

Галицький фаховий коледж імені В'ячеслава Чорновола
відділення комп'ютерних технологій
циклова комісія інформатики та комп'ютерних дисциплін

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач відділенням

комп'ютерних технологій

Стефурак Н.А. / _____ /

підпис

« ____ » _____ 2023 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проєкту

освітньо-кваліфікаційного рівня «молодший спеціаліст»

зі спеціальності 122 «Комп'ютерні науки»

на тему : «Інформаційна система обліку матеріальних цінностей для
автоматизованої системи введення торгівлі»

Студентки групи КН-41 Кушко Т.В.

(підпис)

Керівник проєкту

Івасьєв С.В.

(підпис)

Консультанти:

з техніко-економічного
обґрунтування

Меленчук Л.І.

(підпис)

нормоконтролер

Кульчинська Н.З.

(підпис)

Тернопіль – 2023

Галицький фаховий коледж імені В'ячеслава Чорновола
відділення комп'ютерних технологій
циклова комісія інформатики та комп'ютерних дисциплін

ЗАТВЕРДЖУЮ
Завідувач відділенням
комп'ютерних технологій

Стефурак.Н.А. / _____ /
підпис

«___» _____ 2023 р.

ЗАВДАННЯ

на дипломне проектування
на здобуття освітньо-кваліфікаційного рівня «молодший спеціаліст»
студенту _____
(прізвище, ім'я та по-батькові студента)

1. Тема проєкту _____

затверджена наказом по коледжу від “___” _____ 2022 р., № _____

2. Термін здачі студентом завершеного проєкту “___” _____ 2023 р

3. Вихідні дані до проєкту _____

4. Перелік питань, які повинні бути розроблені в проєкті: _____

а) основна частина _____

б) техніко-економічного обґрунтування _____

5. Перелік графічного матеріалу _____

6. Консультанти проєкту: _____

Розділ	Консультанти	Підпис, дата	
		Завдання видано	Завдання прийнято
з техніко-економічного обґрунтування	<div></div> <div>(вчена ступень, звання П.І.Б.</div> <div></div> <div>консультанта)</div>		

КАЛЕНДАРНИЙ ПЛАН

дипломного проєктування

№ п/п	Найменування етапу	Терміни	
		початку	завершення
1.	Вибір теми та ознайомлення з вимогами до дипломного проєкту.	20.11.22 р.	06.12.22 р.
2.	Огляд типових рішень та написання відповідного розділу проєкту.	06.12.22 р.	26.01.23 р.
3.	Дослідження технологій реалізації та написання відповідного розділу проєкту.	26.01.23 р.	14.02.23 р.
4.	Розробка функціональних вимог до проєкту та робота над структурою програмного продукту. Написання відповідного розділу проєкту.	14.02.23 р.	02.03.23 р.
5.	Встановлення та налаштування середовища реалізації та написання відповідного розділу проєкту.	02.03.23 р.	16.03.23 р.
6.	Проектування програмного засобу (функціоналу, інтерфейсу, бази даних продукту) та написання відповідного розділу проєкту.	16.03.23 р.	17.04.23 р.
7.	Реалізація та налаштування програмного засобу та написання відповідного розділу проєкту.	17.04.23 р.	03.05.23 р.
8.	Доопрацювання модулів.	05.05.23 р.	18.05.23 р.
9.	Тестування та налагодження програмного продукту.	18.05.23 р.	01.06.23 р.
10.	Опрацювання економічного розділу дипломного проєкту та оформлення спеціального розділу.	20.05.23 р.	05.06.23 р.
11.	Робота над оформленням пояснювальної записки.	05.06.23 р.	12.06.23 р.
12.	Попередній захист дипломного проєкту та доопрацювання.	15.06.23 р.	
13.	Підготовка до захисту дипломного проєкту.	15.06.22 р.	22.06.23 р.
14.	Захист дипломного проєкту.	27.06.2023	27.06.2023

7. Дата видачі завдання ” ____ ” _____ 2022 р.

Керівник _____ / _____

Завдання прийняв до виконання _____ / _____

Реферат

Дипломний проєкт. Тема: «Інформаційна система обліку матеріальних цінностей для автоматизованої системи введення торгівлі» 63 сторінки, 30 рисунки, 11 таблиці, 9 джерел, 3 додатки.

Предмет дослідження – процеси опрацювання інформаційних потоків в системах обліку матеріальних цінностей.

Об'єктом дослідження є програмний засіб для ведення обліку матеріальних цінностей для автоматизованої системи введення торгівлі.

Метою цього дипломного проєкту є розробка інформаційної системи обліку матеріальних цінностей для автоматизованої системи ведення торгівлі з використанням сучасних технологій програмування, баз даних та інтерфейсів користувача.

Для розробки програми було обрано середовище розробки Visual Studio 2022, середовище спеціалізується на створенні програмних засобів для комп'ютерів на базі операційної системи Windows. Visual studio 2022 підтримує мови програмування: C, C++. Реалізація програмного засобу виконана на мові програмування C, де зрозумілий синтаксис, велика кількість програмної документації та середовище для розробки програмного інтерфейсу.

СКЛАД, SQL, ОБЛІК МАТЕРІАЛЬНИХ ЦІННОСТЕЙ, ПРИХІД, РОЗХІД.

Abstract

Diploma project. Topic: "Information system of accounting for material values for the automated system of entering trade" 63 pages, 30 figures, 11 tables, 9 sources, 3 addition.

The subject of the study is the processes of processing information flows in accounting systems of material values.

The object of the study is a software tool for keeping records of material values for an automated trade entry system.

The goal of this diploma project is the development of an information system of accounting for material values for an automated trading system using modern programming technologies, databases and user interfaces.

The Visual studio 2022 development environment was chosen for program development, the environment specializes in creating software tools for computers based on the Windows operating system. Visual studio 2022 supports programming languages: C, C++. The implementation of the software tool is made in the C programming language, where the syntax is clear, a large amount of software documentation and an environment for the development of the software interface.

STORAGE, SQL, ACCOUNTING OF MATERIAL VALUES, INCOME, EXPENDITURE.

ЗМІСТ

Вступ.....	6
1 Аналіз існуючих рішень і постановка завдань.....	8
1.1 Обґрунтування доцільності створення систем	8
1.2 Аналіз існуючих рішень.....	8
1.3 Обґрунтування доцільності створення системи та постановки завдання.	14
2 Проєктування інформаційної системи.....	16
2.1 Формалізація вимог	16
2.2 Побудова дерева функцій та проєктування структури системи	17
2.3 Проєктування бази даних.....	19
2.4 Проєктування користувацького інтерфейсу	29
3 Реалізація та тестування інформаційної системи	31
3.1 Опис технологій та засобів реалізації.....	31
3.2 Реалізація основних функцій інформаційної системи.	33
3.3 Реалізація користувацького інтерфейсу системи.	38
3.4 Тестування інформаційної системи обліку матеріальних цінностей для автоматизованої системи введення торгівлі.	44
4 Техніко-економічне обґрунтування	50
4.1 Аналіз ринку.....	50
4.2 Розрахунок витрат на проєктування	54
4.3 Обґрунтування необхідності розробки.....	56
Висновки	60
Перелік джерел посилання	62
Додатки	63

					ДП.КН. 23.515.22 000 ПЗ					
Змн.	Арк.	№ докум.	Підпис	Дата	Інформаційна система обліку матеріальних цінностей для автоматизованої системи ведення торгівлі			Літ.	Арк.	Акрушів
Розробив		Кушко Т. В.								
Перевірів		Івасьєв С.В.							5	76
Реценз.		Кузик В.М.						ГФК.ВКТ.ЦКІКД КН-41		
Н. Контр.		Кульчинська Н.З.								
Затверд.		Стефурак Н.А.								

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

SQL – Structured Query Language (мова структурованих запитів).

БД – База даних.

ІТ – Інформаційні технології.

ІСО – Інформаційна система обліку.

ПЗ – Програмне забезпечення.

ОС – Операційна система.

					ДП.КН. 23.515.22 000 ПЗ	Арк.
						5
Зм.	Арк.	№ докум.	Підп.	Дата		

ВСТУП

Інформаційна система обліку матеріальних цінностей для автоматизованої системи ведення торгівлі є важливим інструментом для будь-якої торговельної компанії, що прагне оптимізувати та забезпечити точність своєї діяльності. Ця інформаційна система дозволяє автоматизувати процес обліку матеріальних цінностей, що використовуються в торговельній діяльності, та забезпечує точність та швидкість ведення обліку. Інформаційна система також дозволяє керувати доступом користувачів, створювати звіти про рух матеріальних цінностей та моніторити їх рух.

Розробка цієї інформаційної системи є актуальною та значимою, оскільки у сучасному світі торговельні компанії повинні дотримуватися високих стандартів якості та ефективності ведення діяльності. Застосування інформаційної системи обліку матеріальних цінностей допомагає компаніям зменшити кількість помилок, що можуть бути зроблені при веденні обліку вручну, зберегти час та ресурси на облік матеріальних цінностей та забезпечити точність даних.

Метою цього дипломного проєкту є розробка інформаційної системи обліку матеріальних цінностей для автоматизованої системи ведення торгівлі з використанням сучасних технологій програмування, баз даних та інтерфейсів користувача.

Метою проєкту є :

- Розробити інформаційну систему обліку матеріальних цінностей, яка забезпечує ефективну автоматизацію процесів введення торгівлі.
- Забезпечити точний та швидкий облік матеріальних цінностей, що використовуються в торговельній діяльності.
- Зменшити час та зусилля, витрачені на ручний облік матеріальних цінностей, завдяки автоматизації процесів введення торгівлі.

					ДП.КН. 23.515.22 000 ПЗ	Арк.
						6
Зм.	Арк.	№ докум.	Підп.	Дата		

– Забезпечити зменшення витрат на управління матеріальними цінностями шляхом ефективного планування ресурсів, оптимізації запасів та уникнення зайвих запасів або недостачі товарів.

Використання такої системи допоможе торговельним компаніям забезпечити точний та ефективний облік матеріальних цінностей, що є важливим елементом успішної торговельної діяльності. Крім того, розробка цієї інформаційної системи може сприяти підвищенню конкурентоспроможності торговельних компаній на ринку, що є важливим чинником у сучасних умовах бізнесу.

Для досягнення поставленої мети, у цьому дипломному проєкті будуть розглянуті та використані сучасні технології програмування, баз даних та інтерфейсів користувача, що дозволить розробити функціональну та ефективну інформаційну систему обліку матеріальних цінностей для автоматизованої системи ведення торгівлі.

Предмет дослідження – процеси опрацювання інформаційних потоків в системах обліку матеріальних цінностей.

Об'єктом дослідження є програмний засіб для ведення обліку матеріальних цінностей для автоматизованої системи введення торгівлі.

Цей передбачає розробку та впровадження вискоелективної та надійної інформаційної системи, що дозволить забезпечити автоматизацію процесів обліку матеріальних цінностей. Розробка включатиме аналіз вимог, проєктування архітектури системи, розробку програмного забезпечення, налаштування баз даних, інтеграцію з іншими системами, тестування, впровадження та підтримку системи.

Результати цього дипломного проєкту можуть бути використані для практичного застосування в роботі торговельних компаній та забезпечити покращення їх ефективності та якості роботи. Отже, цей дипломний проєкт має велике значення для розвитку торговельної сфери та інформаційних технологій в цій галузі.

					ДП.КН. 23.515.22 000 ПЗ	Арк.
						7
Зм.	Арк.	№ докум.	Підп.	Дата		

1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ І ПОСТАНОВКА ЗАВДАНЬ

1.1 Обґрунтування доцільності створення систем

З кожним роком кількість пристроїв та їх технічні можливості зростають, що свідчить про постійний технологічний прогрес. Щоб забезпечити стабільну роботу з програмними продуктами, може знадобитися оновлення комп'ютерного обладнання або придбання нових модулів, наприклад, факсів, принтерів, ноутбуків або планшетів, яких немає у вашій компанії. З розвитком комп'ютерних технологій, обладнання стає менш важливим, тому компанії потрібно частіше оновлювати свої апаратні та програмні продукти. Це призводить до того, що врахувати всі ці фактори стає все складніше.

Створення систем обліку є невід'ємною складовою успішного управління будь-якою організацією. Завдяки таким системам можна зібрати, обробити та зберігти інформацію про фінансові та інші операції, що здійснюються в організації. Це, у свою чергу, дозволяє приймати обґрунтовані рішення стосовно розвитку бізнесу та планування його діяльності, що є надзвичайно важливим.

В деяких випадках програмне забезпечення для управління взаємодією з клієнтами може бути дорогим, складним та доступним лише як частина великої системи управління взаємодією з клієнтами, що недоступна в Інтернеті. У зв'язку з цим, розробка веб-сервісів, які призначені для автоматизації конкретних процесів в компаніях, є важливою та популярною.

1.2 Аналіз існуючих рішень

За два роки, до 2023 року, очікується значне збільшення кількості додатків, які будуть мати широкий спектр функцій. Перед тим, як розпочати розробку нового програмного продукту, необхідно провести огляд вже існуючих аналогів.

					ДП.КН. 23.515.22 000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		8

Для порівняльного аналізу веб-додатків, які призначені для обліку різних товарів, необхідно зібрати відповідну інформацію.

Для проведення аналізу було вибрано три сервіси, зокрема IT Invent, Inspector та PrintStore.

Назва сервісу: "IT Invent" (рисунк 1.1). Сервіс має можливість вести облік комп'ютерів, принтерів, програм, комплектуючих, ремонту техніки, обслуговування техніки, надходжень і переміщень, співробітників, а також здійснювати ведення графіка роботи і розкладу.

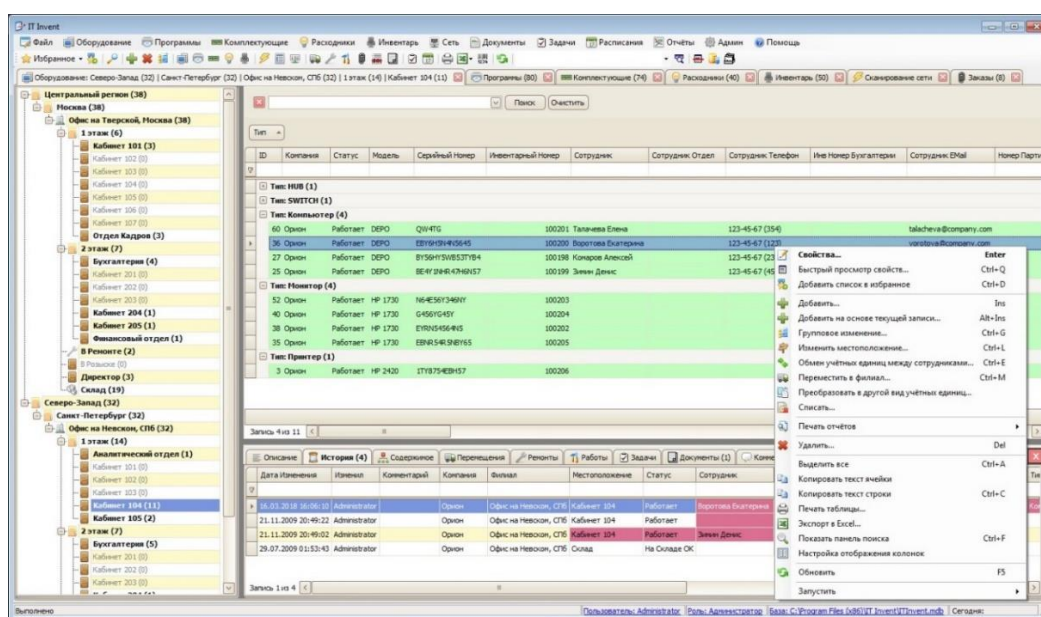


Рисунок 1.1 – Інтерфейс програми «IT Invent»

Основні функціональні можливості програми «IT Invent» включають:

- Сканування мережі та перегляд списку пристроїв за допомогою діапазонів IP-адрес та використання інтерфейсів WMI та SNMP.
- Автоматичне генерування інвентарних номерів для облікових одиниць.
- Зберігання історії змін ключових полів для всіх типів облікових записів.
- Облік підрозділів та розбиття об'єктів обліку по відділах або філіях.
- Облік ремонтів та профілактичних робіт обладнання.

- Логічне прив'язування програм та компонентів до обладнання.
 - Можливість розбиття всіх облікових одиниць на відділи та закріплення їх за відповідними співробітниками.
 - Управління базами даних постачальників та сервісних організацій.
- Робота з партнерами та відділ планування.

Перевагами сервісу IT Invent є широкий спектр функцій, що дозволяє вести детальний облік різних аспектів комп'ютерної техніки та її обслуговування, включаючи надходження і переміщення, ремонт, обслуговування, а також облік співробітників та їх графіки роботи.

Однак, сервіс IT Invent може мати наступні недоліки: складний та важкодоступний інтерфейс, що може потребувати додаткового навчання або досвіду користування; обмеження у можливостях інтеграції з іншими програмними продуктами; висока вартість користування або підписки на сервіс.

Враховуючи ці фактори, перед використанням сервісу IT Invent, може знадобитися додаткове дослідження та порівняння з іншими аналогічними програмними продуктами.

Наступною програмою, яку ми розглянемо, буде "PrintStore" (рисунк.1.2)

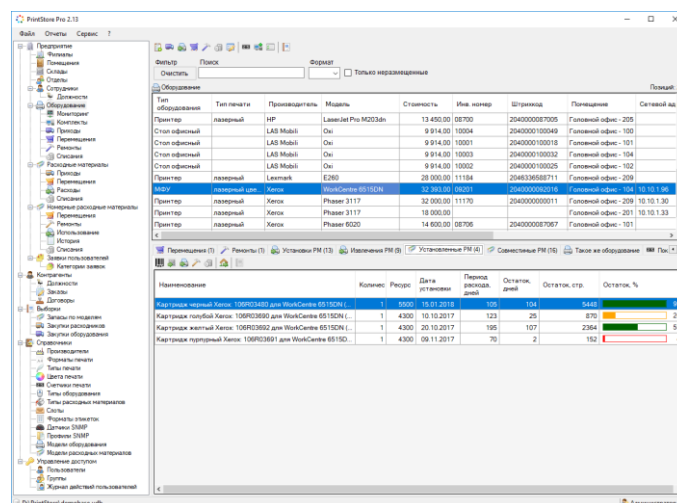


Рисунок 1.2 – Інтерфейс програми «PrintStore»

Програма "PrintStore" була розроблена компанією "ПерфектСофт" і коштує від 0 до 7000 гривень.

Ця програма призначена для вирішення наступних завдань: управління базою даних підприємства, що містить інформацію про клієнтів, співробітників та інші деталі, систематичний облік та документування процесів ремонту обладнання, контроль та реєстрація руху витратних матеріалів для оптимального управління запасами, облік договорів, укладених зі сервісними організаціями.

З метою ефективного управління обслуговуванням, систематизована обробка та управління заявками користувачів для оперативного вирішення їх потреб, відображення актуальних залишків матеріалів для контролю та планування запасів, моніторинг роботи мережевих принтерів для забезпечення їх безперебійної роботи та ефективного використання ресурсів.

"PrintStore" пропонує ряд функціональних можливостей, серед яких є ведення довідників підприємства, облік ремонтів обладнання та інші корисні функції.

Переваги програми PrintStore:

– Дозволяє вести облік руху витратних матеріалів, що є корисною функцією для підприємств, що мають складський облік.

– Дозволяє моніторити роботу мережевих принтерів та зручно відображати актуальні залишки матеріалів, що сприяє своєчасному плануванню закупівель та уникненню непотрібних витрат на обладнання.

– Можливість відображення детальної інформації про обладнання, ремонт та технічне обслуговування, що є корисною функцією для підприємств з багатим технічним парком.

– Є можливість відображення інформації про договори з сервісними організаціями та заявки користувачів, що полегшує взаємодію зі зовнішніми партнерами та покращує обслуговування користувачів.

					ДП.КН. 23.515.22 000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		11

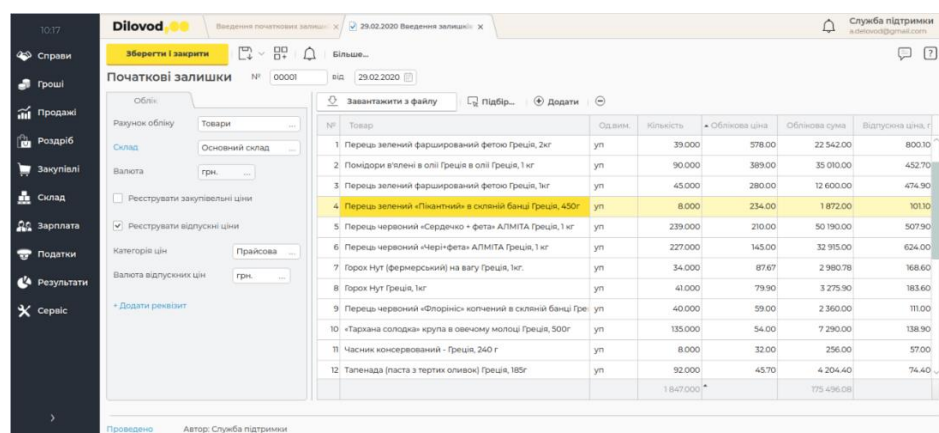
Недоліки програми PrintStore:

–Вартість програми може бути відносно високою для менших підприємств з обмеженим бюджетом.

–В програмі можуть бути деякі недоліки щодо інтерфейсу та функціональності, що можуть бути вирішені через оновлення та патчі.

–Не всі функції можуть бути корисними для кожного підприємства, що може призвести до переплати за зайву функціональність.

Остання програма, яку ми розглянемо зображена на рисунку.1.3.



№	Товар	Од. вим.	Кількість	Облікова ціна	Облікова сума	Відпускна ціна, г
1	Перець зелений фарширований фетого Греція, 2кг	уп	39,000	578.00	22 542.00	800.10
2	Помідори в'ялені в олії Греція в олії Греція, 1 кг	уп	90,000	389.00	35 010.00	452.70
3	Перець зелений фарширований фетого Греція, 1кг	уп	45,000	280.00	12 600.00	474.90
4	Перець зелений «Пікантний» в скляній банці Греція, 450г	уп	8,000	234.00	1 872.00	101.10
5	Перець червоний «Серденко» фета» АЛМІТА Греція, 1 кг	уп	239,000	210.00	50 190.00	507.90
6	Перець червоний «Мері-фета» АЛМІТА Греція, 1 кг	уп	227,000	145.00	32 915.00	624.00
7	Горох Нут (фериерський) на вагу Греція, 1кг	уп	34,000	87.67	2 980.78	168.60
8	Горох Нут Греція, 1кг	уп	41,000	79.90	3 275.90	183.60
9	Перець червоний «Флорініс» копчений в скляній банці Гре	уп	40,000	59.00	2 360.00	111.00
10	«Тарзана солодка» крупа в овсяному молоті Греція, 500г	уп	135,000	54.00	7 290.00	138.90
11	Часник консервованій - Греція, 240 г	уп	8,000	32.00	256.00	57.00
12	Таленада (паста з тертих оливков) Греція, 185г	уп	92,000	45.70	4 204.40	74.40
			1 847,000		775 496.08	

Рисунок 1.3 – Інтерфейс програми «віддалений склад»

Назва розробленої програми - "Віддалений склад", створена компанією "Перфект Софт".

Сервіс надає широкий спектр функціональних можливостей, що включають ведення довідників підприємства, таких як співробітники, відділи, посади, робочі приміщення, склади, контрагенти та обладнання.

Додатково, він дозволяє здійснювати облік ремонтів обладнання та руху витратних матеріалів, включаючи прихід, витрату, перезаправлення та переміщення по складах.

Сервіс також надає можливість вести облік договорів з сервісними організаціями та відображати оперативні залишки матеріалів.

					ДП.КН. 23.515.22 000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		12

Крім того, сервіс забезпечує розрахунок швидкості витрати картриджів та оцінку часу, на який вистачить запасу. Він також має можливість штрих-кодуювання.

Наявний облік принтерів та картриджів у сервісі дозволяє швидко формувати замовлення на відсутні витратні матеріали на основі статистики витрат.

Облік перезаправок картриджів сприяє ефективному управлінню зберіганням історії всіх дій з картриджами та принтерами. До того ж, сервіс надає можливість створення значної кількості звітів для зручного аналізу даних.

Створений веб-додаток призначений для використання на віддалених складах та інтеграції з УТ Управління торгівлею.

Цей додаток надає можливість отримувати доступ до інформаційної бази через Інтернет та забезпечує роботу складської служби шляхом оприбуткування, відвантаження, списання та інвентаризації товарів, включаючи основні функції.

Переваги програми Віддалений склад:

–Програма дозволяє ефективно вести облік товарів на складі, а також контролювати їх рух між складами.

–Забезпечує можливість управління запасами, тобто планування та прогнозування потреб в товарах, що знижує витрати на запаси.

–Програма дає можливість підключити багато складів до системи, що дозволяє централізовано контролювати рух товарів та запаси на всіх складах.

–Забезпечує можливість контролювати терміни придатності товарів та попереджати про наближення дати закінчення терміну придатності, що дозволяє уникнути продажу непридатних товарів.

–Має простий та зрозумілий інтерфейс, що дозволяє легко засвоїти програму.

					ДП.КН. 23.515.22 000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		13

Недоліки програми Віддалений склад:

–Вартість програми може бути досить високою для менших підприємств з обмеженим бюджетом.

–Не дозволяє вести облік фінансових операцій, таких як розрахунки з постачальниками та клієнтами.

–Може виникнути проблема зі синхронізацією даних, якщо на складах використовуються різні версії програми.

–Не забезпечує можливості автоматичного поповнення запасів, що може вимагати додаткової роботи від персоналу.

Враховуючи переваги та недоліки програми Віддалений склад, перед вибором потрібно ретельно розглянути свої потреби та бюджет.

1.3 Обґрунтування доцільності створення системи та постановки завдання.

У сучасному світі, де технології займають важливе місце у житті людей, особливо важливо створювати нові інформаційні ресурси та програмні засоби, що допомагатимуть людям працювати швидше та ефективніше.

Сьогодні автоматизована робота виявляється більш ефективною, ніж ручна, тому що вона дозволяє швидко та легко виконувати функції.

Для проєктування автоматизованої системи введення торгівлі необхідно розробити та впровадити інформаційну систему обліку матеріальних цінностей, яка повинна відповідати наступним ключовим вимогам:

–Забезпечувати зручне та зрозуміле середовище для проведення обліку.

–Надавати можливість доступу до файлів на сервері та їх редагування.

–Забезпечувати можливість збереження, редагування та видалення даних.

					ДП.КН. 23.515.22 000 ПЗ	Арк.
						14
Зм.	Арк.	№ докум.	Підп.	Дата		

- Дозволяти підключатися до існуючих облікових записів.
- Забезпечувати можливість перегляду бази даних товарів з інших складів.

Після аналізування існуючих рішень та їх реалізації можна зробити висновок, що розробка симулятора з обмеженою функціональністю, зручним інтерфейсом та максимально простим буде доцільною.

Це пояснюється тим, що симулятор зазвичай використовується для тестування невеликого обладнання, тому його інтерфейс повинен бути легким та доступним для користувачів.

					ДП.КН. 23.515.22 000 ПЗ	Арк.
						15
Зм.	Арк.	№ докум.	Підп.	Дата		

2ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Формалізація вимог

Під час визначення завдання проєкту було встановлено необхідність розробки інформаційної системи для автоматизації додавання продукту, видалення продукту, редагування та пошук.

На рисунку 2.1 показані бізнес-процеси, які передбачається автоматизувати.

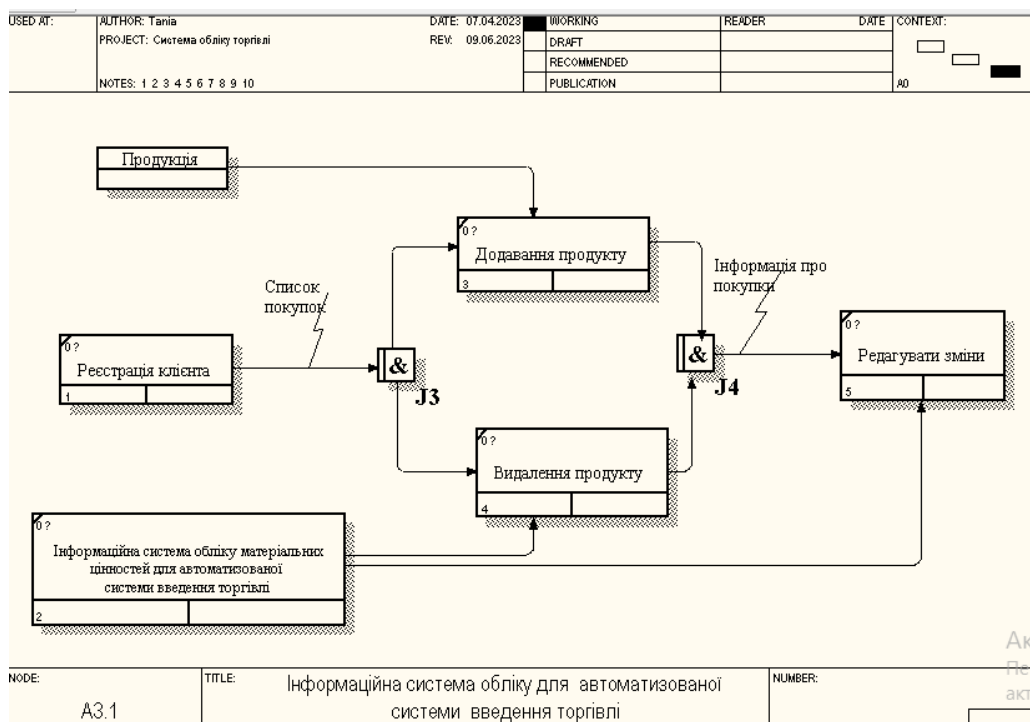


Рисунок 2.1 – Відображає діаграму бізнес-процесу

Ці бізнес-процеси відображають основні етапи та дії, які відбуваються в рамках інформаційної системи обліку матеріальних цінностей.

Рисунок 2.2 демонструє основні складові узагальненої умовно замкненої моделі.



Рисунок 2.2 – Демонструє узагальнену умовно замкнену модель інформаційної системи

На схемі (рисунок. 2.2) показані інформаційні управляючі зв'язки, які описують процес онлайн-покупки:

- 1i - представляє внесення персональних даних клієнта,
- 2i - звернення клієнта до пошуку магазину з метою придбання товарів,
- 3i – додавання наявного товару,
- 4i - прибуття товарів до магазину ,
- 5i - надання магазином доданих товарів,
- 6i - якщо продукція є в наявності в магазині, то працівник магазину не звертається до складу;
- 7i – керівник перевіряє наявний товар.

2.2 Побудова дерева функцій та проєктування структури системи

На рисунку 2.3 зображено дерево функцій системи, яке відображає послідовність функцій та під функцій, що відповідають за роботу інформаційної системи обліку матеріальних цінностей в автоматизованій системі введення торгівлі.

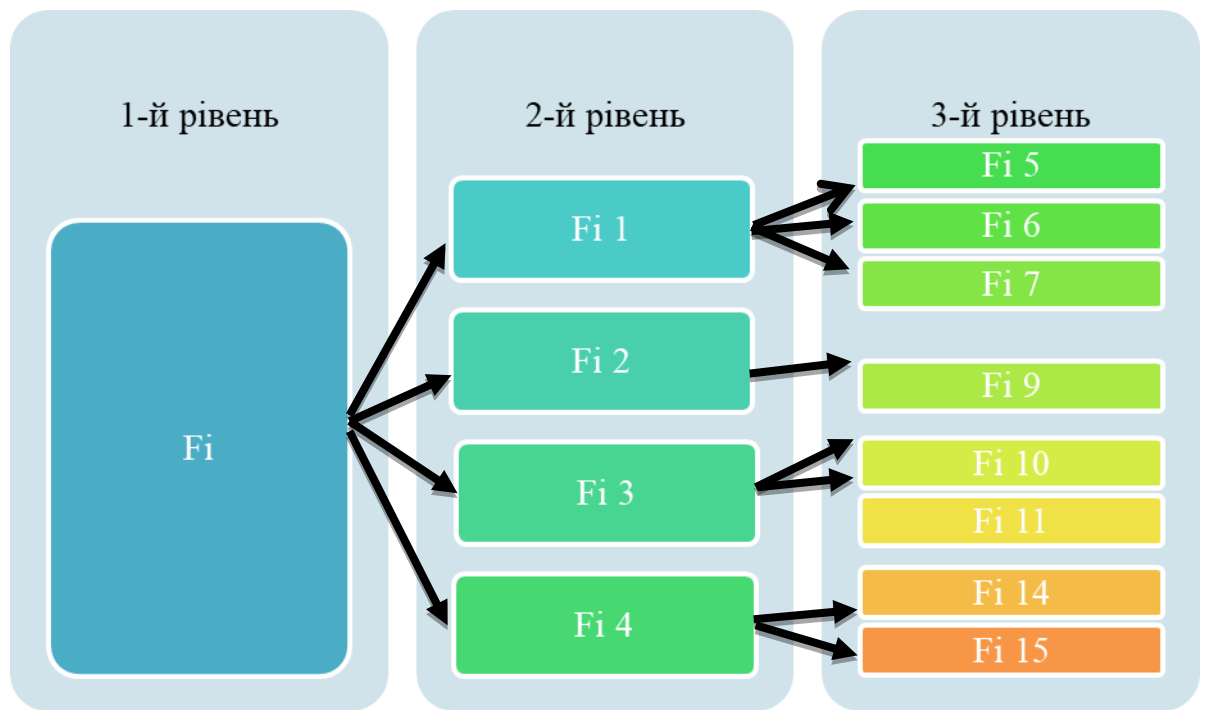


Рисунок 2.3 – Дерево функцій

Fi1 - відстеження руху товарів:

Fi5 - реєстрація надходжень,

Fi6 - моніторинг наявності товарів,

Fi7 - фіксація продажів.

Fi2 - збір інформації про доходи:

Fi9 - підрахунок доходів за певний період.

Fi3 - управління клієнтами:

Fi10 - реєстрація нових клієнтів,

Fi11 - стеження за постійними клієнтами,

Fi4 - моніторинг товарообігу:

Fi14 - контроль наявності товарів на складі,

Fi15 - фіксація продажів.

Головне завдання магазину полягає у забезпеченні оперативного оновлення інформації про товари. Крім цього, необхідно проводити аналіз числових показників для ефективного управління магазином.

2.3 Проєктування бази даних

Після проведення аналізу предметної області було визначено набір інформаційних сутностей: «Клієнт», «Керівник», «Товари», «Магазин», «Рядок замовлення», «Працівники». Для подальшого проєктування було проведено детальний аналіз предметної області з метою встановлення зв'язків між цими сутностями та виконано наступні кроки процесу проєктування .

Аналізуючи взаємозв'язок між сутностями "Керівник" і "Магазин", було встановлено, що вони зв'язані типом "один до багатьох". Хоча цей зв'язок не має великого значення в рамках поточної задачі, оскільки система буде використовуватись лише для одного магазину, в майбутньому він може бути корисним для розширення функціональності та додавання нових магазинів. Тому, відокремлення "Керівника" у окрему сутність значно підвищить масштабованість проєкту в цілому [1].

В магазині необхідно мати працівника. Відповідна сутність дозволяє додавати декількох працівників, якщо в магазині працює більше одного. Тому між сутностями "Магазин" та "Працівник" було встановлено зв'язок такого ж типу - "один до багатьох".

Головна мета проєкту - автоматизація системи введення торгівлі, включаючи облік матеріальних цінностей. З цією метою важливо виділити сутність "Наявні товари". Існування цієї сутності означає, що товари знаходяться в магазині. Тому, було встановлено аналогічний до попереднього зв'язок між сутностями "Наявні товари" та "Магазин".

Між сутностями "Клієнт" та "Наявні товари" було встановлено зв'язок "багато до багатьох", який реалізується за допомогою проміжної сутності "Рядок замовлення".

Для реалізації цього зв'язку було використано неідентифікуючі зв'язки, щоб система мала можливість переглядати покупки конкретних клієнтів та кількість придбань конкретного товару.

					ДП.КН. 23.515.22 000 ПЗ	Арк.
						19
Зм.	Арк.	№ докум.	Підп.	Дата		

На рисунку 2.4 відображено концептуальну модель для розроблюваного проєкту.

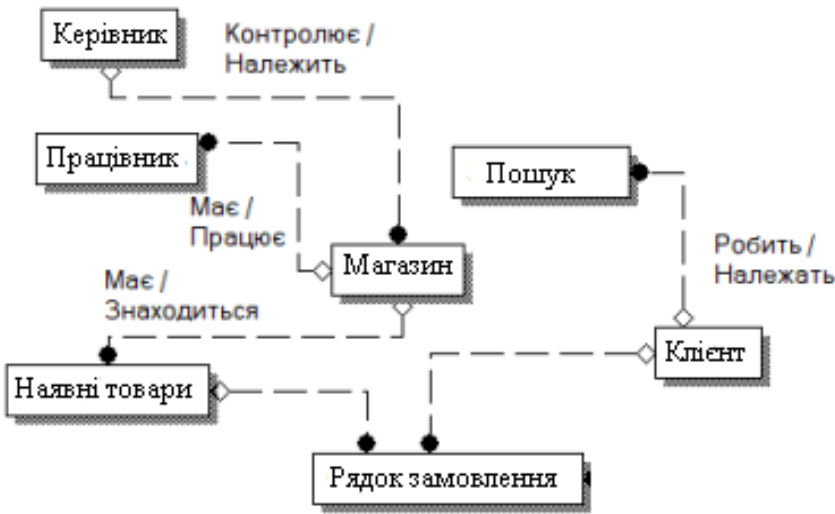


Рисунок 2.4 – відображення концептуальної моделі БД

Після визначення сутностей було необхідно визначити ключі для кожної з них, щоб забезпечити їхню унікальність. Кожна сутність отримала числовий ідентифікатор, який не повторюється і не пов'язаний з іншими сутностями.

Для реалізації зв'язків між сутностями, як показано на рисунку 2.5, було визначено первинні ключі та здійснено їх міграцію.

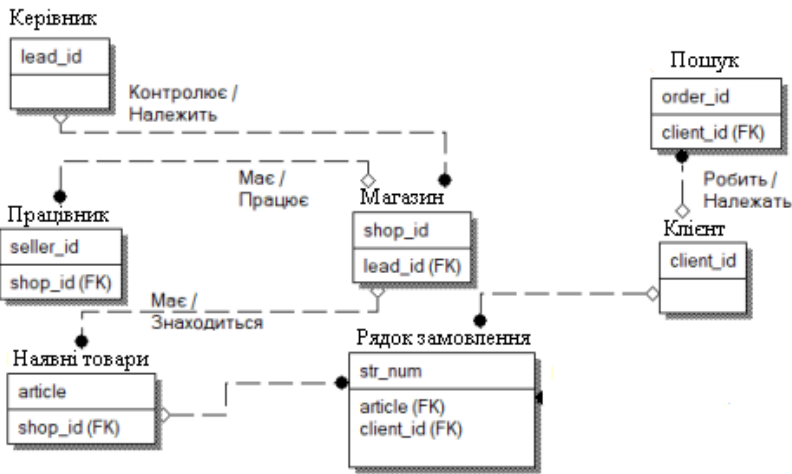


Рисунок 2.5 – Перенесення ключів

Після завершення перенесення ключів було створено набори атрибутів для кожної сутності, що дозволять зберігати всі необхідні дані для ведення обліку матеріальних цінностей. Кожний набір атрибутів наведено у таблиці 2.1.

Таблиця 2.1 –Атрибути для кожної сутності

Складний інформаційний об'єкт	Атрибути інформаційного об'єкту	Характеристики об'єкту
Керівник	lead_id lead_name lead_phone lead_email	Дано про керівника магазину (ім'я, телефон, ел. Адреса).
Клієнт	client_id client_name client_reg	Дано про клієнта (ім'я, інформація про реєстрацію).
Магазин	shop_id shop_address sold_sum bought_sum rent_coast	Дані про магазин (адреса, вартість проданої та закупленої продукції, вартість оренди).

Продовження таблиці 2.1

Наявні товари	article gds_name gds_price gds_count	Дані про наявні товари (артикул, назва, ціна, кількість).
Рядок замовлення	str_num article client_id gcount	Дані про покупки (ідентифікатор рядка, артикул, клієнт, кількість певного товару).

Ці атрибути допомагають ідентифікувати та описувати керівника, клієнта та магазин в контексті інформаційної системи обліку матеріальних цінностей для автоматизованої системи введення торгівлі.

Після перенесення ключів та визначення основних наборів атрибутів було проведено детальний аналіз кожної сутності та її відповідних атрибутів, що зображено в таблиці 2.2. , який було проведено після перенесення ключів та визначення основних наборів атрибутів.

Таблиця 2.2 – Докладний аналіз сутностей та їх атрибутів.

Інформаційний об'єкт сутності	Атрибут сутності	Ідентифікатор сутності	Ключі
Керівник	lead_id	lead_id	Р-ключ
	lead_name	lead_name	-
	lead_phone	lead_phone	-
	lead_email	lead_email	-

Продовження таблиці 2.2

Працівник	seller_id	seller_id	Р-ключ
	seller_name	seller_name	-
	seller_phone	seller_phone	-
	shop_id	shop_id	Ф-ключ
	seller_email	seller_email	-
Клієнт	Client_id	Client_id	Р-ключ
	Client_name	Client_name	-
	Client_reg	Client_reg	-
Магазин	Shop_id	Shop_id	Р-ключ
	shop_address	Shop_address	-
	sold_sum	Sold_sum	-
	bought_sum	bought_sum	-
	lead_id	lead_id	Ф-ключ
	rent_coast	rent_coast	-
Наявні товари	article	article	Р-ключ
	shop_id	shop_id	Ф-ключ
	gds_name	gds_name	-
	gds_price	gds_price	-
	gds_count	gds_count	-
Рядок замовлення	str_num	str_num	Р-ключ
	article	article	Ф-ключ
	client_id	client_id	Ф-ключ
	gcount	gcount	-
	rep_id	rep_id	Ф-ключ

Атрибути допомагають ідентифікувати та описувати керівника, клієнта, працівника, в контексті інформаційної системи обліку матеріальних цінностей для автоматизованої системи введення торгівлі.

Ці атрибути допомагають ідентифікувати та описувати керівника, клієнта, працівника, магазин, наявні товари та рядок замовлення, в контексті інформаційної системи обліку матеріальних цінностей для автоматизованої системи введення торгівлі.

Кожен з цих суб'єктів має свої унікальні характеристики та атрибути, які є важливими для їх ідентифікації та відображення в системі.

Загальна мета цих атрибутів полягає в забезпеченні надійного та точного ідентифікування, опису та відображення кожного з атрибутів в інформаційній системі обліку матеріальних цінностей (таблиця 2.3).

Таблиця 2.3 – Проаналізовано та описано зв'язки між сутностями

Зв'язки між сутностями	Назва зв'язку	Тип зв'язку	Зміст зв'язку
Керівник-Магазин	Контролює/Належить	1:Б	Керівник може контролювати декілька магазинів.
Магазин- Працівник	Має/Працює	1:Б	В одному магазині може працювати кілька працівників, покупець може мати кілька замовлень.
Магазин-Наявні товари	Має/Знаходиться	1:Б	Багато звернень можуть бути прийняті одним працівником
Клієнт-Наявні товари	Купує/Належать	Б:Б	Багато клієнтів можуть купити багато товарів.

Зв'язки між сутностями у програмі інформаційної системи обліку матеріальних цінностей грають важливу роль у забезпеченні інтегрованого та зручного функціонування системи.

Ці зв'язки визначають взаємозв'язок та взаємозалежність між різними сутностями, що сприяє ефективному управлінню, обліку та аналізу матеріальних цінностей.

Правильно встановлені зв'язки між сутностями створюють основу для успішної роботи інформаційної системи обліку матеріальних цінностей для автоматизованої системи введення торгівлі.

Після цього етапу проектування наступним кроком є створення логічної моделі реляційної бази даних.

У таблиця 2.4 описується певний атрибут сутності. Серед атрибутів було виділено ім'я, телефон, електрона пошта.

Таблиця 2.4 – Атрибути сутності lead

Атрибути сутності	Обмеження атрибутів					Ключ
	Тип даних	Межі або допустимі значення	Структура (формат)	Умова	Значення за замовчуванням	
_id	Лічильник	-	-	-	-	Р
_name	Текстовий	-	-	-	-	-
_phone	Текстовий	-	-	-	-	-
_email	Текстовий	-	-	-	-	-

У таблиці 2.5 описується атрибут сутності client. Серед атрибутів було виділено ім'я, інформація про реєстрацію, адресу та інше.

Таблиця 2.5 – Структура таблиці client

Атрибути сутності	Обмеження атрибутів					Ключ
	Тип даних	Межі або допустимі значення	Структура (формат)	Умова	Значення за замовчуванням	
_id	Лічильник	-	-	-	-	P
_name	Текстовий	-	-	-	-	-
_email	Текстовий	-	-	-	-	-

У таблиці 2.6 описується атрибут сутності seller. Серед атрибутів було виділено ім'я, телефон, пошту.

Таблиця 2.6 – Структура таблиці seller

Атрибути сутності	Обмеження атрибутів					Ключ
	Тип даних	Межі або допустимі значення	Структура (формат)	Умова	Значення за замовчуванням	
_id	Лічильник	-	-	-	-	P
_shop	Числовий	-	-	-	-	F

Продовження таблиці 2.6

_name	Текстовий	-	-	-	-	-
_phone	Текстовий	-	-	-	-	-
_email	Текстовий	-	-	-	-	-

У таблиці 2.7 описується атрибуту сутності shop. Серед атрибутів було виділено ім'я, адресу та інше.

Таблиця 2.7 – Структура таблиці shop

Атрибути сутності	Обмеження атрибутів					Ключ
	Тип даних	Межі або допустимі значення	Структура (формат)	Умова	Значення за замовчуванням	
_id	Лічильник	-	-	-	-	P
_lead	Числовий	-	-	-	-	F
_adreess	Текстовий	-	-	-	-	-
_sold	Грошовий	-	-	-	-	-
_bought	Грошовий	-	-	-	-	-
_rent	Грошовий	-	-	-	-	-

У таблиці 2.8 описується атрибут сутності goods. . Серед атрибутів було виділено ім'я, артикул, ціна, магазин, кількість.

Таблиця 2.8 – Структура таблиці goods

Атрибути сутності	Обмеження атрибутів					Ключ
	Тип даних	Межі або допустимі значення	Структура (формат)	Умова	Значення за замовчуванням	
_article	Текстовий	-	-	-	-	P
_shop	Числовий	-	-	-	-	F
_name	Текстовий	-	-	-	-	-
_price	Грошовий	-	-	-	-	-
_count	Числовий	-	-	-	-	-

У таблиці 2.9 описується атрибут сутності order_str. . Серед атрибутів було виділено дата, клієнт, артикул.

Таблиця 2.9 – Структура таблиці order_str

Атрибути сутності	Обмеження атрибутів					Ключ
	Тип даних	Межі або допустимі значення	Структура (формат)	Умова	Значення за замовчуванням	
_id	Лічильник	-	-	-	-	P
_article	Числовий	-	-	-	-	F

Продовження таблиці 2.9

_client	Текстовий	-	-	-	-	F
_count	Числовий	-	-	-	-	-
_date	Дата і час	-	-	-	-	-

У таблицях до кожної сутності був описаний певний його атрибут.

2.4 Проєктування користувацького інтерфейсу

Таким чином, потрібно розбити функціонал на кілька ролей, як зображено на рисунку 2.6.

Обов'язки керівника включають:

- інформації про працівників(додавання, редагування та видалення) ;
- інформації про магазини (додавання, редагування та видалення);
- інформації про товари (додавання, редагування та видалення).

Обов'язки працівника:

- операції з товарами(додавання, редагування та видалення).

Обов'язки зареєстрованої особи:

- створення замовлень;
- перегляд історії своїх покупок.

Обов'язки неавторизованої особи:

- реєстрація;
- авторизація.

Для забезпечення роботи цього функціоналу було створено низку форм, які включають наступні елементи: "Керівник", "Працівник", "Клієнт", "Товар", "Магазин", "Покупки", "Реєстрація" та "Авторизація"[2].

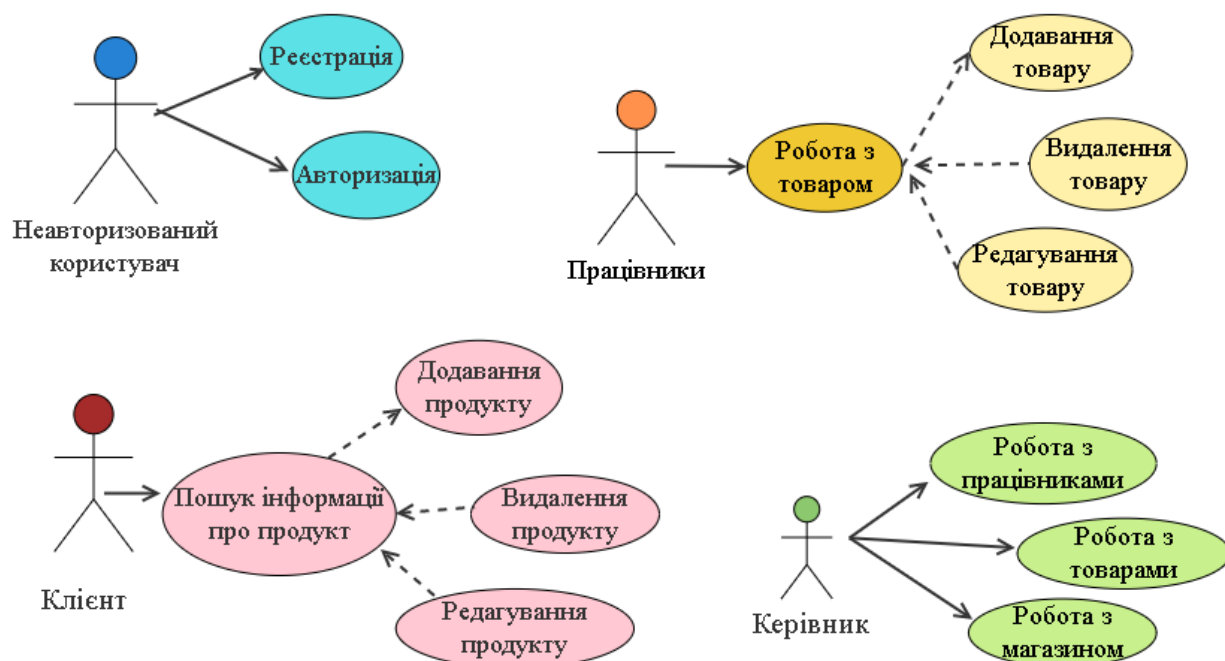


Рисунок 2.6 –Діаграма прецедентів

Форма "Керівник" має забезпечувати функції додавання, видалення та оновлення інформації про керівників, зокрема їхнє ім'я, телефон та електронну пошту.

Форми, пов'язані з сутностями «Продавець», «Клієнт» та «Магазин», мають бути забезпечені однаковим набором функцій для редагування інформації про відповідні об'єкти.

Форма "Товар" повинна дозволяти додавати, редагувати та видаляти окремі екземпляри товарів, а також групу елементів, що відрізняється від попередніх форм.

Форма "Покупки" має забезпечувати можливість внесення інформації про покупки, але не повинна надавати можливість перегляду, редагування або видалення.

Форма «Реєстрація» дозволяє користувачам зареєструватися в системі, тоді як форма «Авторизація» надає можливість входу до системи для вже зареєстрованих користувачів.

3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Опис технологій та засобів реалізації

Для виконання даного дипломного проєкту було вибрано мову програмування C# та розробниче середовище Microsoft Visual Studio.

Visual Studio, розроблений компанією Microsoft, є комплексним інструментарієм для розробки програмного забезпечення.

Це інтегроване середовище розробки (IDE), яке надає розробникам широкий набір інструментів і можливостей для ефективної роботи над різноманітними проєктами [3].

Будучи одним з найпопулярніших інструментів серед розробників, Visual Studio підтримує різні мови програмування, включаючи C#, Visual Basic, C++, F# та інші, що надає гнучкість при реалізації різноманітних програмних рішень. Завдяки своєму багатофункціональному характеру, Visual Studio сприяє підвищенню продуктивності розробників, забезпечує ефективну роботу з кодом, відлагодження програм, тестування та розгортання програмного забезпечення.

Це середовище розробки, яке дозволяє розробникам реалізувати свої ідеї та проєкти на високому рівні як з точки зору функціональності, так і з точки зору якості коду та надійності програми.

C# (C-Sharp) є однією з мов програмування, що належить до сімейства мов .NET Framework. Вона знайшла широке застосування у розробці різноманітних типів додатків, таких як веб-додатки, мобільні додатки, додатки для платформи Windows, хмарні додатки та багато інших [4].

Для розробки на C# доступні різноманітні технології та інструменти, які сприяють зручній розробці, налагодженню та впровадженню програмного забезпечення. Ось кілька прикладів таких технологій та інструментів: Microsoft Visual Studio, Visual Studio Code, .NET Framework, .NET Core, ASP.NET, Entity Framework [5].

					ДП.КН. 23.515.22 000 ПЗ	Арк.
						31
Зм.	Арк.	№ докум.	Підп.	Дата		

Ці технології та засоби реалізації C# надають розробникам потужні інструменти для створення різноманітних програм.

C# має свої переваги, а саме:

- Стиль програмування: C# є сучасною, об'єктно-орієнтованою мовою програмування з синтаксисом, який є зрозумілим і легким для вивчення. Вона надає розробникам потужні інструменти для створення складних програм з керованим кодом і великою кількістю бібліотек.

- Крос-платформеність: За допомогою .NET Core розробники можуть створювати крос-платформені додатки на C#, що означає, що один і той же код можна використовувати на різних операційних системах, таких як Windows, macOS і Linux.

- Велика спільнота: C# має велику спільноту розробників, що означає наявність великої кількості ресурсів, документації, форумів і бібліотек, які сприяють швидкому вирішенню проблем та підтримці розробників.

- Інтеграція з .NET Framework: C# є основною мовою для розробки на платформі .NET Framework, яка надає широкі можливості для створення різноманітних додатків, включаючи настільні, веб- та мобільні додатки.

- Масштабованість: C# добре підходить для розробки великих і складних проєктів. Вона підтримує модульну архітектуру, яка дозволяє розробникам розділити програму на незалежні компоненти для більш зручного керування і розширення [6].

У процесі розробки дипломного проєкту була використана MySQL, безкоштовна система управління реляційними базами даних.

MySQL - це одна з найпопулярніших в світі відкритих реляційних систем керування базами даних (RDBMS), яка розробляється та підтримується компанією Oracle Corporation. Вона використовується широко для ефективного зберігання та управління даними у різних видів програмних додатків.

					ДП.КН. 23.515.22 000 ПЗ	Арк.
						32
Зм.	Арк.	№ докум.	Підп.	Дата		

MySQL надає розробникам широкий спектр функцій та можливостей, які допомагають створювати ефективні та надійні бази даних. Вона підтримує стандартні операції реляційних баз даних, такі як створення таблиць, вставка, оновлення та видалення даних. Крім того, вона також надає розширені можливості, такі як транзакції для забезпечення консистентності даних, індекси для покращення швидкодії запитів, обмеження цілісності для забезпечення правильності даних, а також механізми оптимізації запитів для ефективного виконання складних запитів [7].

Однією з особливостей MySQL є його широка підтримка для різних платформ, включаючи Windows, Linux та macOS. Це дає розробникам можливість використовувати MySQL на різних серверах та операційних системах залежно від їх потреб.

3.2 Реалізація основних функцій інформаційної системи.

Для втілення необхідного функціоналу була створена база даних, яка відповідає фізичній структурі, зображеній на рисунку 3.1.

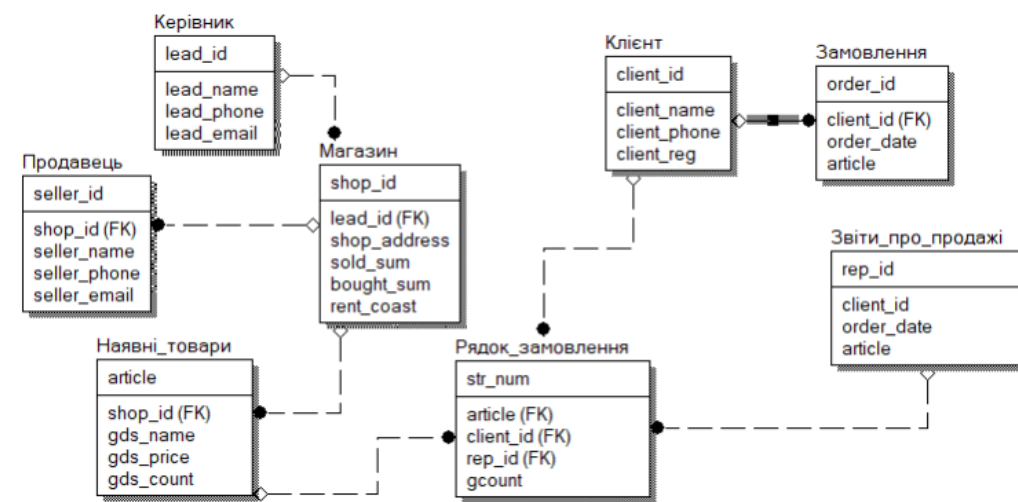


Рисунок 3.1 – Реалізація структури БД

Згідно з фізичною структурою, було створено запити для створення таблиць (див. лістинг А1 у додатку А), які включають унікальні поля, індекси

та ключі (див. лістинг A2 у додатку A). Також були розроблені зберезувальні процедури, які дозволяють додавати, змінювати та видаляти дані, виконуючи додаткові маніпуляції з вхідними параметрами.

Для об'єкта "магазин" було створено 3 зберезувальні процедури (див. лістинг A3 у додатку A):

- create_shop (створення магазину);
- update_shop (оновлення магазину);
- delete_shop (видалення магазину).

Процедура create_shop отримує 5 параметрів: ідентифікатор керівника, адресу магазину, загальну вартість проданих і закуплених товарів на момент додавання запису про конкретний магазин та вартість оренди. Далі проводиться перевірка, чи є вже магазин з такою самою адресою. Якщо збіг не знайдено, запис буде доданий, в іншому випадку запит не буде виконаний.

Процедура update_shop отримує такі ж параметри, як і create_shop, а також додатково отримує ідентифікатор запису, який потрібно змінити. Після цього проводиться перевірка можливості внесення змін, шляхом порівняння значень атрибутів наявних записів з переданими значеннями для оновлення. Якщо збігів не знайдено, запис з ідентифікатором, переданим в процедуру оновлення, буде змінений.

Процедура delete_shop отримує 1 параметр - ідентифікатор запису. Оскільки вхідний параметр є цілочисельним значенням, не потрібно екранування. Додана перевірка наявності запису з переданим ідентифікатором перед виконанням запиту на видалення, що дозволяє розширити функціонал в майбутньому.

Було розроблено 3 зберезувальні процедури (див. лістинг A4 у додатку A) для сутності "Продавець". Ці процедури включають:

- create_seller (створення продавця);
- update_seller (оновлення інформації про продавця);
- delete_seller (видалення продавця).

					ДП.КН. 23.515.22 000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		34

Збережувальна процедура `create_seller` отримує 4 параметри: ідентифікатор магазину, до якого призначений певний продавець, ім'я продавця, номер телефону та електронна пошта. Після цього виконується перевірка наявності записів, в яких хоча б одне поле співпадає з полями нового запису. Якщо знайдено збіг, запис про нового продавця не буде доданий. У протилежному випадку буде створений новий запис з інформацією про продавця.

Збережувальна процедура `update_seller` приймає 5 параметрів: ідентифікатор запису, який потрібно змінити, ідентифікатор магазину, за яким закріплений продавець, оновлене ім'я продавця та його контакти: номер телефону та електронна пошта. Якщо модифікація наявного запису не спричинить появи збігів полів ім'я, електронна пошта, номер телефону, то запис буде оновлено, в іншому випадку оновлення не відбудеться.

Процедура `delete_seller` отримує 1 параметр - ідентифікатор запису про продавця, який необхідно видалити. Після цього вона виконує запит на видалення з використанням цього параметра.

Було реалізовано 3 збережувальні процедури (див. лістинг A5 у додатку A) для сутності "Клієнт". Ці процедури включають:

- `create_client` (створення клієнта);
- `update_client` (оновлення інформації про клієнта);
- `delete_client` (видалення клієнта).

Збережувальна процедура `create_client` приймає 3 параметри: ім'я клієнта, номер телефону та електронну адресу. Перед вставкою запису в базу даних, виконується перевірка наявності збігів в значеннях полів з існуючими записами. Параметри процедури екрануються для захисту від недоречного використання та передаються у прототип запиту. Запис буде додано лише у випадку, якщо не виявлено збігів з існуючими записами.

Збережувальна процедура `update_client` приймає 4 параметри: ідентифікатор клієнта, ім'я, телефон та електронну пошту. Після цього виконується перевірка наявності збігів між значеннями полів існуючих

					ДП.КН. 23.515.22 000 ПЗ	Арк.
						35
Зм.	Арк.	№ докум.	Підп.	Дата		

записів та переданими параметрами. Аналогічно до попередньої процедури, параметри екрануються перед виконанням запиту. Якщо не виявлено збігів, процедура виконує запит на оновлення інформації про клієнта.

Збережувальна процедура `delete_client` приймає один параметр - ідентифікатор клієнта. Після передачі цього параметра, виконується перевірка наявності запису про клієнта з вказаним ідентифікатором. Якщо в базі даних знайдено запис, відповідний до заданого ідентифікатора, то виконується запит на його видалення.

Було розроблено 3 збережувальні процедури для сутності "Товар":

- `create_goods`;
- `update_goods`;
- `delete_goods`.

У процедурі `create_goods`, яка використовується для створення товару, передаються 5 параметрів: артикул товару, ідентифікатор магазину, назва, ціна та кількість. Після отримання цих параметрів процедура перевіряє, чи існують записи з артикулом нового товару. Якщо такий запис знайдено, то кількість товарів з параметрів процедури додається до наявної кількості відповідного поля. У випадку, якщо збіг не знайдено, процедура створює новий запис в таблиці.

У процедурі `update_goods`, яка використовується для оновлення товару, передаються 4 параметри: артикул товару, назва, ціна та кількість. Ці параметри екрануються, а потім виконується підготовлений запит на оновлення даних. Це дозволяє змінювати значення полів, таких як назва, ціна та кількість, для відповідного товару.

У процедурі `delete_goods`, яка використовується для видалення товару, передається 1 параметр - артикул товару. Ця процедура виконує запит на видалення запису з таблиці, використовуючи переданий артикул товару.

Для забезпечення реєстрації та авторизації користувачів у системі була створена таблиця з назвою "users". Для безпечної взаємодії з цією таблицею,

					ДП.КН. 23.515.22 000 ПЗ	Арк.
						36
Зм.	Арк.	№ докум.	Підп.	Дата		

щоб уникнути потенційних атак SQL-ін'єкцій, було реалізовано три збережувальні процедури (див. лістинг А6 у додатку А):

- create_user;
- change_user_pass;
- delete_user.

У збережувальній процедурі create_user передаються 4 вхідні параметри: ім'я користувача, номер телефону, електронна пошта та пароль. Після цього виконується перевірка наявності запису зі значеннями, що співпадають зі значеннями нового запису. Якщо збігів не знайдено, то після екранування параметрів новий запис додається до таблиці.

У збережувальній процедурі change_user_pass передаються 2 параметри: електронна пошта та новий пароль. Після перевірки коректності даних процедура оновлює запис з відповідною електронною поштою, змінюючи значення пароля.

У збережувальній процедурі delete_user передається 1 вхідний параметр - ідентифікатор користувача. Ця процедура виконує запит на видалення запису з таблиці, використовуючи переданий ідентифікатор користувача.

Для реєстрації покупок користувачів були розроблені збережувальні процедури, які відповідають за введення та зміну даних у таблиці "order_str" (див. лістинг А7 у додатку А):

- create_order_str;
- update_order_str;
- delete_order_str.

Збережувальна процедура create_order_str отримує 5 параметрів: артикул товару, ідентифікатор клієнта, ціну за одиницю товару з урахуванням знижки, кількість товарів кожного виду та дату покупки. Після цього виконується запит на додавання запису без додаткових перевірок,

					ДП.КН. 23.515.22 000 ПЗ	Арк.
						37
Зм.	Арк.	№ докум.	Підп.	Дата		

оскільки всі змінні перетворені в конкретні типи, які не дозволяють виконати SQL-ін'єкцію.

Збережувальна процедура `update_order_str` приймає 6 параметрів: ідентифікатор запису про певну одиницю товару, артикул товару, ідентифікатор клієнта, ціну товару, кількість та дату. Запит виконується без додаткових перевірок з причин аналогічних попередньому випадку.

Збережувальна процедура `delete_order_str` приймає 1 параметр - ідентифікатор запису і видаляє його.

Збережувальні процедури, які дозволяють додавати, змінювати та видаляти дані, виконуючи додаткові маніпуляції з вхідними параметрами, є важливою складовою інформаційної системи обліку матеріальних цінностей для автоматизованої системи введення торгівлі.

Ці процедури забезпечують гнучкість та функціональність системи, дозволяючи користувачам ефективно взаємодіяти з даними та забезпечувати актуальність та цілісність інформації. Ці процедури дозволяють користувачам системи додавати нові дані, змінювати вже існуючі записи та видаляти непотрібну інформацію.

3.3 Реалізація користувацького інтерфейсу системи.

Для встановлення обмежень доступу користувачів були реалізовані функції авторизації та реєстрації, які вимагають мінімальної особистої інформації від користувача. Процес виконується шляхом заповнення форми (див. лістинг Б1 у додатку Б), яка складається з набору полів (рисунок. 3.2).

Вікно реєстрації в програмному засобі містить наступні поля:

- Логін.
- Пароль.
- Ім'я.
- Прізвище.
- Дата народження.
- Поштова адреса.

					ДП.КН. 23.515.22 000 ПЗ	Арк.
						38
Зм.	Арк.	№ докум.	Підп.	Дата		

Registers

Login

Password

Name

LastName

Birthday

Email

Зареєструватися

Рисунок 3.2 – Форма реєстрації

Після того, як користувач зареєструвався, йому необхідно ввести свій логін і пароль у форму авторизації (див. лістинг Б2 у додатку Б) згідно з тими, які були вказані при реєстрації (рисунок. 3.3). Після цього виконується запит до бази даних з метою перевірки наявності відповідного облікового запису.

MainWindow

Login

Password

Register Sign in

Рисунок 3.3 – Форма авторизації

Після успішної авторизації користувач отримує доступ до функціоналу системи, який включає роботу з керівниками, магазинами, продавцями, клієнтами, товарами та підсумковими даними.

Для роботи з керівниками доступний віджет, який містить 4 вкладки (див. лістинг Б3 у додатку Б): "Додати", "Редагувати", "Видалити" і "Пошук". Кожна вкладка надає форму для виконання певних дій з базою даних.

Вкладка "Додати" дозволяє додавати новий товар. Для цього працівник повинен заповнити форму (див. лістинг Б4 у додатку Б), яка зображена на рисунку 3.4 і надіслати дані, натиснувши кнопку "Додати запис".

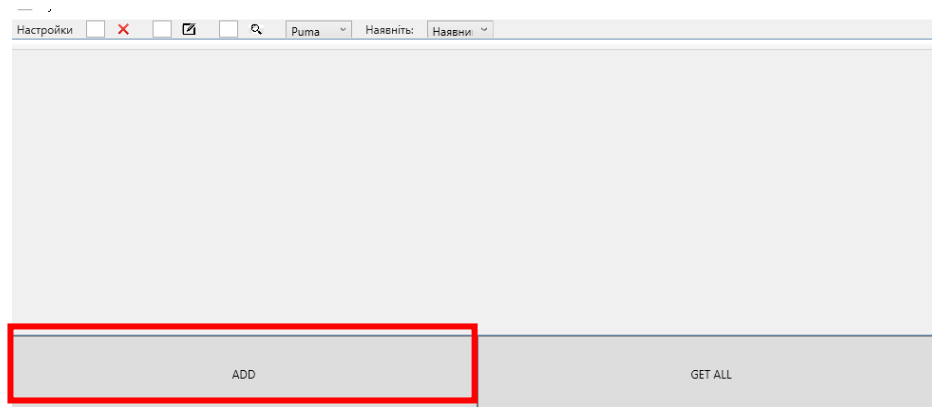


Рисунок 3.4 – Кнопка додавання товару

Натиснувши на кнопку "Додати товар", нам відкриється форма з полями для додавання певного товару (рисунку.3.5).

A form titled 'DodElements' with a close button. It contains several input fields: 'Color' and 'Model' (text inputs), 'Enter a price' and 'Size' (text inputs), 'Enter Shop' (a dropdown menu with 'Puma' selected) and 'State' (a dropdown menu with 'Навантаж.' selected). There is also a date picker labeled 'Select a date' with a calendar icon. At the bottom is a button labeled 'ADD ELEMENT'.

Рисунок 3.5 – Форма додавання

					ДП.КН. 23.515.22 000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		40

Після натискання кнопки "Додати запис", інформація буде передана до скрипту (див. лістинг Б5 у додатку Б), який опрацьовує значення, введені у поля форми, та зберігає їх у базі даних. При натиску кнопки "Отримати все", воно оновить та добавить новий товар (рисунок. 3.6)

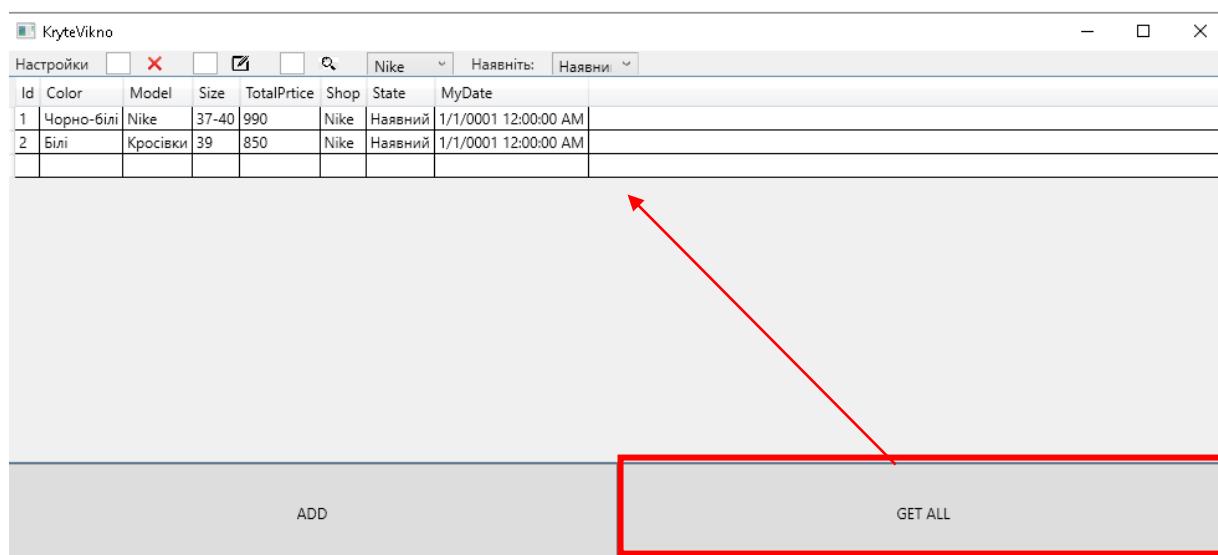


Рисунок 3.6 – Кнопка "Отримати все"

Після появи добавленого товару, його можна " Редагувати" та "Видалити".

Кнопки "Редагувати" та "Видалити" розміщені у верхньому куті секції "Настройки".

Для активації кнопки "Редагувати" (див. лістинг Б6 у додатку Б), необхідно ввести відповідний номер товару у відповідну клітинку (рисунок3.7).

Для тестування функціональності розробленого програмного засобу було внесено тестові записи. Під час тестування було виявлено помилки валідації, котрі були усунуті на подальших етапах проєктування.

Model	Size	TotalPrice
Nike	37-40	990
Кросівки	39	850

Color:

Model:

Enter a price:

Size:

Enter Shop:

State:

Рисунок 3.7 – Форма та клітинка для вводу редагування

Після внесення змін у дані та натискання кнопки "Додати товар", спрацьовує асинхронний обробник події надсилання форми (див. лістинг Б7 у додатку Б), який передає значення полів до скрипту (див. лістинг Б8 у додатку Б). Скрипт обробляє ці значення і викликає відповідну збережувальну процедуру.

Для того, щоб видалити товар, необхідно вказати номер товару та натиснути кнопку "Видалити"(рисунок.3.8). Після цього спрацьовує обробник відповідної події (див. лістинг Б9 у додатку Б).

Color	Model	Size	TotalPrice	Shop	State
Чорно-білі	Nike	37-40	990	Nike	На

Рисунок 3.8 – Кнопка "Видалити"

В магазині є випадаючий список, в якому можна обрати бренд товару та перейти до повного переліку товарів цього бренду (рисунок.3.9).

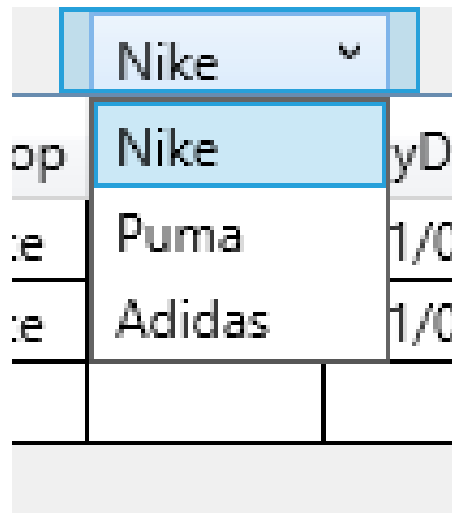


Рисунок 3.9 – Випадаючий список бренду

Також є випадаючий список, де можна переглянути товар який є в наявності, який прибуде та котрий вже відбув (рисунок.3.10).

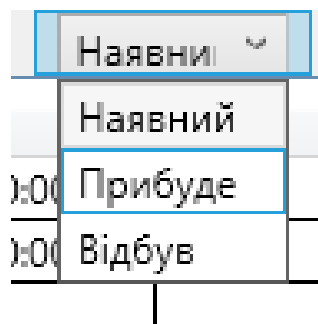


Рисунок 3.10 – Випадаючий список наявності товару

Розроблений користувацький інтерфейс інформаційної системи є результатом досліджень, аналізу вимог користувачів та використання сучасних технологій.

Він забезпечує зручну, ефективну та інтуїтивно зрозумілу роботу з системою, сприяючи поліпшенню продуктивності та задоволення користувачів.

3.4 Тестування інформаційної системи обліку матеріальних цінностей для автоматизованої системи введення торгівлі.

Для перевірки правильності функціонування процедури збереження "create_lead" було розроблено низку запитів, які наведені в лістингу В1, що додається до звіту. Для підтвердження правильності роботи цієї процедури було виконано виклик з необхідними параметрами, як показано на рисунку 3.11.

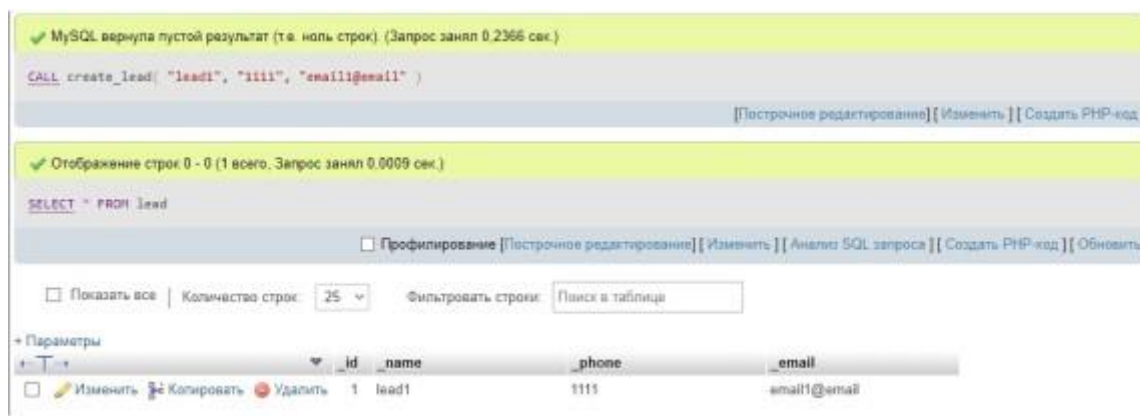


Рисунок 3.11 – Виконання процедури збереження "create_lead".

Ця процедура провела перевірку та додала новий запис до системи. Після цього була протестована процедура "update_lead" для оновлення існуючого запису, як показано на рисунку 3.12.

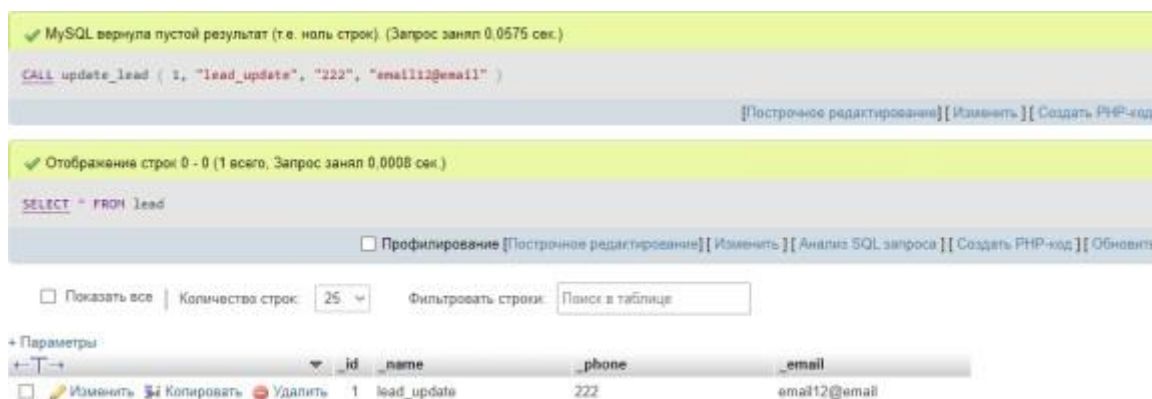


Рисунок 3.12 – Виконання процедури "update_lead"

Це призвело до успішного оновлення запису. Після цього була викликана процедура для додавання запису про магазин, в якій використовувався ідентифікатор існуючого керівника, як показано на рисунку 3.13.

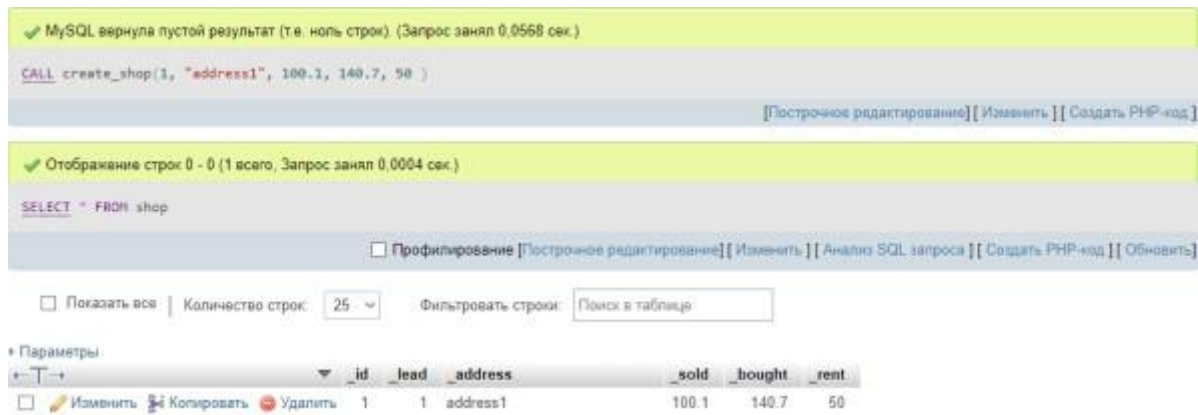


Рисунок 3.13 – Процедура збереження "create_shop".

Після успішного створення нового запису про магазин, була викликана відповідна процедура для оновлення цього запису, передаючи параметри, які вказані на рисунку 3.14.

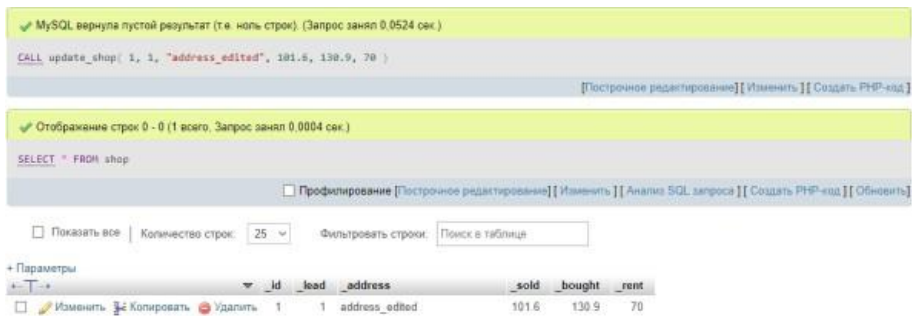


Рисунок 3.14 – Процедура "update_shop"

Після того, як було внесені зміни до запису про магазин, було додано новий запис про продавця з використанням ідентифікатора існуючого магазину та параметрів, які наведені на рисунку 3.16.

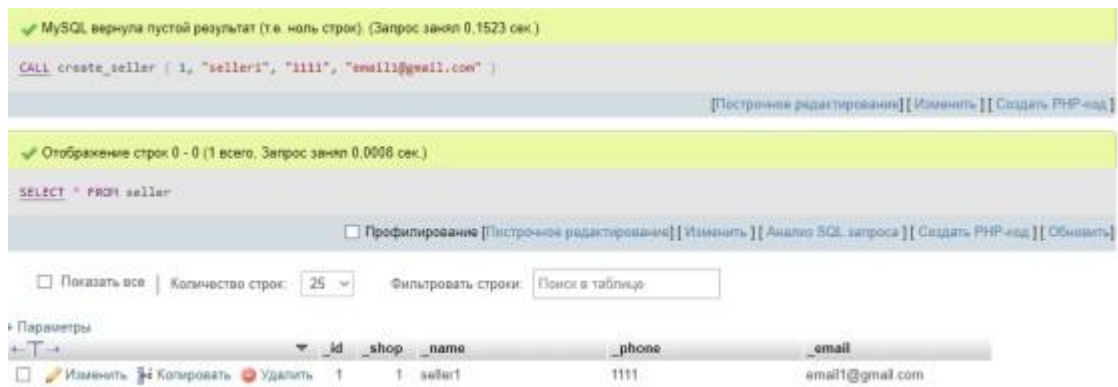


Рисунок 3.15 – Процедура "create_seller"

Після успішного створення запису про продавця, відбулося оновлення цього запису з використанням процедури "update_seller" та параметрів, які наведені на рисунку 3.16.

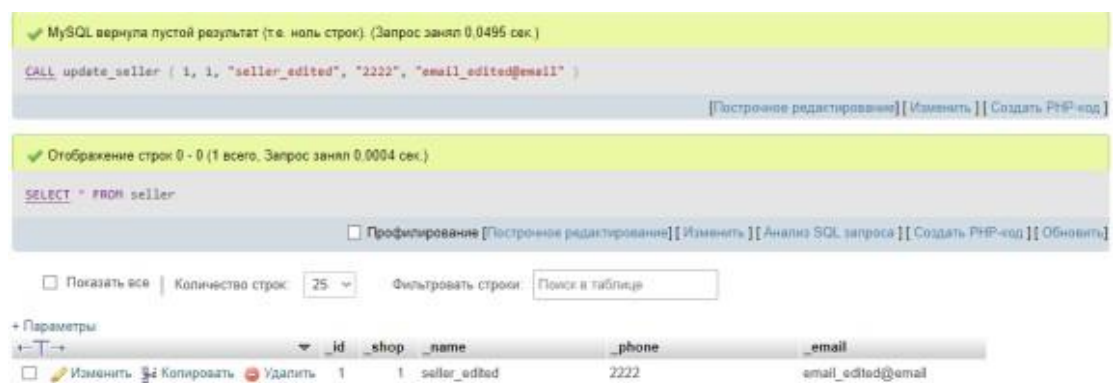


Рисунок 3.16 – Процедура " update_seller "

Після того, як було оновлено запис про продавця, було проведено тестування процедури "create_client" з використанням параметрів, які вказані на рисунку 3.17.

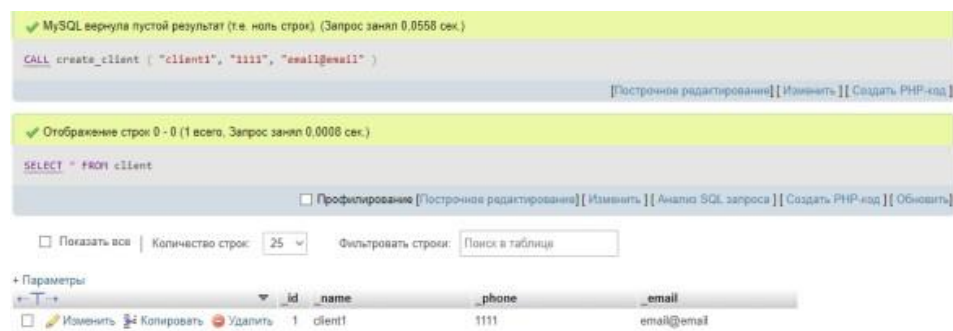


Рисунок 3.17 – Процедура " create_client "

Було проведено тестування процедури "update_client" з використанням параметрів, які наведені на рисунку 3.18, після успішного створення запису про клієнта.

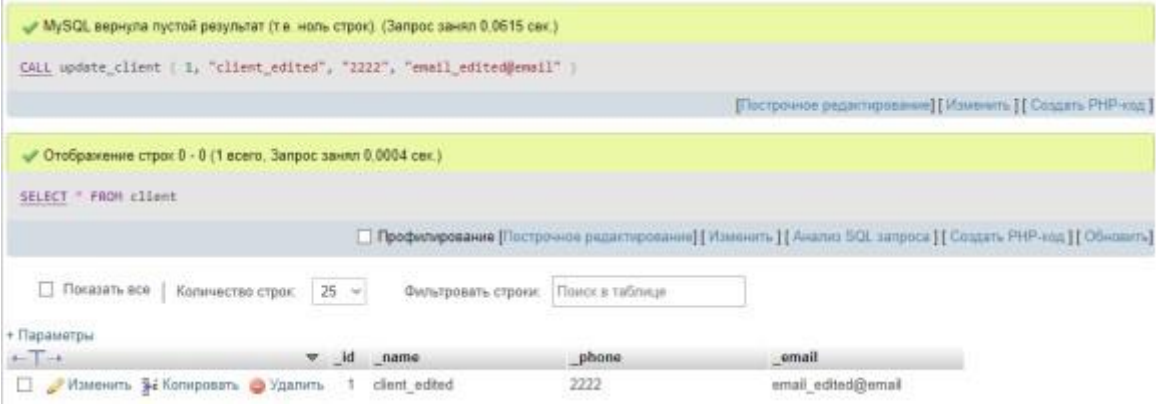


Рисунок 3.18 – Процедура " update_client "

Після цього була проведена перевірка правильності роботи процедури "create_goods" шляхом передачі параметрів, які вказані на рисунку 3.19.

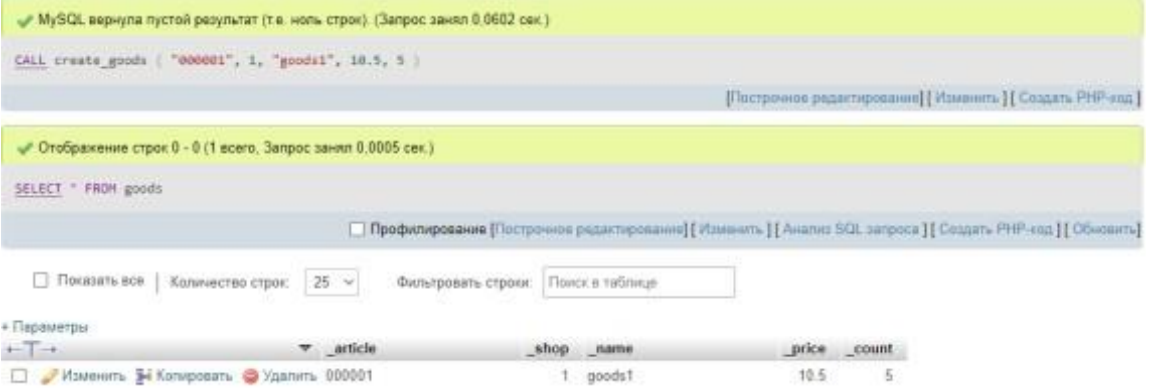


Рисунок 3.19 – Процедура " create_goods "

Було проведено тестування процедури "update_goods" з використанням параметрів, необхідних для оновлення запису, які наведені на рисунку 3.20, після успішного додавання запису про товар.

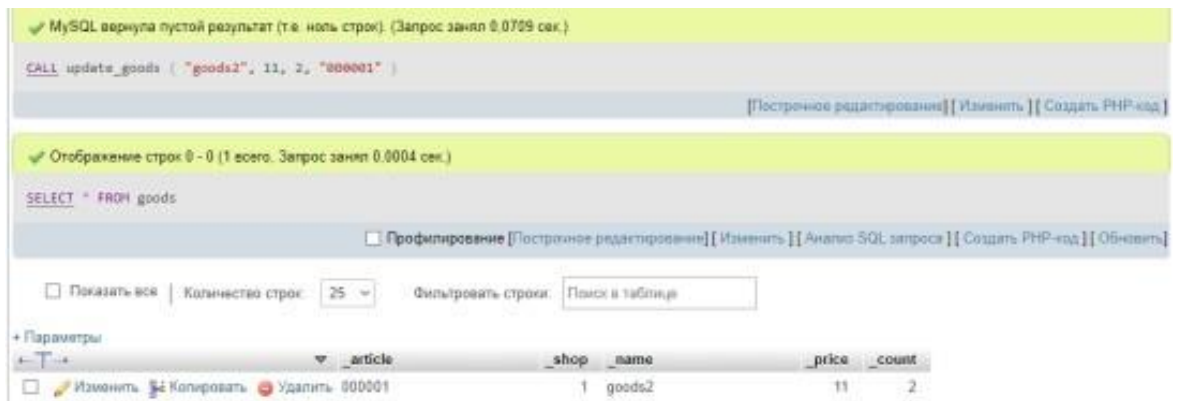


Рисунок 3.20 – Процедура " update_goods "

Після цього було проведено тестування процедури "create_user" з використанням параметрів, які представлені на рисунку 3.21.

Під час тестування було перевірено, чи правильно працює процедура "create_user" при введенні різних комбінацій параметрів. Це включало виконання таких дій, як введення вірних даних користувача, перевірка наявності унікального імені користувача, встановлення відповідної ролі користувача та інші дії, необхідні для успішного створення облікового запису.

Тестування процедури "create_user" з використанням параметрів, які представлені на рисунку 3.21, допомагає підтвердити правильність її роботи та забезпечити високу якість функціонування програми в цілому.

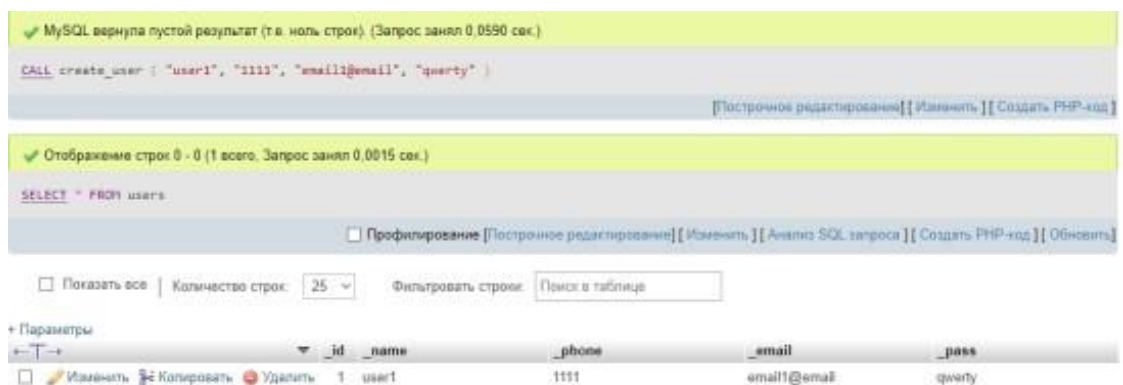


Рисунок 3.21 – Процедура " create_user "

Важливо зазначити, що при реєстрації користувача через відповідну форму його пароль буде збережено в базі даних у захешованому вигляді, який не може бути переглянутий або розшифрований адміністратором бази даних.

Тестування інформаційної системи є важливим етапом в її реалізації. Цей процес включає в себе перевірку функціональності, надійності, продуктивності та безпеки системи.

Основною метою тестування є виявлення помилок, неполадок та недоліків в роботі системи перед її впровадженням і використанням реальними користувачами.

Під час тестування не виявлено жодних проблем, що могли б поставити під загрозу правильну роботу інформаційної системи.

					ДП.КН. 23.515.22 000 ПЗ	Арк.
						49
Зм.	Арк.	№ докум.	Підп.	Дата		

4 ТЕХНІКО-ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ

4.1 Аналіз ринку

Аналіз ринку збуту продукту "Інформаційна система обліку матеріальних цінностей для автоматизованої системи введення торгівлі" показує, що попит на цей продукт зростає, оскільки все більше підприємств переходять на автоматизовану систему введення торгівлі.

Основними конкурентними перевагами продукту є:

- Висока ефективність та точність обліку матеріальних цінностей;
- Автоматизація процесу введення торгівлі, що знижує час на опрацювання документів та зменшує ризик помилок;
- Відповідність всім стандартам та вимогам щодо обліку матеріальних цінностей та систем введення торгівлі.

Однак, на ринку присутні конкурентні компанії, які пропонують аналогічні продукти. Щоб успішно конкурувати на ринку, необхідно забезпечити:

- Високу якість продукту та після продажний сервіс;
- Конкурентоспроможні ціни;
- Швидку та якісну технічну підтримку та консультації.

Для реклами продукту можна використовувати інтернет-маркетингові інструменти, такі як рекламні кампанії у Google та соціальних мережах, а також просування на спеціалізованих сайтах. Для просування продукту можна також використовувати рекомендації клієнтів, які вже використовують цей продукт.

Інформаційні системи обліку матеріальних цінностей для автоматизованої системи введення торгівлі вже існують на ринку, тому можна вважати, що вона є модифікацією вже існуючих систем. Проте, конкретні техніко-економічні та експлуатаційні характеристики нового виробу можуть бути відмінними від існуючих систем на ринку, що може забезпечити певні конкурентні переваги. Таким чином, виріб може мати як

					ДП.КН. 23.515.22 000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		50

елементи новизни, так і використовувати технології, які вже використовуються на ринку, проте має власні технічні та функціональні особливості.

Потенційними замовниками (покупцями) виробу "Інформаційної системи обліку матеріальних цінностей для автоматизованої системи введення торгівлі" можуть бути підприємства, що займаються оптовою або роздрібною торгівлею, складською логістикою, виробництвом та інші суб'єкти господарювання, які мають потребу в точному та ефективному обліку матеріальних цінностей та товарів. Також, ці системи можуть бути корисними для підприємств, що займаються логістикою, доставкою, ремонтом і технічним обслуговуванням техніки, які також мають потребу в ефективному контролі за матеріальними цінностями. Залежно від потреб та специфіки діяльності, потенційними замовниками можуть бути як невеликі підприємства, так і великі корпорації.

Інформаційна система обліку матеріальних цінностей для автоматизованої системи введення торгівлі, буде реалізовуватись на ринку послуг з автоматизації торгівлі та управління матеріальними цінностями. Також потенційними ринками можуть бути ринки програмного забезпечення та ІТ-рішень для бізнесу. Залежно від функціональних можливостей і особливостей виробу, можуть бути визначені конкретні сегменти ринку, такі як малі та середні підприємства роздрібною торгівлі, оптові торгові компанії, складські комплекси, тощо.

Очікується попит на Інформаційну систему обліку матеріальних цінностей для автоматизованої системи введення торгівлі з боку підприємств, що займаються торгівлею та складським господарством. У зв'язку з розвитком інформаційних технологій та автоматизації бізнес-процесів, попит на таку систему може зростати. Крім того, у зв'язку зі зростанням конкуренції в бізнесі, підприємства будуть шукати способи для підвищення ефективності своєї діяльності, що може стати додатковим стимулом для попиту на дану систему.

					ДП.КН. 23.515.22 000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		51

В залежності від специфіки виробу та ринку, на якому буде здійснюватись його реалізація, можуть застосовуватись різні методи продажу. Одним з можливих методів є продаж через спеціалізовані торгові точки або магазини, де покупці можуть ознайомитись з виробом та його функціональними характеристиками, а також отримати професійну консультацію. Для залучення більш широкої аудиторії можна використовувати Інтернет-магазини та інші онлайн-платформи.

Також можуть бути застосовані різноманітні маркетингові акції та промо-акції, наприклад, знижки на перші покупки або програми лояльності для постійних клієнтів. Окрім того, можна вести співпрацю зі спеціалізованими компаніями, які вже мають своїх клієнтів та готові пропонувати нові вироби своїм партнерам. Важливо також враховувати особливості ринку та конкуренції, щоб вибрати оптимальний метод продажу для даного виробу.

На ринку аналогічної продукції конкурентами Інформаційної системи обліку матеріальних цінностей для автоматизованої системи введення торгівлі можуть бути такі компанії:

- 1) SAP - має в своєму арсеналі рішення для управління логістикою, виробництвом, фінансами та іншими галузями, включаючи системи управління матеріальними цінностями.
- 2) Oracle - компанія надає послуги з управління матеріальними цінностями, зокрема, в рамках своєї системи управління ланцюгом постачання.
- 3) Microsoft - пропонує бізнес-рішення для управління ланцюгом постачання та іншими процесами, які можуть включати управління матеріальними цінностями.
- 4) IBM - має в своєму портфоліо рішення для управління матеріальними цінностями, включаючи системи управління запасами та логістикою.

					ДП.КН. 23.515.22 000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		52

5) Infor - компанія спеціалізується на рішеннях для управління бізнес-процесами, включаючи управління матеріальними цінностями.

6) Epicor - надає комплексні рішення для управління виробництвом, логістикою та іншими процесами, що можуть включати управління матеріальними цінностями.

7) Odoo - компанія надає відкриті рішення для управління бізнес-процесами, включаючи управління матеріальними цінностями.

Ці конкуренти мають великий досвід в розробці програмного забезпечення для управління бізнес-процесами та засобів автоматизації, що може становити конкурентну перевагу в порівнянні з новим виробом.

Організація сервісного обслуговування є важливою складовою виробничого процесу і може значно впливати на задоволення клієнтів та підвищення їх лояльності до компанії. У разі Інформаційної системи обліку матеріальних цінностей для автоматизованої системи введення торгівлі, рекомендовано передбачити до продажне та після продажне обслуговування наступним чином:

До продажне обслуговування:

- Розробити інформаційний пакет для клієнтів, що містить детальну інформацію про можливості системи та її переваги над конкурентами;
- Провести навчання клієнтів щодо користування системою та надати консультації стосовно вибору оптимального пакету послуг;
- Забезпечити можливість співпраці з іншими постачальниками, які можуть доповнити або покращити функціонал системи, що пропонується [8].

Після продажне обслуговування:

- Надати підтримку користувачам, що полягає в розв'язанні технічних проблем та підтримці при використанні системи;
- Забезпечити можливість оновлення та покращення функціоналу системи;

					ДП.КН. 23.515.22 000 ПЗ	Арк.
						53
Зм.	Арк.	№ докум.	Підп.	Дата		

– Розробити систему звітності, що дозволить збирати та аналізувати дані про проблеми та питання, які виникають у користувачів, та приймати відповідні заходи щодо їх вирішення.

Організація до продажного та після продажного обслуговування може допомогти збільшити продажі та підвищити лояльність клієнтів до компанії .

4.2 Розрахунок витрат на проєктування

Розрахуємо витрати на розробку проєкту "Інформаційна система обліку матеріальних цінностей для автоматизованої системи введення торгівлі" за трудовим методом і оформимо кошторис витрат за формою таблиці 4.1.

Таблиця 4.1 – Кошторис витрат на проєктування

Найменування статей витрат	Сума, грн	Обґрунтування
1 Зарплата проєктувальників.	120 000	Зарплата проєктувальників, зайнятих на проєкті протягом 6 місяців
2. Відрахування на соціальні потреби.	30 000	Відрахування на соціальне страхування та медичне обслуговування
3. Контрагентські роботи і послуги.	50 000	Оренда сервера та інше програмне забезпечення
4. Витрати на відрядження.	10 000	Витрати на відрядження проєктувальників для взаємодії з замовником
5. Інші прямі витрати.	20 000	Витрати на оренду приміщення та комунікації
6. Усього прямих витрат.	230 000	Сума витрат на прямі витрати

Продовження таблиці 4.1

7. Накладні витрати.	46 000	Відсоток від прямих витрат (20%) на оплату послуг керівника проєкту та бухгалтера
8. Планові накопичення.	34 500	Відсоток від прямих витрат (15%) на формування резервного фонду проєкту
9. Усього, кошторисна вартість проєкту.	310 500	Сума всіх витрат на проєкт, включаючи прямі, накладні витрати та планові накопичення
10. Податок на додану вартість.	62 100	20% від кошторисної вартості проєкту
11. Загалом, договірна ціна розробки Зп.	372 600	Сума кошторисної вартості проєкту з урахуванням податку на додану вартість

Зарплата проєктувальників.

Для розрахунку заробітної плати проєктувальників на тему "Інформаційна система обліку матеріальних цінностей для автоматизованої системи введення торгівлі" було виконано наступні кроки:

1) Створено таблицю з наступними стовпцями: "N" - порядковий номер, "Посада" - назва посади, "Оклад" - заробітна плата без урахування відрахувань, "Відрахування" - сума грошей, які заробітній платі працівника будуть утримані згідно з законодавством, "Кількість" - кількість працівників з даною посадою, "Зарплата" - розрахункова заробітна плата з урахуванням відрахувань.

2) Заповнено таблицю для трьох посад: "Головний інженер", "Програміст", "Тестувальник".

3) Обчислено відрахування для кожної посади, використовуючи вказану в таблиці суму відрахувань від окладу. Результати додано до стовпця "Відрахування".

					ДП.КН. 23.515.22 000 ПЗ	Арк.
						55
Зм.	Арк.	№ докум.	Підп.	Дата		

4) Обчислено заробітну плату для кожної посади, використовуючи формулу: зарплата = оклад - відрахування. Результати додано до стовпця "Зарплата".

5) Обчислено загальну зарплату за період проєкту (6 місяців) для кожної посади, перемноживши розрахункову зарплату на кількість працівників і на кількість місяців. Результати додано до відповідного стовпця.

6) Обчислено загальну зарплату для всіх посад, додавши розрахункову зарплату за період для кожної посади.

7) Додано рядок "Усього зарплата", де зазначено загальну зарплату за проєкт.

8) Результати було сформовано в таблиці, яка зображена в таблиці 4.2.

Таблиця 4.2 – Розрахунок заробітної плати проєктувальників

N	Посада	Оклад,	Відрахування	Кількість		Зарплата
1	Головний інженер	25 000	7500	1	6	150 000
2	Програміст	20 000	5000	2	6	240 000
3	Тестувальник	15 000	3000	1	6	90 000
	Усього зарплата:		15 500			480 000

4.3 Обґрунтування необхідності розробки

Розробка інформаційної системи обліку матеріальних цінностей для автоматизованої системи введення торгівлі є дуже важливою задачею для будь-якої компанії, яка займається торгівлею.

Основні переваги такої системи полягають у тому, що вона дозволяє підтримувати точний облік матеріальних цінностей, що зменшує ймовірність втрати або крадіжки. Крім того, система дозволяє знизити час на проведення інвентаризації, що також зменшує витрати компанії.

Додатковою перевагою такої системи є те, що вона може бути інтегрована з іншими системами управління, такими як системи управління запасами або системи управління проєктами, що дозволяє підвищити ефективність роботи всієї компанії в цілому.

Таким чином, розробка інформаційної системи обліку матеріальних цінностей для автоматизованої системи введення торгівлі є необхідною для підвищення ефективності і точності управління матеріальними цінностями компанії, зниження витрат і підвищення загальної продуктивності.

Пропонована інформаційна система обліку матеріальних цінностей для автоматизованої системи введення торгівлі повинна задовольнити наступні потреби замовників (покупців):

- Ефективність: Система повинна бути ефективною в роботі і забезпечувати швидкий та точний облік матеріальних цінностей, що дозволить покупцеві отримувати актуальну інформацію про наявність товарів, їх рух та стан запасів.

- Надійність: Система повинна бути надійною та безпечною, щоб уникнути втрати даних, що може призвести до серйозних проблем в бізнесі. Крім того, система повинна мати можливість зберігати інформацію про всі транзакції, що відбуваються в процесі торгівлі, що дозволить покупцеві моніторити діяльність своєї компанії та проводити аналізи продажів.

- Гнучкість: Система повинна бути гнучкою, щоб вона могла адаптуватися до змін у бізнес-потребах компанії, наприклад, зміна типу матеріальних цінностей, зміна методів обліку тощо.

- Легкість використання: Система повинна бути легкою в використанні та містити інтуїтивний інтерфейс, щоб знизити час на навчання персоналу та забезпечити швидкий доступ до необхідної інформації.

					ДП.КН. 23.515.22 000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		57

– Сумісність: Система повинна бути сумісною з іншими системами, що використовуються в компанії, такими як системи управління запасами, системи управління проєктами тощо, щоб забезпечити повний облік та інтеграцію даних між цими системами.

Загалом, інформаційна система обліку матеріальних цінностей для автоматизованої системи введення торгівлі повинна задовольнити потреби замовників у точному та швидкому обліку матеріальних цінностей, що забезпечить ефективне управління запасами та покращення бізнес-процесів.

Крім того, система повинна бути надійною та безпечною, щоб уникнути втрати даних та містити інтуїтивний інтерфейс, що знизить час на навчання персоналу та забезпечить швидкий доступ до необхідної інформації.

Важливим аспектом є гнучкість системи, щоб вона могла адаптуватися до змін у бізнес-потребах компанії та бути сумісною з іншими системами, що використовуються в компанії.

В цілому, інформаційна система обліку матеріальних цінностей є необхідним інструментом для покупців, який забезпечує ефективне та точне управління матеріальними цінностями, що дозволить покращити продуктивність та ефективність діяльності компанії.

Інформаційна система обліку матеріальних цінностей для автоматизованої системи введення торгівлі може позитивно вплинути на економічні показники компанії за допомогою покращення управління запасами, оптимізації процесів виробництва та зменшення затрат на управління матеріальними цінностями.

Перш за все, з використанням інформаційної системи обліку матеріальних цінностей буде забезпечено точний та своєчасний облік запасів. Це дозволить уникнути надмірного запасу товарів, який займає велику частину складу та може стати причиною зменшення ліквідності компанії. З іншого боку, нестача товарів на складі може призвести до зниження продуктивності та втрати замовлень.

					ДП.КН. 23.515.22 000 ПЗ	Арк.
						58
Зм.	Арк.	№ докум.	Підп.	Дата		

Інформаційна система обліку матеріальних цінностей дозволяє підтримувати оптимальний рівень запасів, що знизить затрати на складське господарство та підвищить ефективність управління запасами.

Отже, запровадження проєкту "Інформаційна система обліку матеріальних цінностей для автоматизованої системи введення торгівлі" може сприяти покращенню ефективності управління запасами, покращенню якості обліку матеріальних цінностей, оптимізації робочих процесів, підвищенню якості обслуговування клієнтів, зменшенню витрат на управління матеріальними цінностями, підвищенню безпеки та контролю над запасами, покращенню звітності та аналітики.

					ДП.КН. 23.515.22 000 ПЗ	Арк.
						59
Зм.	Арк.	№ докум.	Підп.	Дата		

ВИСНОВКИ

У даному дипломному проєкті була розроблена і впроваджена інформаційна система обліку матеріальних цінностей для автоматизованої системи введення торгівлі. Процес розробки включав аналіз вимог і потреб бізнесу, проєктування архітектури системи, розробку програмного забезпечення, тестування і впровадження.

Результати роботи над дипломним проєктом свідчать про успішну реалізацію інформаційної системи, яка демонструє високу функціональність і коректну роботу. Інформаційна система дозволяє ефективно вести облік матеріальних цінностей, що використовуються в процесі торгівлі, забезпечує точність та достовірність даних, а також забезпечує швидкий доступ до необхідної інформації.

Однією з головних переваг інформаційної системи є автоматизація багатьох рутинних процесів, що дозволяє зменшити ручну працю і підвищити продуктивність персоналу. Вона спрощує процес введення торгових операцій, дозволяючи швидко і точно реєструвати дані про продукти, їх рух та кількість. Автоматизація багатьох процесів сприяє зниженню людського фактора та помилок, що можуть виникнути при ручному обліку.

Програма включає в себе такі функціональні можливості, як реєстрація, додавання, видалення та редагування товару та забезпечення безпеки матеріальних цінностей. Крім того, система має інтерфейс, який дозволяє зручно та просто користуватися програмою, виконувати необхідні операції та отримувати актуальну інформацію.

Завдяки розробленій програмі інформаційної системи обліку матеріальних цінностей, автоматизована система введення торгівлі отримує потужний інструмент для керування та оптимізації процесів, забезпечуючи ефективну торгівельну діяльність. Розроблена програма дозволяє спростити

					ДП.КН. 23.515.22 000 ПЗ	Арк.
						60
Зм.	Арк.	№ докум.	Підп.	Дата		

процеси введення, обліку та контролю матеріальних цінностей, забезпечуючи швидкий доступ до актуальної інформації про товари, склад та рух товарів у магазині.

У подальшому розвитку проекту можливе розширення функціональності інформаційної системи, включаючи впровадження модулів аналізу продажів, прогнозування попиту, управління знижками та акціями, а також інтеграцію з іншими системами управління та обліку. Це дозволить забезпечити ще більш гнучкий та ефективний підхід до ведення торговельного бізнесу.

У підсумку, дипломний проєкт з розробки інформаційної системи обліку матеріальних цінностей для автоматизованої системи введення торгівлі є вагомим внеском у сферу торгівлі. Результати роботи свідчать про досягнення поставлених цілей, а сама система відповідає сучасним вимогам та стандартам. Цей проєкт має практичну цінність і може бути успішно використаний в реальних умовах для покращення ефективності та управління в сфері торгівлі.

					ДП.КН. 23.515.22 000 ПЗ	Арк.
						61
Зм.	Арк.	№ докум.	Підп.	Дата		

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Інформаційно-пошукові системи. *Облік матеріальних цінностей*: вебсайт.URL: <http://esu.com.ua/> (дата звернення 10.05.2023) .

2.MySQL. *Реалізаційна система базами даних*: вебсайт.URL: <https://uk.wikipedia.org/wiki/MySQL>(дата звернення : 16.05.2023).

3. Проєктування бази даних. *Інформаційні системи та технології*: вебсайт.URL: https://pidru4niki.com/11570718/bankivska_sprava/proektuvannya_baz_danih (дата звернення: 20.05.2023).

4. About SQLite. *Полегшена реалізаційна система керування БД*: вебсайт.URL: <https://www.sqlite.org/index.html> (дата звернення 21.05.2023) .

5. MySQL. *Вільна система керування реалізаційними базами даних*: вебсайт.URL: <https://www.mysql.com/> (дата звернення 21.05.2023) .

6. Пасічник В. В. Організація баз даних та знань / В. В. Пасічник, В. А. Резніченко. – К.: Видавнича група BHV, 2006. – 384 с..

7. Пономаренко В. С. Інформаційні системи в управлінні персоналом.. Навчальний посібник / В. С. Пономаренко, І. В. Журавльова, І. Л. Латишева. – Харків: Вид. ХНЕУ, 2008. – 336 с. (Укр. мов.) Електрон. аналог друк. вид.: режим доступу: <http://www.repository.hneu.edu.ua/> (дата звернення 02.06.2023).

8. Переваги с#. *Об'єктно-орієнтована мова програмування*: вебсайт.URL: <https://www.04141.com.ua/list/326508>(дата звернення : 03.06.2023).

					ДП.КН. 23.515.22 000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		62

ДОДАТКИ

Додаток А

Запити на створення об'єктів бази даних

Лістинг А1 – Запит на створення таблиць

```
DROP DATABASE IF EXISTS unice;
CREATE DATABASE unice;
USE unice;

CREATE TABLE client
(
  _id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
  _name VARCHAR(50) NOT NULL,
  _phone VARCHAR(10) NULL,
  _email VARCHAR(30) NULL )ENGINE=InnoDB;

CREATE TABLE lead
(
  _id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
  _name VARCHAR(50) NULL,
  _phone VARCHAR(10) NULL,
  _email VARCHAR(50) NULL )ENGINE=InnoDB;

CREATE TABLE shop
(
  _id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
  _lead INT NULL,
  _address VARCHAR(50) NOT NULL,
  _sold FLOAT NULL,
  _bought FLOAT NULL,
  _rent FLOAT NULL )ENGINE=InnoDB;

CREATE TABLE seller
(
  _id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
  _shop INT NOT NULL,
  _name VARCHAR(50) NOT NULL,
  _phone VARCHAR(10) NULL,
  _email VARCHAR(50) NULL )ENGINE=InnoDB;

CREATE TABLE goods
(
```

					ДП.КН. 23.515.22 000 ПЗ	Арк.
						63
Зм.	Арк.	№ докум.	Підп.	Дата		

```

    _article VARCHAR(6) NOT NULL PRIMARY KEY,
    _shop INT NULL,
    _name VARCHAR(50) NOT NULL,
    _price FLOAT NULL,
    _count INT NULL )ENGINE=InnoDB;

CREATE TABLE order_str
(
    _id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
    _article VARCHAR(6) NOT NULL,
    _client INT NULL,
    _price FLOAT NULL,
    _count INT NOT NULL,
Продовження лістингу A1
    _date DATE NULL )
ENGINE=InnoDB;

CREATE TABLE users
(
    _id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
    _name VARCHAR(50) NOT NULL,
    _phone VARCHAR(10) NULL,
    _email VARCHAR(30) NULL,
    _pass VARCHAR(32) NULL )ENGINE=InnoDB;

```

Лістинг А2 – Запит на створення ключів, індексів та унікальних полів

```

ALTER TABLE shop ADD UNIQUE (_id);
ALTER TABLE seller ADD UNIQUE (_id);
ALTER TABLE goods ADD UNIQUE (_article);
ALTER TABLE lead ADD UNIQUE (_id);
ALTER TABLE client ADD UNIQUE (_id);

ALTER TABLE shop ADD INDEX (_lead);
ALTER TABLE seller ADD INDEX (_shop);
ALTER TABLE goods ADD INDEX (_shop);
ALTER TABLE order_str ADD INDEX (_client);

ALTER TABLE shop
ADD FOREIGN KEY (_lead)
REFERENCES lead (_id)
ON DELETE CASCADE
ON UPDATE CASCADE ;

ALTER TABLE seller
ADD FOREIGN KEY (_shop)

```

					ДП.КН. 23.515.22 000 ПЗ	Арк.
						64
Зм.	Арк.	№ докум.	Підп.	Дата		

```

REFERENCES shop (_id)
ON DELETE CASCADE
ON UPDATE CASCADE ;

ALTER TABLE goods
ADD FOREIGN KEY (_shop)
REFERENCES shop (_id)
ON DELETE CASCADE
ON UPDATE CASCADE ;

ALTER TABLE order_str
ADD FOREIGN KEY (_client)
REFERENCES client(_id)
ON DELETE CASCADE
ON UPDATE CASCADE ;

```

Лістинг А3 – Запит на створення зберезувальних процедур shop

```

DELIMITER //
DROP PROCEDURE IF EXISTS create_shop//
CREATE PROCEDURE create_shop ( IN lead INT, IN address
VARCHAR(50), IN sold FLOAT, IN bought FLOAT, IN rent FLOAT )
BEGIN
    - IF NOT EXISTS(SELECT 1 FROM shop WHERE shop._address =
        address) THEN
SET @s = 'INSERT INTO shop (shop._lead, shop._address,
shop._sold, shop._bought, shop._rent) VALUES(?, ?, ?, ?, ?)';
    - PREPARE stmt FROM @s;
EXECUTE stmt USING lead, address, sold, bought, rent;
    - END IF;
END //

DELIMITER //
DROP PROCEDURE IF EXISTS update_shop//
CREATE PROCEDURE update_shop ( IN id INT, IN lead INT, IN
address VARCHAR(50), IN sold FLOAT, IN bought FLOAT, IN rent
FLOAT )
BEGIN
IF NOT EXISTS(SELECT 1 FROM shop WHERE shop._address =
address)AND
EXISTS(SELECT 1 FROM shop WHERE shop._id = id) THEN
SET @s = 'UPDATE shop SET _lead=?, _address=?, _sold=?,
_bought=?, _rent=? WHERE _id=?';
    PREPARE stmt FROM @s;
EXECUTE stmt USING lead, address, sold, bought, rent, id;

```

					ДП.КН. 23.515.22 000 ПЗ	Арк.
						65
Зм.	Арк.	№ докум.	Підп.	Дата		

```

END IF;
END //

DELIMITER //
DROP PROCEDURE IF EXISTS delete_shop//

CREATE PROCEDURE delete_shop (IN id INT)
BEGIN
    - IF EXISTS(SELECT 1 FROM shop WHERE shop._id = id) THEN
    - DELETE FROM shop WHERE shop._id = id;
    - END IF;
END //

```

Лістинг А4 – Запит на створення збережувальних процедур seller

```

DELIMITER //
DROP PROCEDURE IF EXISTS create Seller//
CREATE PROCEDURE create Seller ( IN shop INT, IN name
VARCHAR(50), IN phone VARCHAR(10), IN email VARCHAR(50) )
BEGIN
IF NOT EXISTS(SELECT 1 FROM seller WHERE seller._name = name)
AND
    - NOT EXISTS(SELECT 1 FROM seller WHERE seller._phone =
phone) AND
    - NOT EXISTS(SELECT 1 FROM seller WHERE seller._email =
email) THEN
SET @s = 'INSERT INTO seller (seller._shop, seller._name,
seller._phone, seller._email) VALUES(?, ?, ?, ?)';
    - PREPARE stmt FROM @s;
EXECUTE stmt USING shop, name, phone, email;
    - END IF;
END //
DELIMITER //
DROP PROCEDURE IF EXISTS update Seller//
CREATE PROCEDURE update Seller ( IN id INT, IN shop INT, IN name
VARCHAR(50), IN phone VARCHAR(10), IN email VARCHAR(50) )
BEGIN
IF NOT EXISTS(SELECT 1 FROM seller WHERE seller._shop = shop)
OR
    - NOT EXISTS(SELECT 1 FROM seller WHERE seller._name = name)
OR
    - NOT EXISTS(SELECT 1 FROM seller WHERE seller._phone =
phone) OR
    - NOT EXISTS(SELECT 1 FROM seller WHERE seller._email =
email) AND

```

					ДП.КН. 23.515.22 000 ПЗ	Арк.
						66
Зм.	Арк.	№ докум.	Підп.	Дата		

```

- EXISTS(SELECT 1 FROM seller WHERE seller._id = id) THEN
SET @s = 'UPDATE seller SET _shop=?, _name=?, _phone=?, _email=?
WHERE seller._id=?';
- PREPARE stmt FROM @s;
EXECUTE stmt USING shop, name, phone, email, id;
- END IF;
END //
DELIMITER //
DROP PROCEDURE IF EXISTS delete_seller//
CREATE PROCEDURE delete_seller (IN id INT)
BEGIN
- IF EXISTS(SELECT 1 FROM seller WHERE seller._id = id) THEN
- DELETE FROM seller WHERE seller._id = id;
- END IF;
END //

```

Лістинг А5 – Запит на створення зберезувальних процедур client

```

DELIMITER //
DROP PROCEDURE IF EXISTS create_client//
CREATE PROCEDURE create_client ( IN name VARCHAR(50), IN phone
VARCHAR(10), IN email VARCHAR(50) )
BEGIN
IF NOT EXISTS(SELECT 1 FROM client WHERE client._name = name)
AND
- NOT EXISTS(SELECT 1 FROM client WHERE client._phone =
phone) AND
- NOT EXISTS(SELECT 1 FROM client WHERE client._email =
email) THEN
SET @s = 'INSERT INTO client ( client._name, client._phone,
client._email) VALUES(?, ?, ?)';
- PREPARE stmt FROM @s;
EXECUTE stmt USING name, phone, email;
- END IF;
END //

DELIMITER //
DROP PROCEDURE IF EXISTS update_client//
CREATE PROCEDURE update_client ( IN id INT, IN name VARCHAR(50),
IN phone VARCHAR(10), IN email VARCHAR(50) )
BEGIN
IF NOT EXISTS(SELECT 1 FROM client WHERE client._name = name)
OR
- NOT EXISTS(SELECT 1 FROM client WHERE client._phone =
phone) OR

```

					ДП.КН. 23.515.22 000 ПЗ	Арк.
						67
Зм.	Арк.	№ докум.	Підп.	Дата		

```

- NOT EXISTS(SELECT 1 FROM client WHERE client._email =
email) AND
- EXISTS(SELECT 1 FROM client WHERE client._id = id) THEN SET
@s = 'UPDATE client SET _name=?, _phone=?, _email=? WHERE
client._id=?';
- PREPARE stmt FROM @s;
EXECUTE stmt USING name, phone, email, id;
- END IF;
END //
DELIMITER //
DROP PROCEDURE IF EXISTS delete_client//
CREATE PROCEDURE delete_client (IN id INT)
BEGIN
- IF EXISTS(SELECT 1 FROM client WHERE client._id = id) THEN
- DELETE FROM client WHERE client._id = id;
- END IF;
END //

```

Лістинг А6 – Запит на створення зберезувальних процедур users

```

DELIMITER //
DROP PROCEDURE IF EXISTS create_user//
CREATE PROCEDURE create_user ( IN name VARCHAR(50), IN phone
VARCHAR(10), IN email VARCHAR(50), IN pass VARCHAR(32) )
BEGIN
IF NOT EXISTS(SELECT 1 FROM users WHERE users._name = name) AND
- NOT EXISTS(SELECT 1 FROM users WHERE users._phone = phone)
AND
- NOT EXISTS(SELECT 1 FROM users WHERE users._email = email)
THEN
SET @s = 'INSERT INTO users (users._name, users._phone,
users._email, users._pass) VALUES(?, ?, ?, ?)';
- PREPARE stmt FROM @s;
EXECUTE stmt USING name, phone, email, pass;
- END IF;
END //

DELIMITER //
DROP PROCEDURE IF EXISTS change_user_pass//
CREATE PROCEDURE change_user_pass (IN email VARCHAR(50), IN pass
VARCHAR(32) )
BEGIN
- IF EXISTS(SELECT 1 FROM users WHERE users._email = email)
THEN
SET @s = 'UPDATE users SET _pass = ? WHERE users._email=?';

```

					ДП.КН. 23.515.22 000 ПЗ	Арк.
						68
Зм.	Арк.	№ докум.	Підп.	Дата		

```

        - PREPARE stmt FROM @s;
EXECUTE stmt USING pass, email;
        - END IF;
END //

DELIMITER //
DROP PROCEDURE IF EXISTS delete_user//
CREATE PROCEDURE delete_user (IN id INT)
BEGIN
    - IF EXISTS(SELECT 1 FROM users WHERE users._id = id) THEN
    - DELETE FROM users WHERE users._id = id;
    - END IF;
END //

```

Лістинг А7 – Запит на створення зберезувальних процедур order_str

```

DELIMITER //
DROP PROCEDURE IF EXISTS create_order_str//
CREATE PROCEDURE create_order_str ( IN article VARCHAR(6), IN
client INT, IN price FLOAT, IN counter INT, IN date DATE )
BEGIN
INSERT INTO order_str (order_str._article, order_str._client,
order_str._price, order_str._count, order_str._date ) VALUES(
article, client, price, counter, date );
END //

DELIMITER //
DROP PROCEDURE IF EXISTS update_order_str//
CREATE PROCEDURE update_order_str (IN id INT, IN article
VARCHAR(6), IN client INT, IN price FLOAT, IN counter
VARCHAR(50), IN date DATE )

BEGIN

UPDATE order_str SET order_str._article=article,
order_str._client=client,
order_str._price=price,
order_str._count=counter, order_str._date=date WHERE
order_str._id = id;

END //

DELIMITER //
DROP PROCEDURE IF EXISTS delete_order_str//
CREATE PROCEDURE delete_order_str (IN id INT)

```

					ДП.КН. 23.515.22 000 ПЗ	Арк.
						69
Зм.	Арк.	№ докум.	Підп.	Дата		

```
BEGIN

DELETE FROM order_str WHERE order_str._id = id;

END //
```

					ДП.КН. 23.515.22 000 ПЗ	Арк.
						70
Зм.	Арк.	№ докум.	Підп.	Дата		

Додаток Б

Реалізація функціоналу

Лістинг Б1- розмітка форми реєстрації

```
private ICommand _CloseApplication;
    public ICommand CloseApplicationCommand =>
        _CloseApplication ??= new
        LamdaCommand(OnCloseApplicationCommandExecuted,
        CanCloseApplicationCommandExecute);
    private bool CanCloseApplicationCommandExecute(object p)
=> true;
    private void OnCloseApplicationCommandExecuted(object p)
    {
        try
        {

            SimpleUsersDTO _user = new SimpleUsersDTO()
            {
                Login = _window.Login1.Text,
                Password = _window.Password1.Text,
                Name = _window.Name1.Text,
                LastName = _window.LastName1.Text,
                Birthday = _window.Date1.SelectedDate.Value,
                Email = _window.Gmail1.Text
            };

            if (_user != null)
            {
                services1.Add(_user);
                _window.Close();
            }
            else
            {
                MessageBox.Show("ой щось пішло не так");
                _window.Close();
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show($"ой щось пішло не так\n {ex}");
        }
    }
```

					ДП.КН. 23.515.22 000 ПЗ	Арк.
						71
Зм.	Арк.	№ докум.	Підп.	Дата		

Лістинг Б2 – Форма авторизації

```
private RelayCommand addCommand;
    public RelayCommand AddCommand
    {

        get
        {

            return addCommand ??
                (addCommand = new RelayCommand(obj =>
                {
                    var window = new Registrs(this)
                    {
                        Owner =
Application.Current.MainWindow
                    };
                    _window = window;

                    window.ShowDialog();

                }));
        }
    }
```

Продовження лістингу Б2

```
private RelayCommand logCommand;
    public RelayCommand LoginCommand
    {

        get
        {

            return logCommand ??
                (logCommand = new RelayCommand(obj =>
                {

                    if (services1.IsLog(logtext, pastext))
                    {
                        var window1 = new KryteVikno(this)
                        {
                            Owner =
Application.Current.MainWindow
                        };
                        _kryte = window1;
                    }
                }));
        }
    }
```

					ДП.КН. 23.515.22 000 ПЗ	Арк.
						72
Зм.	Арк.	№ докум.	Підп.	Дата		

```

        window1.ShowDialog();

    }
    else
    {
        MessageBox.Show("Неправильний пароль
або логін");
    }

    }));
    }
}

```

Лістинг Б3 – Кнопка додавання і виводу

```

private RelayCommand _dodel;
public RelayCommand Dodel
{
    get
    {
        return _dodel ??
            (_dodel= new RelayCommand(obj =>

                var window3 = new DodElements(this)
                {
                    Owner =
Application.Current.MainWindow
                };
                _dod = window3;

                window3.ShowDialog();

            }
            ));
    }
}

```

Лістинг Б4 – Кнопка додавання елементу в БД

```

private ICommand _DodApplication;

```

					ДП.КН. 23.515.22 000 ПЗ	Арк.
						73
Зм.	Арк.	№ докум.	Підп.	Дата		

```

        public ICommand DodApplicationCommand => _DodApplication
        ??= new LamdaCommand(OnDodCommandExecuted,
        CanDodApplicationCommandExecute);
        private bool CanDodApplicationCommandExecute(object p)
        => true;
        private void OnDodCommandExecuted(object p)
        {

            try
            {
                BootDTO _book;
                if (istext1 != "Нааявний")
                {
                    _book = new BootDTO()
                    {
                        Color = _dod.nameb.Text,
                        Model = _dod.nameg.Text,
                        Size = _dod.namea.Text,
                        TotalPrtice = _dod.namep.Text,
                        Shop = _dod.name1S.Text,
                        State = _dod.name2S.Text,
                        MyDate = _dod.Date1.DisplayDate
                    };
                }
                else
                {

                    _book = new BootDTO()

```

Продовження лістингу Б4

```

        {
            Color = _dod.nameb.Text,
            Model = _dod.nameg.Text,
            Size = _dod.namea.Text,
            TotalPrtice = _dod.namep.Text,
            Shop = _dod.name1S.Text,
            State = _dod.name2S.Text

        };
    }
    if (_book != null)
    {
        services2.Add(_book);
        _dod.Close();
    }
    else

```

					ДП.КН. 23.515.22 000 ПЗ	Арк.
						74
Зм.	Арк.	№ докум.	Підп.	Дата		

```

        {

            MessageBox.Show("ой щось пішло не так");
            _dod.Close();

        }
    }catch(Exception ex)
    {
        MessageBox.Show("ой щось пішло не так");
    }

}

```

Лістинг Б5 – Кнопка отримати все

```

private RelayCommand _getAll;
public RelayCommand GetAll1
{
    get
    {
        return _getAll ??
            (_getAll = new RelayCommand(obj =>
            {
                Boots = new
ObservableCollection<BootDTO>(services2.GetAll());
                _kryte.BIGDATA.ItemsSource = Boots;

            }
            ));
    }
}

```

Лістинг Б6 – Команда редагувати

```

private RelayCommand _rename;
public RelayCommand Rename
{
    get
    {
        return _rename ??
            (_rename = new RelayCommand(obj =>

```

					ДП.КН. 23.515.22 000 ПЗ	Арк.
						75
Зм.	Арк.	№ докум.	Підп.	Дата		

```

        {
            int l =
Convert.ToInt32(_kryte.ren1.Text);

            var window5 = new DodElements(this)
            {
                Owner =
Application.Current.MainWindow
            };
            _dod = window5;
            services2.Remove(1);
            window5.ShowDialog();
        }
    ));
}
}
}

```

Лістинг Б7 – Команда видалити

```

private RelayCommand _remove;
public RelayCommand Remove_
{
    get
    {
        return _remove ??
            (_remove = new RelayCommand(obj =>
            {
                int l =
Convert.ToInt32(_kryte.Remtext.Text);
                services2.Remove(1);
                Boots = new
ObservableCollection<BootDTO>(services2.GetAll());
                _kryte.BIGDATA.ItemsSource = Boots;
            }
            ));
    }
}
}
}

```

Лістинг Б8 – Команда пошуку

```

private RelayCommand _search;
public RelayCommand Search

```

					ДП.КН. 23.515.22 000 ПЗ	Арк.
						76
Зм.	Арк.	№ докум.	Підп.	Дата		

```

        {
            get
            {
                return _search ??
                    (_search = new RelayCommand(obj =>
                    {
                        string l = _kryte.ren2.Text;
                        string d = _kryte.shop1.Text;

                        Boots = new
ObservableCollection<BootDTO>(services2.FindDTO(1, 1, 1, 1, d,
1)));

                        _kryte.BIGDATA.ItemsSource = Boots;

                    }
                ));
            }
        }
private string pumatext = "Puma";
public string Pumatext
{
    get
    {
        return pumatext;
    }
    set
    {
        if (pumatext == value)
        {
            return;
        }
        pumatext = value;
    }
}
private string istext = "Наявний";
public string Istext
{
    get
    {
        return istext;
    }
    set

```

					ДП.КН. 23.515.22 000 ПЗ	Арк.
						77
Зм.	Арк.	№ докум.	Підп.	Дата		

```

        {
            if (istext == value)
            {
                return;
            }
            istext = value;
        }
    }
    /// <summary>
    /// //add
    /// </summary>
    private string istext1 = "Наявний";
    public string Istext1
    {
        get
        {
            return istext1;
        }
        set
        {
            if (istext1 == value)
            {
                return;
            }
            istext1 = value;
        }
    }

    //команда пошуку пума
    private RelayCommand _searchPuma;
    public RelayCommand SearchPuma
    {
        get
        {
            return _search ??
                (_search = new RelayCommand(obj =>

                    string d = _kryte.shop1.Text;

                    Boots = new
ObservableCollection<BootDTO>(services2.FindDTO(d, d, d, d, d,
d));

                    _kryte.BIGDATA.ItemsSource = Boots;
                ));
        }
    }

```

					ДП.КН. 23.515.22 000 ПЗ	Арк.
						78
Зм.	Арк.	№ докум.	Підп.	Дата		


```

        }
        ));
    }
}
private string istext2 = "Puma";
public string Istext2
{
    get
    {
        return istext2;
    }
    set
    {
        if (istext2 == value)
        {
            return;
        }
        istext2 = value;
    }
}

public void lol1()
{
    string d = _kryte.shop1.Text;
    Boots = new
ObservableCollection<BootDTO>(services2.FindDTO(d, d, d, d, d,
d));
}

```

					ДП.КН. 23.515.22 000 ПЗ	Арк.
						79
Зм.	Арк.	№ докум.	Підп.	Дата		

Додаток В

Тестування зберезувальних процедур

Лістинг В1 – запити для тестування ЗП

```
SELECT * FROM lead;
CALL create_lead( "lead1", "1111", "email1@email" );
  SELECT * FROM lead; CALL update_lead ( 1, "lead_update", "222",
"email12@email"
  );
SELECT * FROM lead;

SELECT * FROM shop;
  CALL create_shop(1, "address1", 100.1, 140.7, 50 );
SELECT * FROM shop;
CALL update_shop( 1, 1, "address_edited", 101.6, 130.9, 70
  );

  SELECT * FROM shop;

  SELECT * FROM seller;
CALL create_seller ( 1, "seller1", "1111", "email1@gmail.com"
  );

  SELECT * FROM seller;
CALL update_seller ( 1, 1, "seller_edited", "2222",
"email_edited@email" );
  SELECT * FROM seller;
SELECT * FROM client;
CALL create_client ( "client1", "1111", "email@email" );

SELECT * FROM client;
  CALL update_client ( 1, "client_edited", "2222",
"email_edited@email");
SELECT * FROM client;
  SELECT * FROM goods;
  CALL create_goods ( "000001", 1, "goods1", 10.5, 5 );
  SELECT * FROM goods; CALL update_goods ( "goods2", 11, 2,
"000001" );

SELECT * FROM goods;
  SELECT * FROM users;
CALL create_user ( "user1", "1111", "email1@email", "qwerty" );
SELECT * FROM users;
  CALL change_user_pass ("email1@email", "no-qwerty" );
  SELECT * FROM users;
```

					ДП.КН. 23.515.22 000 ПЗ	Арк.
						80
Зм.	Арк.	№ докум.	Підп.	Дата		

```

CALL delete_user ( 1 );
SELECT * FROM users;

SELECT * FROM order_str;
CALL create_order_str( "000001", 1, 150, 2, '2021-06-04' );
SELECT * FROM order_str;
CALL update_order_str( 1, "000001", 1, 160, 2, '2021-06-04' );
SELECT * FROM order_str;
CALL delete_order_str( 1 );
SELECT * FROM order_str;
CALL delete_goods ( 1 );
SELECT * FROM goods;
CALL delete_client ( 1 );
SELECT * FROM client;
CALL delete_seller ( 1 );
SELECT * FROM seller;
CALL delete_shop ( 1 );
SELECT * FROM shop;
CALL delete_lead ( 1 );
SELECT * FROM lead;

```

					ДП.КН. 23.515.22 000 ПЗ	Арк.
						81
Зм.	Арк.	№ докум.	Підп.	Дата		

ВІДГУК

на дипломний проєкт

студентки відділення комп'ютерних технологій

Галицького фахового коледжу імені В'ячеслава Чорновола

студентки IV курсу групи КН-41

Кушко Тетяні Василівній

(прізвище та ініціали)

Спеціальність 122, „Комп'ютерні науки”

Керівник ДП: Івасьєв С.В.

Тема: «Інформаційна система обліку матеріальних цінностей для автоматизованої системи ведення торгівлі»

1. Загальна характеристика студента: студен при виконанні дипломного проєкту продемонстрував позитивні сторони та набуті фахові навички. Поставлені завдання виконував старанно та вчасно.

2. Практична або теоретична цінність опрацьованих питань: розраблено програмний засіб для ведення обліку матеріальних цінностей, що дозволяє ефективно автоматизувати бізнес процеси у торгівлі. Проведено тестування системи та виконана розробка інтерфейсу користувача. Проведена економічна оцінка проєктного рішення автоматизації обліку сімейного бюджету.

3. Недоліки роботи: в роботі доцільно привести обладнання на якому проводилось тестування програмного засобу та навести мінімальні вимоги до нього.

4. Загальний висновок: робота відповідає вимогам до дипломних проєктів освітньо кваліфікаційного рівня «молодший спеціаліст» зі спеціальності 122 «Комп'ютерні науки» та заслуговує оцінку «відмінно»

Керівник дипломного проєкту _____



РЕЦЕНЗІЯ
на дипломний проєкт
студентки відділення комп'ютерних технологій
Галицького фахового коледжу імені В'ячеслава Чорновола

студентки IV курсу групи КН-41

_____ Кушко Тетяни Василівни _____
(прізвище та ініціали)

Спеціальність 122 «Комп'ютерні науки»

Обсяг дипломного проєкту: 81 стор.

Кількість сторінок записки: 62 стор.

Тема: «Інформаційна система обліку матеріальних цінностей для
автоматизованої системи ведення торгівлі»

1. Актуальність теми: зміст дипломного проєкту відповідає назві та поставленому завданню. Розкрито актуальність теми _____


2. Практична або теоретична цінність опрацьованих питань: _____
у роботі показана практична значущість. Автор мотивує своє рішення щодо вибраної платформи розробки та реалізації інтерфейсної частини сайту компанії _____

3. Недоліки роботи: _____ суттєвих недоліків не виявлено.

4. Загальний висновок _____ дипломний проєкт виконаний згідно вимог і заслуговує на позитивну оцінку.

Рецензент _____ Кузик В.М., викладач комп'ютерних дисциплін ЦК
Інформатики та комп'ютерних дисциплін _____

«23» червня 2023р.



(підпис)

Ім'я користувача:
Василь Кузик

Дата перевірки:
14.06.2023 12:38:08 EEST

Дата звіту:
14.06.2023 12:42:28 EEST

ID перевірки:
1015599803

Тип перевірки:
Doc vs Internet + Library

ID користувача:
100012366

Назва документа: Кушко

Кількість сторінок: 52 Кількість слів: 8043 Кількість символів: 62895 Розмір файлу: 5.08 MB ID файлу: 1015247713

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

6.8%
Схожість

Найбільша схожість: 4.4% з джерелом з Бібліотеки (ID файлу: 1008320885)

1.44% Джерела з Інтернету

59

Сторінка 54

6.02% Джерела з Бібліотеки

27

Сторінка 54

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

1

Підозріле форматування

9
сторінок