи для подальшої роботи з ними.

Пакет «viewer» включає класи, спеціально розроблені для забезпечення перегляду таблиць з наборами даних з бази даних, зокрема таблиці «Товари». Основна мета цього пакету полягає в наданні користувачу зручного інтерфейсу для огляду різних типів товарів.

Клас «TrendingProductsViewer»: Цей клас відповідає за створення вікна, яке дозволяє користувачу переглядати актуальні товари. Він взаємодіє з базою даних, витягає необхідну інформацію про актуальні товари з таблиці «Товари» і відображає їх у вигляді зручної таблиці. Користувач може переглядати дані про товари, такі як ціна, вартість виробництва, дохід, бренд, категорія, розмір товару, стан, шлях до зображень товарів, дату додавання і дату продажу. Клас «SalesArchiveManager»: Цей клас забезпечує функціонал для перегляду проданих товарів. Він створює вікно, в якому користувач може переглядати список проданих товарів з бази даних. Клас використовує можливості бази даних для отримання відповідних записів з таблиці «Товари» з відповідним статусом. Крім того, він також використовує клас «DateChooser» для фільтрації товарів за обраною датою продажу. Клас «DateChooser»: Цей клас надає можливість користувачу обирати дату продажу у певному діапазоні. Він використовується у вікні «SalesArchiveManager» для встановлення фільтру по даті, що дозволяє користувачу переглядати товари, продані у обраний період.

Пакет «addupdate» надає функціонал для створення та редагування товарів. Він містить такі класи: «Uploader» – клас, який надає можливість створювати та редагувати товари у вигляді одного вікна, динамічно змінюючи його залежно від потреб користувача; «ImageFileFilter» – клас, що фільтрує файли та знаходить зображення, коли користувач завантажує зображення.

Останній пакет «styleAdds» включає один клас під назвою «Fonter». Цей клас надає можливість використовувати власні кастомні шрифти у будь-якій

області застосунку. Він дозволяє замінювати базові шрифти бібліотеки на кастомні та завантажені мануально для покращення зовнішнього вигляду та стилю інтерфейсу.

За такою архітектурою класів у даному вебзастосунку забезпечуємо структурованість та організованість програмного коду, а також розділення функціональності на відповідні компоненти, що полегшує його розробку та підтримку.

Програма, яку розробляємо, є віконним застосунком на Java Swing. Головне вікно програми має розмір 1920×1080 пікселів і налаштовано з такими параметрами:

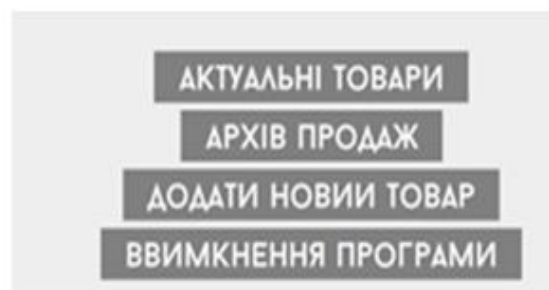
- `setExtendedState(JFrame.MAXIMIZED_BOTH);`
- `setUndecorated(true), setVisible(true).`

У головному вікні присутні два блоки. Перший блок має розмір 1920×360 пікселів і містить .png зображення з параметрами: горизонтальне позиціонування по середині (`horizontal:middle`), вертикальна позиція 100 пікселів (`Yposition:100`), ширина 680 пікселів (`Width:680`) та висота 240 пікселів (`Height:240`).

У другому блоку, що йде після першого блоку, знаходяться 5 кнопок, розташованих вертикально зі вирівнюванням по середині. Текст на кнопках відображається шрифтом Arial. Меню головного вікна можна побачити на рисунку 3.4 б), на рисунку 3.4 а) – зображено логотип застосунку.



а)



б)

Рисунок 3.4 – Головне вікно програми, інтерфейс застосунку

Дане вікно має декілька кнопок:

- перша кнопка має назву «Актуальні товари» і при натисканні відкривається вікно з назвою «ВІКНО2»;
- друга кнопка називається «Архів продажів» і при натисканні відкривається вікно з назвою «ВІКНО3»;
- третя кнопка позначена як «Додати новий товар» і при натисканні відкривається вікно з назвою «ВІКНО4»;
- четверта кнопка призначена для вимкнення програми.

Вікно «ВІКНО2» складається з трьох блоків, які розташовані горизонтально один за одним.

Перший блок має розмір 1920×160 пікселів і містить .png зображення з параметрами: горизонтальне позиціонування по середині (horizontal:middle), вертикальна позиція 20 пікселів (Yposition:20), ширина 340 пікселів (Width:340) та висота 120 пікселів (Height:120).

Другий блок має розмір 1920×720 пікселів і містить таблицю, яку можна прокручувати вертикально. Розміри також становлять 1920×720 пікселів. Таблиця містить дані з таблиці Goods з умовою «WHERE Goods.goodStatus = 'В наявності'». При натисканні на рядок таблиці, він виділяється.

Третій блок має розмір 1920×200 пікселів і містить 4 кнопки:

- кнопка «Змінити товар» (Кнопка10) відкриває вікно з назвою «ВІКНО11». Друге вікно має такі кнопки;
- кнопка «Позначити проданим» (Кнопка11) відправляє SQL-запит для зміни поля goodStatus на «Продано» в таблиці Goods, а також записує поточну дату та час у поле dateSold. Після цього таблиця з другого блоку перезавантажується з бази даних;
- кнопка «Видалити товар» (Кнопка12) відправляє SQL-запит для видалення поточного екземпляра з таблиці Goods, після чого таблиця з другого блоку перезавантажується з бази даних;
- кнопка «Повернутися» (Кнопка13) повертає до попереднього «ГОЛОВНОГО МЕНЮ».

На рисунку 3.5 можна побачити друге вікно, меню «Перегляд актуальних товарів».

id	Ціна	Собівартість	Заробіток	Бренд	Категорія	Розмір	Стан	Зображення	Дата публікації
1	200.0	50.0	150.0	Вооноо	Нижня білизна	M	Бу		2023-05-15 16:53:43 96953
3	300.0	70.0	230.0	Вооноо	Платья	M	Ідеальний		2023-05-15 16:57:00 172262
5	300.0	60.0	240.0	ThePonyStyleTeam	Слониці	S	Новий		2023-05-15 16:59:26 753731
2	150.0	30.0	120.0	New Look	Слониці	S	Ідеальний		2023-05-15 16:55:24 970668
4	400.0	100.0	300.0	Вооноо	Платья	L	Новий		2023-05-15 16:57:46 790071

ПОМІТИТИ ПРОДАЖ      ЗМІНИТИ ТОВАР      ВИДАЛИТИ ТОВАР      ПОВЕРНУТИСЯ

Рисунок 3.5 – Вікно «Перегляд актуальних товарів», інтерфейс застосунку

Для під'єднання програми до бази даних (СУБД PostgreSQL) у даному Java Swing застосунку довелося використовувати JDBC (Java Database Connectivity) драйвер. JDBC драйвер є стандартним інтерфейсом для з'єднання реляційних баз даних з програмами, написаними на мові Java.

Щоб встановити з'єднання з БД PostgreSQL, спочатку додали необхідну залежність JDBC драйвера до даного проєкту. Після цього, було використано відповідні класи та методи JDBC для створення підключення до бази даних.

У кодї програми вказали необхідні параметри підключення, такі як URL бази даних, ім'я користувача, пароль тощо. Після успішного підключення до БД PostgreSQL за допомогою JDBC драйвера, могли виконувати SQL-запити для отримання, збереження та взаємодії з даними з таблиць бази даних.

Використання JDBC драйвера дало можливість ефективно взаємодіяти з СУБД PostgreSQL у даному Java Swing застосунку, забезпечуючи стабільність та надійність підключення до бази даних і здійснення необхідних операцій з даними. Лістинг 3.1 містить Java підключення до Бази даних.

Лістинг 3.1 Підключення до бази даних:

```

package org.example.SQL;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
public class DatabaseConnection {
    private static final String URL = "jdbc:postgresql://localhost:5432/Kosht";
    private static final String USER = "postgres";
    private static final String PASSWORD = "password";
    public static Connection getConnection() throws SQLException {
        return DriverManager.getConnection(URL, USER, PASSWORD);
    }
}

```

При натисканні кнопки «Актуальні товари» у головному меню даної програми, відбувається процес отримання актуальних даних про товари з бази даних. Щоб це зробити, використовуємо JDBC (Java Database Connectivity) драйвер для з'єднання Системи Управління Базами Даних PostgreSQL з даним Java Swing застосунком.

У даному скрипті спочатку створюємо об'єкт «DefaultTableModel», який визначає структуру таблиці і назви її стовпців. Параметр «columnNames» визначає назви стовпців таблиці, а «0» вказує, що на даний момент у таблиці немає жодного рядка даних.

Потім виконуємо цикл, в якому обходяться результати запиту до бази даних. Метод «next()» переміщує курсор до наступного рядка результату

запиту. У циклі отримуємо значення кожного стовпця для поточного рядка за допомогою відповідних методів, наприклад, «getInt», «getDouble», «getString», які вказують на тип даних, який очікуємо для кожного стовпця.

Значення кожного стовпця зберігаємо у відповідні змінні, такі як «id», «price», «brand», тощо. Потім створюємо масив «rowData», який містить значення всіх стовпців для поточного рядка даних.

Далі, за допомогою методу «addRow», додаємо масив «rowData» до моделі таблиці «DefaultTableModel». Кожний рядок даних представляє окремий товар і містить значення його ідентифікатора, ціни, бренду, категорії, розміру, стану, шляху до зображення, дати додавання та дати продажу.

Нарешті, повертаємо модель таблиці «model», яка містить всі актуальні дані про товари. Це дозволяє відображати ці дані у вікні програми, де користувач може переглядати і взаємодіяти з актуальними товарами. Запит до бази даних та заповнення таблиці відбуваються відповідно до вибору користувача кнопки «Актуальні товари» у головному меню. Лістинг 3.2 містить Java код у якому робиться запит від Бази даних на отримання набору даних.

Лістинг 3.2 Завантаження даних з БД у вікні «Актуальні товари»:

```
DefaultTableModel model = new DefaultTableModel(columnNames, 0);
while (resultSet.next()) {
    int id = resultSet.getInt("goodid");
    double price = resultSet.getDouble("price");
    double primecost = resultSet.getDouble("primecost");
    double income = resultSet.getDouble("income");
    String brand = resultSet.getString("brand");
    String category = resultSet.getString("category");
    String goodsize = resultSet.getString("goodsized");
    String condition = resultSet.getString("condition");
    String imagespath = resultSet.getString("imagespath");
```

```

String dateadded = resultSet.getString("dateadded");
String datesold = resultSet.getString("datesold");
Object[] rowData = {id, price, primecost, income, brand, category,
goodsized, condition, imagespath, dateadded, datesold};
model.addRow(rowData);}
return model;

```

При натисканні на кнопку «Актуальні товари», відбувається зв'язок з базою даних PostgreSQL за допомогою JDBC драйвера. Запит до бази даних виконується за допомогою SQL-скрипту, який витягує необхідну інформацію про актуальні товари. Результати запиту обробляються та відображаються у вікні програми за допомогою таблиці, яка створена з використанням класу DefaultTableModel.

Загалом, даний програмний застосунок використовує різні компоненти та модулі, які співпрацюють між собою для забезпечення функціональності та надання користувачам зручного інтерфейсу для роботи з базою даних і управління товарами та продажами.

Головне вікно програми є стартовим пунктом, з якого користувач може вибрати необхідні дії. У головному вікні розташовані різні блоки та кнопки. Кнопки виконують різні дії, такі як перегляд актуальних товарів, доступ до архіву продажів, додавання нового товару та перегляд статистики. Крім того, є кнопка для вимкнення програми.

Крім того, програма має інші вікна, такі як «ВІКНО2», «ВІКНО3» і т.д., які відкриваються при взаємодії з різними кнопками. Наприклад, вікно «ВІКНО2» містить блоки з інформацією про товари, які можуть бути змінені або позначені як продані. Вікна взаємодіють з базою даних для отримання, збереження та оновлення інформації.

Під час розроблення даного вебзастосунку було приділено велику увагу дизайну та доцільності його інтерфейсу. Було вирішено не обмежуватись стандартними наборами стилів, доступними у Java Swing. Замість цього, було

вирішено розробити власний набір стилів, що дозволило покращити загальний вигляд та враження від користування застосунком.

Одним зі значних змін було встановлення кастомних шрифтів у даному вебзастосунку. Було обрано шрифти, які краще відповідали даним потребам і створювали бажану атмосферу у користувачів. Це дало застосунку оригінальності та стильності.

Також, звернули особливу увагу на дизайн кнопок та таблиць. Замість стандартних виглядів, впровадили зміни, щоб стиль кнопок відповідав загальному вигляду даного вебзастосунку. Було підібрано кольори, форми та інші елементи, щоб забезпечити зручність користувачам та покращити їх взаємодію з застосунком.

В результаті цих зусиль було розроблено вебзастосунок з привабливим та сучасним інтерфейсом. Дані зміни в дизайні додали зручності та естетичності, створюючи приємне враження для користувачів.

У підсумку, розробка даного вебзастосунку була важливим дослідницьким процесом, який дозволив розширити свої знання та навички в області програмування та баз даних. Було використано різні технології та інструменти, щоб забезпечити ефективну та надійну роботу даного вебзастосунку.

Одним із ключових компонентів даного проєкту була база даних PostgreSQL. Вона забезпечувала можливість зберігання великої кількості даних про товари та їх статуси. Використання JDBC драйвера дозволило здійснювати зв'язок між даною програмою та базою даних, виконувати SQL-запити та отримувати необхідні результати. Це дозволило ефективно управляти та оновлювати інформацію про товари у даному вебзастосунку.

Під час розробки також зосередилися на забезпеченні безпеки та надійності даного вебзастосунку. Було враховано вразливості та ризики, пов'язані з обробкою даних користувачів та доступом до бази даних. Застосування найкращих практик забезпечило захист інформації та запобігання несанкціонованому доступу до даних.

У результаті дана розробка вебзастосунку дозволила створити потужний та функціональний продукт, який задовольняє потреби даних користувачів. Набули цінного досвіду в розробці вебзастосунків та використанні баз даних, що стане у пригоді в подальших проєктах. Завдяки даним зусиллям та співпраці успішно втілили дані ідеї та створили вебзастосунок, який має потенціал для подальшого розвитку та успіху.

### 3.4 Інструкція користувача

Крок 1. Запустіть програму і зверніть увагу на головне вікно (рис. 3.6), яке відображається при запуску.



Рисунок 3.6 – Головне меню для навігації

Крок 2. У головному вікні знайдіть кнопку меню з назвою «Актуальні товари» (рис. 3.7). Натисніть на цю кнопку або виберіть цей пункт меню для відкриття вікна «Актуальні товари».

**АКТУАЛЬНІ ТОВАРИ**

Рисунок 3.7 – Кнопка «Актуальні товари» у головному меню

Крок 3. У вікні «Актуальні товари» (рис. 3.8) ви зможете переглядати товари, які знаходяться в наявності. Ви побачите інформацію про ціни, вартість виробництва, дохід, бренд, категорію, розмір товару, стан, шлях до зображень, дату додавання та інші деталі про товари. Маніпулюйте з даними таблиці за допомогою панелі кнопок знизу вікна.

**КОЛІТ** ПЕРЕГЛЯД АКТУАЛЬНИХ ТОВАРІВ

id	Ціна	Собівартість	Заробіток	Бренд	Категорія	Розмір	Стан	Зображення	Дата публікації
1	200.0	50.0	150.0	Вошео	Нижня білизна	M	Бу		2023-05-15 16:53:43.989953
3	300.0	70.0	230.0	Вошео	Плаття	M	Ідеальний		2023-05-15 16:57:00.172262
4	400.0	100.0	300.0	Вошео	Плаття	L	Новий		2023-05-15 16:57:46.790071

ПОМІТИТИ ПРОДАЖ      ЗМІНИТИ ТОВАР      ВИДАЛИТИ ТОВАР      ПОВЕРНУТИСЯ

Рисунок 3.8 – Кнопка «Повернутися» у меню «Актуальні товари»

Крок 4. У вікні «Актуальні товари» натисніть кнопку «Повернутися» (рис. 3.9), щоб повернутися до головного меню.

**ПОВЕРНУТИСЯ**

Рисунок 3.9 – Кнопка «Повернутися»

Крок 5. Щоб переглянути архів продаж, знайдіть кнопку меню з назвою «Архів продаж»(рис. 3.10). Натисніть на цю кнопку меню.

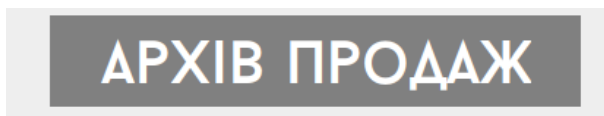


Рисунок 3.10 – Кнопка «Архів продаж»

Крок 6. У вікні «Архів продаж» (рис. 3.11) ви зможете переглядати товари, які були продані. Ви побачите інформацію про ціни, вартість виробництва, дохід, бренд, категорію, розмір товару, стан, шлях до зображень, дату додавання та дату продажу товарів. Маніпулюйте з даними таблиці за допомогою панелі кнопок знизу вікна. По завершенні роботи у цьому вікні поверніться до головного меню.

Коліт ПЕРЕГЛЯД АРХІВУ ПРОДАЖУ

id	Ціна	Собівартість	Заробіток	Бренд	Категорія	Розмір	Стан	Зображення	Дата публікації	date sold
1	300.0	60.0	240.0	ТыяПретут, elle	Спідниця	S	Новий		2023-05-15 16:59:26 765731	2023-05-15 17:02:49 939
2	150.0	30.0	120.0	New Look	Спідниця	S	Ідеальний		2023-05-15 16:55:24 970668	2023-05-15 17:02:37 974

ПАНЕЛЬ КНОПОК:

- ПОВЕРНУТИ В АКТУАЛЬНЕ
- ВИДАЛИТИ ТОВАР
- ПЕРЕГЛЯНУТИ ЗА ПЕРІОД...
- ПОВЕРНУТИСЯ

Рисунок 3.11 – Меню «Перегляд архіву продаж»

Крок 7. Щоб створити новий товар, знайдіть кнопку меню з назвою «Створення нового товару» (рис. 3.12). Натисніть на цю кнопку або виберіть цей пункт меню, щоб відкрити вікно, де ви зможете ввести необхідну інформацію про новий товар.

## ДОДАТИ НОВИЙ ТОВАР

Рисунок 3.12 – Кнопка «додати новий товар»

Крок 8. У вікні «Новий товар» (рис. 3.13) ви зможете створити нову одиницю товару, для цього занесіть дані до полів введення, оберіть зображення за потребою, та головне натисніть кнопку «Зберегти» (рис. 3.14). Після того як буде занесено новий запис у БД розгорнеться «Головне меню».

KOLLIT НОВИЙ ТОВАР

ID: 6

ЦІНА:

СОБІВАРІСТЬ:

ЗАРОБОТОК:

БРЕНД:

КАТЕГОРІЯ: Вербний одяг

РОЗМІР: XXX

СТАН: Новий

ЗОБРАЖЕННЯ: ОБЕРІТЬ ЗОБРАЖЕННЯ...

ЗБЕРЕГТИ

ПОВЕРНУТИСЯ

Рисунок 3.13 – Вікно «Новий товар»

ЗБЕРЕГТИ

Рисунок 3.14 – Кнопка «Зберегти»

Крок 9. Після завершення роботи з програмою, перейдіть до головного меню і знайдіть кнопку меню з назвою «Ввимкнення програми» (рис. 3.15). Натисніть на цю кнопку меню, щоб коректно закрити програму.

## ВВИМКНЕННЯ ПРОГРАМИ

Рисунок 3.15 – Кнопка «Завершення роботи»

Ці кроки допоможуть вам використовувати основні функції програми і навігуватися між різними вікнами та меню.

### 3.5 Тестування розробленої моделі

Після завершення розробки вебзастосунку було проведено тестування для перевірки функціональності та якості розробленої моделі. Тестування є важливою частиною процесу розробки, оскільки дозволяє виявити та виправити помилки, переконатися в стабільності програми та впевнитися, що вона задовольняє вимоги користувачів.

В процесі тестування були використані різні методи та підходи. Починаючи з модульного тестування, де окремі компоненти та функціональність програми перевірялися незалежно, до інтеграційного тестування, де взаємодія між компонентами та модулями перевірялася у комплексі. Також проводилося функціональне тестування, де перевірялися відповідність програми вимогам та очікуванням користувачів.

У процесі тестування було перевірено наступні аспекти розробленої моделі: перевірялася правильність роботи основних функцій програми, таких як перегляд актуальних товарів, архів продаж, створення нових товарів тощо. Впевнено, що програма виконує заплановані завдання та діє відповідно до очікувань. Тестування спрямовувалося на виявлення та усунення помилок, таких як збої програми, некоректна реакція на дії користувача або непередбачувана поведінка. Впевнено, що програма працює стабільно та надійно без непередбачених ситуацій.

Тестування включало оцінку зручності та інтуїтивності інтерфейсу користувача, перевірку наявності необхідних функцій та можливість їх правильного використання. Впевнено, що програма надає зручний та доступний інтерфейс для користувачів. Тестування також включало оцінку швидкодії програми, перевірку використання ресурсів системи та оптимізацію роботи програми. Впевнено, що програма працює ефективно та мінімізує негативний вплив на продуктивність системи.

В результаті тестування були виявлені деякі недоліки та помилки, які були виправлені та піддані повторному тестуванню для перевірки коректності

виправлень. Загалом, тестування дало позитивний результат, підтверджуючи якість та функціональність розробленої моделі.

За для перевірки якості розробленого вебзастосунок проведемо тестування на прикладній задачі:

Крок 1. Запустимо вебзастосунок, у головному меню перейдемо до вікна «Додати новий товар». Далі заповнимо усі поля для введення даних товару як зображено на рисунку 3.16, у поля, пов'язані із грошима, а саме «Ціна», «Собівартість», «Заробіток» заносимо суворо числа з плаваючою точкою, у інші поля для введення пишемо текст, у випадваючих списках обираємо підходящі для нас варіанти, та завантажимо зображення у застосунок з локального сховища скориставшись функцією виклику вікна «FileChooser», яка викликається по натисканні на кнопку із назвою «ОБЕРІТЬ ЗОБРАЖЕННЯ...».

КОШІТ ТОВАР 3

ID: 3

ЦІНА: 300.0

СОБІВАРТІСТЬ: 70.0

ЗАРОБІТОК: 230.0

БРЕНД: Boohoo

КАТЕГОРІЯ: Плаття

РОЗМІР: М

СТАН: Ідеальний

ЗОБРАЖЕННЯ: ОБЕРІТЬ ЗОБРАЖЕННЯ...

ЗБЕРЕГТИ

ПОВЕРНУТИСЯ

Рисунок 3.16 – Вікно «Актуальні товари», створення нової одиниці товару, заповненні поля для введення та завантаженні зображення

Крок 2. Перевіримо у СУБД PostgreSQL чи дописався новий запис у БД, знімок БД можна побачити на рисунку 3.17.

The screenshot shows a PostgreSQL database interface with a query window and a data output table. The query executed is:

```
1 SELECT * FROM public.goods
2 ORDER BY goodid ASC
```

The data output table contains the following records:

goodid [PK] integer	price numeric (10,2)	primecost numeric (10,2)	income numeric (10,2)	brand text	category text	goodsize text	condition text	goodstatus text	imagespath text	dateadded timestamp w	datesold timestamp without time zone
1	200.00	50.00	150.00	Voohoo	Нижня бі...	M	Б/у	В наявно...	D:\idea_p...	2023-05-...	[null]
2	150.00	30.00	120.00	New Look	Спідниці	S	Ідеальний	Продано	D:\idea_p...	2023-05-...	2023-05-15 17:02:37.974
3	300.00	70.00	230.00	Voohoo	Плаття	M	Ідеальний	В наявно...	D:\idea_p...	2023-05-...	[null]
4	400.00	100.00	300.00	Voohoo	Плаття	L	Новий	В наявно...	D:\idea_p...	2023-05-...	[null]
5	300.00	60.00	240.00	ThePretty...	Спідниці	S	Новий	Продано	D:\idea_p...	2023-05-...	2023-05-15 17:02:40.039

Рисунок 3.17 – СУБД PostgreSQL вибірка з таблиці «Goods»

Крок 3. Перевіримо у вікні «Перегляд актуальних товарів», Вікно «Перегляд актуальних товарів» можна побачити на рисунку 3.18. Як можна побачити функціонал створення нової одиниці товару працює справно.

The screenshot shows a web application interface titled "KOLLIT ПЕРЕГЛЯД АКТУАЛЬНИХ ТОВАРІВ". It displays a table of goods with the following columns: id, Ціна, Собівартість, Заробіток, Бренд, Нижня білизна, Категорія, Розмір, Стан, Зображення, and Дата публікації. The table contains three rows of data:

id	Ціна	Собівартість	Заробіток	Бренд	Нижня білизна	Категорія	Розмір	Стан	Зображення	Дата публікації
1	200.0	50.0	150.0	Voohoo	Нижня білизна	Категорія	M	Б/у	[Image]	2023-05-15 16:53:43.969553
3	300.0	70.0	230.0	Voohoo	Плаття	Категорія	M	Ідеальний	[Image]	2023-05-15 16:57:00.172262
4	400.0	100.0	300.0	Voohoo	Плаття	Категорія	L	Новий	[Image]	2023-05-15 16:57:46.780071

At the bottom of the interface, there are four buttons: ПОМІТИТИ ПРОДАЖ, ЗМІНИТИ ТОВАР, ВИДАЛИТИ ТОВАР, and ПОВЕРНУТИСЯ.

Рисунок 3.18 – Вікно «Перегляд актуальних товарів»

Крок 4. Перевіримо у вікні «Перегляд архіву продажу», Вікно «Перегляд архіву продажу» можна побачити на рисунку 3.19. Далі натиснемо на кнопку «ПЕРЕГЛЯНУТИ ЗА ПЕРІОД...», за для перевірки функції вибіркового перегляду даних товарів з бази даних у PostgreSQL.

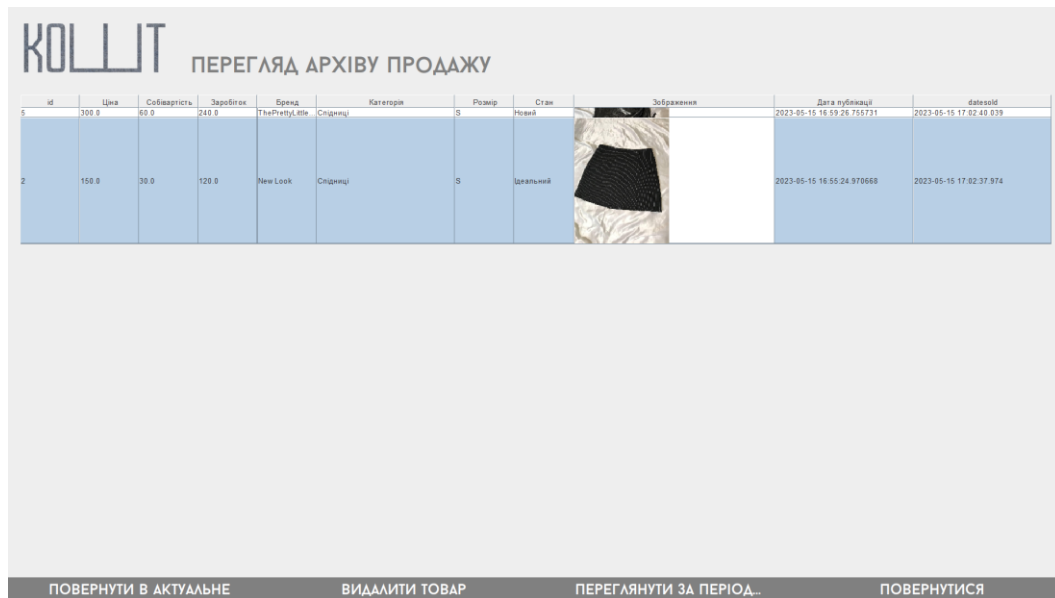


Рисунок 3.19 – Вікно «Перегляд архіву продажу»

Крок 5. Як можна побачити на рисунку 3.20 вікно відчинилось й функція працює справно.

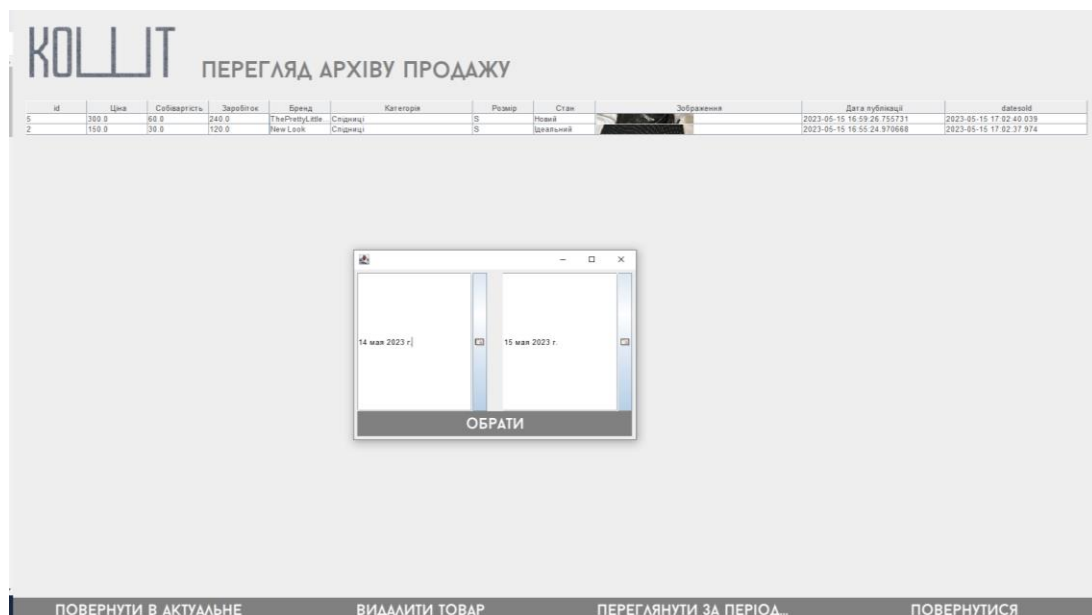


Рисунок 3.20 – Вікно «Обрання дат»

Після завершення всіх необхідних етапів тестування та впровадження необхідних виправлень, можна підвести підсумки щодо ефективності та якості розробленої моделі. Результати тестування свідчать про те, що модель успішно пройшла всі випробування та відповідає вимогам, поставленим перед нею. Вона виявила високу стабільність та надійність у роботі, забезпечуючи користувачам бажаний функціонал та задоволення від використання.

Процес тестування виявив деякі недоліки та помилки, але завдяки цьому було можливо виправити та вдосконалити розроблену модель. Виправлення виявлених проблем дозволило підвищити якість та надійність вебзастосунку, забезпечивши його безперебійну та оптимальну роботу. Користувачам буде забезпечено зручний та ефективний інструмент, що задовольнятиме їхні потреби та вимоги.

Загальною метою тестування було забезпечення високої якості розробленої моделі, а також впевненості у її правильному функціонуванні. Завдяки тестуванню було досягнуто цих цілей, а отримані результати підтверджують успішність розробки вебзастосунку та його відповідність вимогам та очікуванням користувачів.

## ВИСНОВКИ

У рамках кваліфікаційної роботи був розроблений і реалізований вебзастосунок для ведення обліку одиниць товару для продажу, які будуть мати електронне представлення для операцій над ними (додаток А).

У процесі розробки були використані різні технології та інструменти. База даних PostgreSQL була використана для зберігання та керування інформацією про товари та їх атрибути. Використання JDBC драйвера дозволило забезпечити зв'язок між програмою та базою даних.

Особлива увага приділялась дизайну та доцільності інтерфейсу. Власний набір стилів та шрифтів був розроблений для покращення зовнішнього вигляду застосунку. Це дало можливість створити привабливий та зручний для користувачів інтерфейс, що сприяє комфортному використанню програми.

Архітектура класів була створена з урахуванням зручності та логічного розподілу функціоналу. Розділення на пакети дозволило структурувати код та забезпечити зручний доступ до потрібних класів. Головний клас «MainFrame» виконує роль головного вікна програми та забезпечує взаємодію з іншими.

Після завершення розробки було проведено тестування розробленої моделі. В результаті були виявлені деякі помилки та недоліки, які були виправлені для забезпечення якості та надійності програми. Тестування підтвердило успішність розробки, а модель продемонструвала стабільну та оптимальну роботу, задовольняючи потреби користувачів.

Розроблений вебзастосунок відповідає вимогам та надає користувачам очікуваний функціонал. В процесі роботи було досягнуто високої якості, стабільності та надійності програмного забезпечення. Отримані результати будуть використані для подальшого вдосконалення та розвитку системи у відповідності до потреб користувачів та вимог ринку.

Результати роботи апробовано у вигляді тез доповідей під час III Міжнародної науково-практичної конференції «MODERN PROBLEMS OF SCIENCE, EDUCATION AND SOCIETY» [37].

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Творошенко, І. С. (2018). Особливості застосування сучасних принципів штучного інтелекту до розробки ефективних механізмів моделювання складних систем. *Science and Technology of the Present Time: Priority Development Directions of Ukraine and Poland*, 101-104.
2. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Zeghid M. (2022) Tools for fast metric data search in structural methods for image classification, *IEEE Access*, 10, pp.
3. Мантула, Е., & Машталір, С. (2014). Adaptive polynomial neuronetwork predicting model of time series and its training. *Eastern-European Journal Of Enterprise Technologies*, 2(4(68)), 16.
4. Gorokhovatskyi V., Tvoroshenko I., Kobylin O., and Vlasenko N. (2023) Search for visual objects by request in the form of a cluster representation for the structural image description, *Advances in Electrical and Electronic Engineering*, 21(1), pp. 21-24.
5. Schwinn, B. (1989). Parallelität und intelligentes Backtracking in logischen Programmen / Parallelism and Intelligent Backtracking in Logic Programs. *It – Information Technology*, 31(4).
6. Гороховатський В.О., Творошенко І.С., Чмутов Ю.В. (2022) Застосування систем ортогональних функцій для формування простору ознак у методах класифікації зображень, *Сучасні інформаційні системи*, 6(3), С. 3-5.
7. Гороховатський В., Передрій О., Творошенко І., Марков Т. (2023) Матриця відстаней для множини компонентів структурного опису як інструмент для створення класифікатора зображень, *Сучасні інформаційні системи*, 7(1), С. 6-11.
8. Творошенко, І.С. (2021). Технології прийняття рішень в інформаційних системах: навч. посібник. *Харків: ХНУРЕ*.
9. Гороховатський, В.О., & Творошенко, І.С. (2021). Методи інтелектуального аналізу та оброблення даних: навч. посібник. *ХНУРЕ*.

10. Кобилін, О.А., & Творошенко, І.С. (2021). Методи цифрової обробки зображень: навч. посібник. Харків: ХНУРЕ.
11. Bodyanskiy, Ye. V., Pliss, IP., Shafronenko, A. Yu., & Kalynychenko, OV. (2022). Нечітка довірча кластеризація даних на основі аналізу щільності розподілу даних та їх піків. *Radio Electronics, Computer Science, Control*, 3, 58.
12. Гороховатський В.О., Творошенко І.С. (2022) Аналіз багатовимірних даних за описом у формі множини компонент: монографія. Харків: ХНУРЕ, 123 с.
13. Mashtalir, S., & Mashtalir, V. (2020). Spatio-temporal video segmentation. In *Advances in Spatio-Temporal Segmentation of Visual Data* (pp. 161-210). Springer, Cham.
14. Mashtalir, S., & Mashtalir, V. (2016, August). Sequential temporal video segmentation via spatial image partitions. In *2016 IEEE First International Conference on Data Stream Mining & Processing (DSMP)* (pp. 239-242). IEEE.
15. Bodyanskiy, Ye. V., Pliss, IP., & Shafronenko, A. Yu. (2022). Кластеризація масивів даних на основі комбінованої оптимізації функцій щільності розподілу та еволюційного методу котячих зграй. *Radio Electronics, Computer Science, Control*, 4, 61-61.
16. Bodyanskiy, Y. V., Tyshchenko, O. K., & Mashtalir, S. V. (2019, June). Fuzzy clustering high-dimensional data using information weighting. In *International Conference on Artificial Intelligence and Soft Computing* (pp. 385–395). Springer, Cham.
17. Mashtalir, S. V., Stolbovyi, M. I., & Yakovlev, S. V. (2019). Clustering video sequences by the method of harmonic k-means. *Cybernetics and Systems Analysis*, 55(2), 200-206.
18. Bodyanskiy, Y., Grimm, P., Mashtalir, S., & Vinarski, V. (2010, July). Fast training of neural networks for image compression. In *Industrial Conference on Data Mining* (pp. 165-173). Springer, Berlin, Heidelberg.
19. Хоггер К. (2008). Введення в логічне програмування. М. Мир. С. 348.

20. Lyashenko, V., Sinelnikova, T., Zeleniy, O., & Babker, A. M. A. (2020, November 7). Specifics of Using Image Processing Techniques for Blood Smear Analysis. *Journal La Medihealthico*, 1(6), 1-6.

21. Shafronenko, A. Yu., Bodyanskiy, Ye. V., & Holovin, O. O. (2023). Кластеризація масивів даних на основі модифікованого алгоритму сірого вовка. *Radio Electronics, Computer Science, Control*, 1, 73-73.

22. Andrew Ramdayal. (2018). *A Guide to the Project Management Body of Knowledge (PMBOK Guide) – 5th ed.* Project Management Institute, pp 616.

23. Bilonoh, B., Bodyanskiy, Y., Kolchygin, B., & Mashtalir, S. (2022). Tunable Polynomial Activation Function for Artificial Neural Networks. In *Lecture Notes in Computational Intelligence and Decision Making: 2021 International Scientific Conference «Intellectual Systems of Decision-making and Problems of Computational Intelligence»*, Proceedings (pp. 624-633). Springer International Publishing.

24. Bogucharsky, S. I., Mashtalir, S. V., & Stolbovyi, M. I. (2018). Fast Detection of Changes in Properties of Multidimensional Time Series Based on an Identification Approach to Model Ensemble. *Systems of Information Processing*, 3, 74-78. Ivan Kozhedub Kharkiv National Air Force University.

25. Hu, Z., Mashtalir, S. V., Tyshchenko, O. K., & Stolbovyi, M. I. (2018). Clustering Matrix Sequences Based on the Iterative Dynamic Time Deformation Procedure. *International Journal of Intelligent Systems and Applications*, 10(7), 66-73. Modern Education and Computer Science Press.

26. Bodyanskiy, Y., Shafronenko, A., & Mashtalir, S. (2020). Online Robust Fuzzy Clustering of Data with Omissions Using Similarity Measure of Special Type. In *Lecture Notes in Computational Intelligence and Decision Making* (pp. 637-646). Springer.

27. Mashtalir, S., Mashtalir, V., & Stolbovyi, M. (2018, August 21). Representative Based Clustering of Long Multivariate Sequences with Different Lengths. In *2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP)* (pp. 545-548). IEEE.

28. Cohen M.A., Grossberg S. (2017). Absolute stability of global pattern formation and parallel memory storage by competitive neural networks//IEEE Trans. Syst., Man, Cybern. V. 13. N 5. P. 815 – 826.

29. Шафроненко, А., Бодяньський, Є., & Плісс, І. (2022). Нечіткі методи інтелектуального аналізу даних. *GlobeEdit*, 105.

30. Mustafa, S. K., Kopot, M., Ahmad, M. A., Lyubchenko, V., & Lyashenko, V. (2020). Interesting applications of mobile robotic motion by using control algorithms. *WARSE*.

31. Gadetska, S., Gorokhovatskyi, V., Stiahlyk, N., & Vlasenko, N. (2022). Aggregate Parametric Representation of Image Structural Description in Statistical Classification Methods. *CEUR22*.

32. Kobylin, O., & Lyashenko, V. (2020). Time series clustering based on the k-means algorithm. *Journal La Multiapp*.

33. Dubnitskiy, V., Kobylin, A., Kobylin, O., Kushneruk, Y., & Sheviakov, Y. (2022). Calculation of function values of a complex variable  $z$  with an interval argument defined in hyperbolic form. *Advanced Information Systems*, 6(3), 83-91.

34. Гороховатський, В.О., & Творошенко, І.С. (2022). Аналіз багатовимірних даних за описом у формі множини компонент. Монографія, ХНУРЕ, 124 с.

35. Zeleniy, O., Rudenko, D., Lyubchenko, V., & Lyashenko, V. (2022). Image Processing as an Analysis Tool in Medical Research. *IJAAR*.

36. Cherednichenko, O., Vovk, M., Yanholenko, O., & Yakovleva, O. (2020). Towards the Technology of Employers' Requirements Collection Development. *Integrated Computer Technologies in Mechanical Engineering: Synergetic Engineering*, 1113, 228-239. Springer Nature.

37. Машталір С.В., Гавришева К.А., (2023) Особливості проектування застосунків із використанням високорівневої мови та згідно з об'єктно-орієнтованою парадигмою. III Міжнародна науково-практична конференція «MODERN PROBLEMS OF SCIENCE, EDUCATION AND SOCIETY», С. 76-77.