

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: «Розробка системи прийняття рішень структурної одиниці
комерційної мережі з підвищеною продуктивністю роботи з
врахуванням парадигм ООП, C#, MVC»

Виконав: студент IV курсу, групи СП-42
спеціальності _____

121 – Інженерія програмного забезпечення

(шифр і назва спеціальності)

(підпис) Дячишин Д.А.
(прізвище та ініціали)

Керівник _____
(підпис) Цуприк Г.Б.
(прізвище та ініціали)

Нормоконтроль _____
(підпис) Стоянов Ю.М.
(прізвище та ініціали)

Завідувач кафедри _____
(підпис) Петрик М.Р.
(прізвище та ініціали)

Рецензент _____
(підпис) Матійчук Л.П.
(прізвище та ініціали)

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет _____ комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра _____ програмної інженерії
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Петрик М.Р.

(підпис)

(прізвище та ініціали)

« _____ » _____ 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня _____ бакалавр
(назва освітнього ступеня)

за спеціальністю _____ 121 – Інженерія програмного забезпечення
(шифр і назва спеціальності)

студенту _____ Дячишину Денису Андрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи _____ «Розробка системи прийняття рішень
структурної одиниці комерційної мережі з підвищеною продуктивністю роботи
з врахуванням парадигм ООП, С, MVS».

Керівник роботи _____ Цуприк Галина Богданівна, к.т.н., доц.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 18 » квітня 2025 року № 4/7-333

2. Термін подання студентом завершеної роботи « _____ » червня 2025 р.

3. Вихідні дані до роботи: Завдання на розробку системи

4. Зміст роботи (перелік питань, які потрібно розробити)

Обрати напрям дослідження та об'єкт, сформулювати, погодити та затвердити тему кваліфікаційної роботи; розробити та затвердити завдання;

проаналізувати завдання, зробити огляд та проаналізувати бібліографічні матеріали щодо

стану справ та тенденцій в обраному напрямку дослідження;

сформулювати завдання, обґрунтувати цілі дослідження;

розробити та спроектувати;

описати технології, етапи розробки та сам продукт,

розробити план тестування програмного продукту; оформити допоміжну документацію;

проаналізувати роботу щодо питань з дотримання положень про безпеку життєдіяльності та

основи охорони праці; оформити пояснювальну записку; зробити відповідні висновки

за результатами виконаної роботи, представити роботу на розгляд ЕК.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

Ілюстративна частина кваліфікаційної роботи оформляється у вигляді слайдів, висвітлює основні розділи та етапи роботи та містить:

тему роботи, мету, предмет та об'єкт дослідження,

сформульовані завдання, перелічено етапи виконання кваліфікаційної роботи, короткий

аналіз актуальності теми, варіант архітектури,

представлено опис програмного забезпечення; опис програмного забезпечення та тестування.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності,	Лазарюк В.В.	19.06.2025	19.06.2025
основи охорони праці			
Нормоконтроль	Стоянов Ю.М.	15.06.2025	15.06.2025

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Вибір та погодження теми з керівником. Створення та затвердження завдання	27.01-02.02.25	виконано
2.	Затвердження теми КР наказом ректора	18.04.2025	виконано
3.	Аналіз технічного завдання, підбір та аналіз бібліографічних матеріалів необхідних для виконання кваліфікаційної роботи	20.04.2025	виконано
4.	Проектування та розробка програмного продукту	11.05.2025	виконано
5.	Тестування та дослідна експлуатація програмного забезпечення	19.05.2025	виконано
6.	Створення допоміжної документації. Написання та перевірка на відповідність вимогам пояснювальної записки	25.05.2025	виконано
7.	Формування записки кваліфікаційної роботи	28.05.2025	виконано
8.	Перевірка програмою антиплагіат	29.05.2025	виконано
9.	Безпека життєдіяльності, основи охорони праці	19.06.2025	виконано
10.	Нормоконтроль	15.06.2025	виконано
11.	Попередній захист	09.06.2025	виконано
12.	Захист кваліфікаційної роботи	26.06.2025	виконано

Студент

(підпис)

Дячишин Д.А.

(прізвище та ініціали)

Керівник роботи

(підпис)

Цуприк Г.Б.

(прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота ОР «бакалавр» містить: 78с., 13 рис., 6 табл, 42 літ. джер., 14 дод.

Кваліфікаційна робота, аналіз проблеми, дослідження, технології розробки, програмний продукт, проєктування, конструювання, ооп, с#, mvs, безпека життєдіяльності, охорона праці

Об'єктом розроблення є система реалізації з підвищеною продуктивністю роботи з врахуванням парадигм об'єктно-орієнтованого підходу.

Предметом дослідження був аналіз та вибір підходу розробки, зокрема розглянуто об'єктно-орієнтований та процедурний підходи, проаналізовано їхні переваги та недоліки. Для створення схеми бази даних виокремлено основні сутності, сконструйовано концептуальну модель бази даних, UML діаграми класів, обґрунтовано вибір мови програмування.

Ціль роботи полягає в використанні сучасних ІТ технологій при розробці безпечної системи, яка б оптимізувала та автоматизувала роботу та була спроможна вирішувати конкретні завдання відповідно до вимог замовника.

Система розроблена для Windows, мовою С#, в якості середовища розробки обрано Microsoft Visual Studio.

ABSTRACT

The qualification work for the bachelor's degree program contains: 78 p., 13 fig., 6 tables, 42 lit. ger., 14 add.

Qualification work, analysis of problem , research, development technologies, software product, design, construction, object-oriented programming, c#, mvs, life safety, occupational safety

The object of development is an implementation system with increased work productivity taking into account the paradigms of the object-oriented approach.

The subject of the study was the analysis and choice of the development approach, in particular, the object-oriented and procedural approaches were considered, their advantages and disadvantages were analyzed. To create a database schema, the main entities were identified, a conceptual database model was constructed, UML class diagrams were constructed, and the choice of programming language was justified.

The goal of the work is to use modern IT technologies in the development of a secure system that would optimize and automate work and be able to solve specific tasks in accordance with the customer's requirements.

The system was developed for Windows, in C#, Microsoft Visual Studio was chosen as the development environment

ЗМІСТ

ВСТУП	8
1 ПРОЄКТУВАННЯ ТА КОНСТРУЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ.	11
1.1 Аналіз по предметній області	11
1.2 Обґрунтування використаних технологій	14
1.2.1 Обґрунтування вибору MS Visual Studio	14
1.2.2 Обґрунтування вибору об'єктно-орієнтованої мови C#	15
1.2.3 Обґрунтування вибору інструмента візуального проєктування баз даних	16
1.2.4 Обґрунтування доцільності використання MySQL	17
1.3 Пошук актантів та варіантів використання	18
1.4 Опис архітектури програмної системи	22
1.4.1 Побудова UML-діаграми залежності класів	25
1.5 Структура бази даних	27
1.5.1 Перелік таблиць для розроблюваної системи	28
1.5.2 Структура таблиць	29
1.5.3 Побудова схеми реляційної бази даних	33
1.6 Реалізація основних класів та методів	34
2 ТЕСТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ	42
2.1 Вибір середовища тестування	42
2.1.1 Тестування ПЗ	43
2.1.2 Автоматизоване тестування програмного продукту	47
3 Безпека життєдіяльності, основи охорони праці	50
5.1 Безпека життєдіяльності	50
5.1.2 Ліквідація надзвичайних ситуацій	50
5.2 Основи охорони праці	55
ВИСНОВКИ	59
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	61

ДОДАТКИ.....	65
Додаток А Апробація результатів роботи.....	66
Додаток Б Слайди презентації.....	71
Додаток В Відгук	
Додаток Г Рецензія	
Додаток Д Диск	

ВСТУП

Вибір теми роботи пов'язаний саме з тим, що сьогодні відбувається суттєве переформатування та фактично зміна вимог до такого роду програмних продуктів. Пов'язано це з різними причинами і перш за все з тривалим терміном дистанціювання, від ковіду і аж до сьогодні коли вже четвертий рік триває повномасштабне вторгнення

Сьогодні можна спостерігати значним поширенням закладів мульти торгівлі, в яких можна придбати найрізноманітніші товари (Аврора, Та-да, Prostor та інші: від їжі й одягу до товарів для риболовлі та дрібної техніки і т.д.). Такий підхід дещо ускладнює роботу, адже для забезпечення безперебійної і злагодженої роботи таких закладів потрібна одночасна кропітка робота багатьох працівників, які забезпечують облік товарів та аналіз, вчасне і рівномірне наповнення складу, здійснення швидке оформлення. Не дивлячись на ніби то простоту, така діяльність вимагає заздалегідь продуманих і вчасних наказів від керівництва, компетентності працівників та їх кваліфікованість, досвіду, та й контролю в цілому.

Таким чином метою обраної реалізації варта означити пошук , обґрунтування та реалізацію можливості для оптимізації та автоматизації такої роботи та можливості контролю та прийняття рішень навіть віддалено.

Аналізуючи існуючі рішення варта зазначити, на виробництві вже вкрай актуальна та швидко розвивається автоматизація процесів. Таку ж мету можна реалізувати і в закладах торгівлі за рахунок автоматизованих інформаційних систем з можливим врахуванням специфіки їх діяльності. В системах такого типу часто застосовують бази даних, в якій міститься уся необхідна інформація. Зв'язок з базою даних по мережі дозволяє одночасну та безперебійну роботу кількох користувачів такої системи, що в свою чергу покращує комунікацію та ефективність роботи. Завдяки автоматизованим процесам, можна відмовитися від значної кількості персоналу, а також суттєво скоротити затрати на обробку

значної кількості документів. Збільшується швидкість операції продажу товарів, що дозволяє бачити стан речей «в цілому».

Така система дозволить працівникам виконувати свої функції, зосереджуючись лише на їх безпосередньому завданні. В цілому такі програми є продуктивні і широко поширюються у своїх сферах діяльності. Саме з цієї причини ефективніше замовити таку розробку, акцентуючи увагу на вимоги конкретного замовника, ніж використовувати та пробувати їх допрацьовувати вже існуючі.

Також важливим критерієм такого типу системи повинен бути захист інформації від стороннього втручання. В розроблюваній програмній системі такий захист забезпечується авторизацією, за допомогою якої працівник зможе увійти в систему під своїм логіном і паролем чи при необхідності змінювати його так часто, наскільки це потрібно, таким чином забезпечуючи надійність інформаційної системи .

Основними завданнями розробки є:

- 1) проєктування бази даних для збереження інформації якою володітиме адміністратор під час своєї роботи;
- 2) проєктування бази даних для збереження інформації, якою володітиме працівник під час своєї роботи;
- 3) проєктування бази даних для збереження інформації про певний товар, який знаходиться в магазині;
- 4) моделювання основних класів;
- 5) розробка дизайну з спрощеним інтуїтивно зрозумілим інтерфейсом;
- 6) розробка безпосередньо самого програмного продукту;

Розробка програми здійснюватиметься з допомогою використання можливостей мови програмування C#. Розробку програми можна розбити на створення інтерфейсу, який призначений для взаємодії користувача з системою, та розробку функціоналу програми, який відповідає за роботу самої програмної системи. Програмний інтерфейс та програмний функціонал повинні взаємодіяти між собою .

Як засіб для написання програмного продукту запропоновано використати мову програмування C#, основною перевагою якої є простота розробки таких програм та зручна взаємодія з базою даних. І як наслідок в якості середовища розробки буде використовуватись Microsoft Visual Studio.

В якості СУБД запропоновано MySQL Server, оскільки – це СУБД, яку досить легко налаштувати і підключити до даної програмної системи. Не варто також забувати про велику швидкість виконання команд та можливість безперебійно працювати із великим обсягом різнотипної інформації, наявність простої і ефективної системи безпеки, що вплинуло на вибір бази даних.

1 ПРОЄКТУВАННЯ ТА КОНСТРУЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

1.1 Аналіз по предметній області

Першим етапом при проєктування бази даних не важливо якого типу вважається детальний аналіз по предметній області, який зазвичай завершується побудовою концептуальної схеми / інформаційної структури. На цьому етапі аналізують запити користувачів, вибирають інформаційні об'єкти та їх основні характеристики а на основі здійсненого аналізу формують структуру предметної області, яка як правило не залежить ані від програмного ані від технічного середовища, в яких буде здійснено реалізацію програми [1-3].

Дана предметна область представляється торгівельною мережею для магазину електронних приладів та компонент. При використанні паперової обробки інформації виникають наступні проблеми: потрібна велика кількість працівників, і обробка інформації є повільною та незручною [4-7].

Для вирішення цих проблем потрібно створити програмний продукт, який би оптимізував та забезпечив роботу всіх працівників, діяльність яких стосується обробки інформації. Система дозволить адміністратору відслідковувати всі операції продажу, також можна переглянути інформацію про співробітників даного торгівельного закладу [8-13].

Система призначена для використання трьома типами акторів серед яких старший адміністратор, адміністратор і касир.

Касир – його головною роботою є забезпечення продуктивного та швидкого продажу товару . Основною характеристикою даного користувача є його швидкість і надійність роботи.

Адміністратор – він наділений функціями такими як : додавання нового товару , коректування інформації , видалення товару а також має право додавати і видаляти касирів.

Старший адміністратор – людина яка сліткує за роботою торгівельної мережі.

База даних ж повинна містити дані про товар, який продає мережа , що включає всі його характеристики (ціну, найменування, дату виготовлення, інше). Необхідним є і відображення об'єднаної інформації про покупки, яка буде вміщувати дані про куплений товар, його ціну, дату продажу і т.д.

Інформаційна система повинна бути багатокористувацькою а також обмежувати доступ користувачів .

Дані користувача – в цій таблиці повинна зберігатись інформація про продавця, а саме:

- ім'я ;
- прізвище;
- логін ;
- пароль;
- електронна пошта.

Дані про товар – повинна зберігатись інформація про товар, а саме:

- назва;
- ціна;
- дата виготовлення;
- серійний номер;
- короткі відомості про товар;
- місце знаходження товару;
- категорія;
- країна виробник.

Дані про покупки – в цій таблиці повинна зберігатись інформація про продані товари, зокрема:

- ідентифікатор покупки;
- ідентифікатор товару;
- назва товару;
- ціна на товар;
- ідентифікатор продавця.

Дані про товари в корзині – в цій таблиці повинна зберігатись тимчасова інформація про товари в корзині:

- назва;
- ціна;
- дата виготовлення;
- серійний номер;
- статус товару;
- дата продажу;
- короткі відомості про товар;
- категорія;
- країна-виробник.

Отже, проаналізувавши предметну область, я зробив висновок про те, що потрібно реалізувати роботу як мінімум з п'яти основних сутностей: старший адміністратор, адміністратор, продавець, покупки і товар, також повинна бути взаємодія між цими сутностями.

Взаємодія — широкий загальний термін, що позначає таку сумісну дію кількох об'єктів або суб'єктів, при якій результат дії одного з них впливає на інші.

Така система в цілому буде надавати значний приріст в продуктивності роботи, без будь-яких сторонніх затрат (окрім затрат на її впровадження).

1.2 Обґрунтування використаних технологій

Далі я обґрунтую доцільність обирання конкретних технологій які, на мою думку, оптимально підходять для реалізації задуманої ідеї і та є доступними.

1.2.1 Обґрунтування вибору MS Visual Studio

MS Visual Studio є серією продуктів фірми Microsoft, які безпосередньо містять інтегроване середовище призначенням якого є розробка програмного забезпечення, а також цілий ряд інших, не менш важливих інструментальних засобів. Такі продукти дають можливість розробляти як програми консольного типу, так і програми із графічним інтерфейсом, включно із тими, що підтримують технологію Windows Forms, крім того web-сайти, web-застосунки, web-сервіси для всіх платформ, які підтримуються MSoft Windows, WindowsMobile, .NET Framew., .NET Compact Framew. та MS Silverlight [14-18].

Microsoft Visual Studio зазвичай включає один чи декілька із наступних компонент:

- Visual Basic NET , а до його появи Visual Basic;
- Visual C++ ;
- Visual C# ;
- Visual J# ;
- Visual Studio Debugger ;
- Microsoft SQL Server ;
- MSDE Visual Source Safe.

Попередньо в складі Visual Studio були включені наступні продукти:

- Visual InterDev;
- Visual J++ ;
- Visual J# ;
- Visual FoxPro;
- Visual Source Safe .

Visual Studio побудовано з застосуванням архітектури, якою підтримується можливість для використання Add-Ins –доповнень, тобто йдеться про плагіни від зовнішніх розробників, і це, в цілому дає, змогу розширити можливості безпосередньо середовища розробки.

1.2.2 Обґрунтування вибору об'єктно-орієнтованої мови C#

C# є об'єктно-орієнтованою мовою програмування із можливостями безпечної системи типізації під платформу .NET.

Важливо, що синтаксис C# є близьким і до C++ і до мови Java. Особливістю є те, що C# володіє строгою статичною типізацією та підтримує поліморфізм, враховане можливість перевантаження операторів, передбачено вказівники на функції-члени класів, атрибутів, події, властивості та винятки, є можливість залишати коментарі у зручному XML форматі. Цінним є й те, що багато можливостей перейшло від попередників, мови C++ та Delphi, Модули та Smalltalk та опираючись на практику їх використання виключено деякі моделі, які зарекомендувались як проблематичні при розробці програмних систем, зокрема, для прикладу, можна навести множинне упадкування класів [19-21] .

Крім цього, як вже я зазначав, C# є дуже подібною до мови програмування Java, створену для вирішення проблеми роззосереджених обчислень. Беручи за основу популярну C++, в Java виключено потенційно небезпечні моменти, наприклад, вказівники без можливості контролю виходу за межі. А вже для прямого її призначення - роззосереджених обчислень було розроблено концепцію

т.з. віртуальної машини, а також машинно-не залежного байт-коду, який є таким собі посередником між вихідним текстом програми та апаратними інструкціями для персонального комп'ютера чи інтелектуального пристрою іншого типу.

В результаті корпорацією Microsoft було створено свій власний продукт Java, який й було названо C#, який і успадкував від Java концепцію віртуальної машини (середовище.NET), байт-код (MSIL) і підвищену безпеку вихідного коду програм, крім цього врахувавши попередній досвід використання програм написаних на Java [22-24].

1.2.3 Обґрунтування вибору інструмента візуального проектування баз даних

MySQL Workbench вважається інструментом візуального проектування безпосередньо баз даних, в якому інтегруються процеси по проектуванню, з моделювання, створення та експлуатації бази до уніфікованого безшовного середовища під системи баз даних за типомMySQL [25].

Зазвичай MySQL Workbench [26] представлено двома редакціями, а саме:

1)Community Edition з поширенням під безкоштовною ліцензією (GNU GPL);

2)Standard Edition – редакція є доступною при наявності передоплати, яка є щорічною. Друга редакція містить додаткові ще функції, за допомогою яких можна підвищити продуктивність як розробників так і адміністраторів, які працюють із базами.

В пакеті MySQL GUI Tools міститься крос платформне загальнодоступне програмне забезпечення (GNU GPL) призначене під адміністрування безпосередньо серверів бази даних типу MySQL, а заразом і під можливість створювати і працювати з даними всередині самих баз даних даного типу. Цей пакет було розроблено корпорацією SunMicrosystems, проте, в подальшому, його

розробку було призупинено, і, станом на сьогодні, пакет доступний лише з архівів завантажень, які розміщено на сайті самої MySQL.

Інструмент My SQL Workbench вважається одним із перших сімей продуктів, які є доступними у 2-х варіантах. Для того щоби зацікавити розробника основною версією крім StandardEdition (комерційної стандартної версії програми) було випущено Community Edition – загально доступну безкоштовну версію (GNU GPL), яка є також повнофункціональним продуктом, в якому передбачено усі основні можливості, які є і в комерційному варіанті та який є основою для всіх наступних релізів, і є корисним для подальшого розвитку продукту. В свою чергу комерційна версія розширює можливості безкоштовної цілою серією модулів та плагінів, за допомогою яких можна оптимізувати робочий процес через зменшення затрат часу та мінімізацію можливих помилок.

1.2.4 Обґрунтування доцільності використання MySQL

My SQL є загально доступною СУРБД, метою розробки якої є підвищити швидкодію опрацювання громістких баз даних, що пропонує відкритий код, запропонована на протигагу альтернативним платним системам. Від створення є подібною до mSQL, однак, із розвитком розширювалась і сьогодні про MySQL [27] можна говорити як про одну із найбільш поширених систем такого напрямку. Основним її призначенням є можливість створювати динамічні веб-сторінки, завдяки підтримці зі сторони різних мов програмування.

Сточатку MySQL була спробою пристосувати mSQL під власні корпоративні розробки: таблиці, під які використовували під програми нижчого рівня по типу ISAM . Як результат, розрока нового інтерфейсу(SQL), але від m SQL залишили API інтерфейс.

З часом MySQL було сепаровано, що зробило процес розробки не прозорим, зокрема, повернули практику представлення до Enterprise версії MySQL розширених функцій, також було виключено можливість тестового набору, закрили доступ фактично повністю до всієї системи відстеження на помилки та закрили оприлюднення згрупованого логу по змінам, що дає змогу дати оцінку стосовно прив'язки патчів вже до конкретних змін.

1.3 Пошук актантів та варіантів використання

Виходячи з аналізу предметної області далі я визначатиму основні задачі, які повинна виконувати програма.

Для початку програмний продукт повинен володіти зручним інтерфейсом, який забезпечить продуктивну роботу в першу чергу виявлених акторів «продавця, адміністратора та старшого адміністратора».

Також в зв'язку з взаємодією програми з базою даних, необхідно забезпечити надійне з'єднання та безперебійну роботу програми. Потрібно передбачити ряд запитів до неї.

Також програма повинна запобігати випадковим спотворенням інформації.

Проаналізувавши предметну область перед розробкою програми я сформулюю задачі:

- вхід в систему;
- операція покупки товару;
- додавання/редагування товару;
- додавання працівників закладу;
- відображення інформації про товари з фільтрацією даних;
- відображення інформації про покупки ;
- створення кошика для зручності покупки ;
- видаляти товар який продано ;

— видаляти продавця якого звільнено по певній причині.

Для реалізації вище зазначених варіантів використання потрібно логічно пов'язати інформацію в базі даних.

Також необхідно реалізувати систему аутентифікації, для забезпечення проникнення стороннього доступу до бази даних , доступу до покупок і внесення некоректної інформації. Дані, що містять інформацію про авторизацію містяться в базі працівників, які будуть володіти полями «логін» і «пароль».

Таким чином, в результаті детального аналізу предметної області та вимог до програмного забезпечення я склав діаграму варіантів використання продавця (рис.1.1) , адміністратора (рис.1.2) і старшого продавця (рис.1.3).

»

Рисунок 1.1 – Представлення UML діаграми варіантів використання
(для продавця)

Діаграми варіантів використання зображена на рисунку 1.1 дає можливість побачити усі функціональні можливості , якими володіє продавець інформаційної системи.

За допомогою цих розділів продавець перш за все виконує вхід в систему, ввійшовши йому доступний такий функціонал: пошук товару який цікавить покупця, переглянути інформацію про товар, додати товар в корзину, при додаванні всіх товарів які покупець збирається придбати продавець допустив помилку і додав не той товар він легко може видалити його з корзини, після додавання товарів він може оформити замовлення(покупку), також по закінченні роботи чи коли продавцеві потрібно відійти на перерву чи з іншої причини за для безпеки він повинен виконати вихід з системи

•

Рисунок 1.2 – Представлення UML діаграми варіантів використання
(для адміністратора)

Функціонал адміністратора програмної системи зображені на рисунку 1.2 , вона дає можливість побачити функціональні можливості якими володіє адміністратор.

За допомогою варіантів використання адміністратор перш за все виконує вхід в систему , ввійшовши йому доступний такий функціонал : додавання нового товару який прибув у магазин також при помилковому вводі інформації про товар він може безпроблем редагувати некоректну інформацію і замінити її на достовірну або при помилковому внесенні товару в систему адміністратор успішно видалить такий товар , адміністратор додає в систему нового продавця і видаляє продавця який був звільнений з тої чи іншої причини , також по закінченні роботи чи коли адміністратору потрібно відійти на перерву чи з іншої причини за для безпеки він повинен виконати вихід з системи

»

Рисунок 1.3 – Представлення UML діаграми варіантів використання
(для старшого адміністратора)

Функціонал старшого адміністратора програмної системи зображені на рисунку 1.3, він дає можливість побачити функціональні можливості якими володіє старший адміністратор.

За допомогою варіантів використання старший адміністратор перш за все виконує вхід в систему, ввійшовши йому доступний такий функціонал : додавання нового адміністратора який прийнятий на роботу в магазин також при помилковому вводі інформації про адміністратора редагувати цю інформацію і замінити її на достовірну, видалити адміністратора при його звільненні з роботи , та переглядає звітність, також по закінченні роботи чи коли адміністратору потрібно відійти на перерву чи з іншої причини за для безпеки він повинен виконати вихід з системи.

1.4 Опис архітектури програмної системи

Мною було прийнято рішення, що оптимальним буде коли архітектура програми складатиметься з 3-ох основних частин, дві адміністративні та користувацька , структури яких зображена нижче [29-31].

Користувацька частина програми відображається при вході продавця в систему попередньо ввівши логін і пароль після чого продавцеві системи надається можливості необхідні для успішного продажу товару, що представлені на рисунку 1.3.

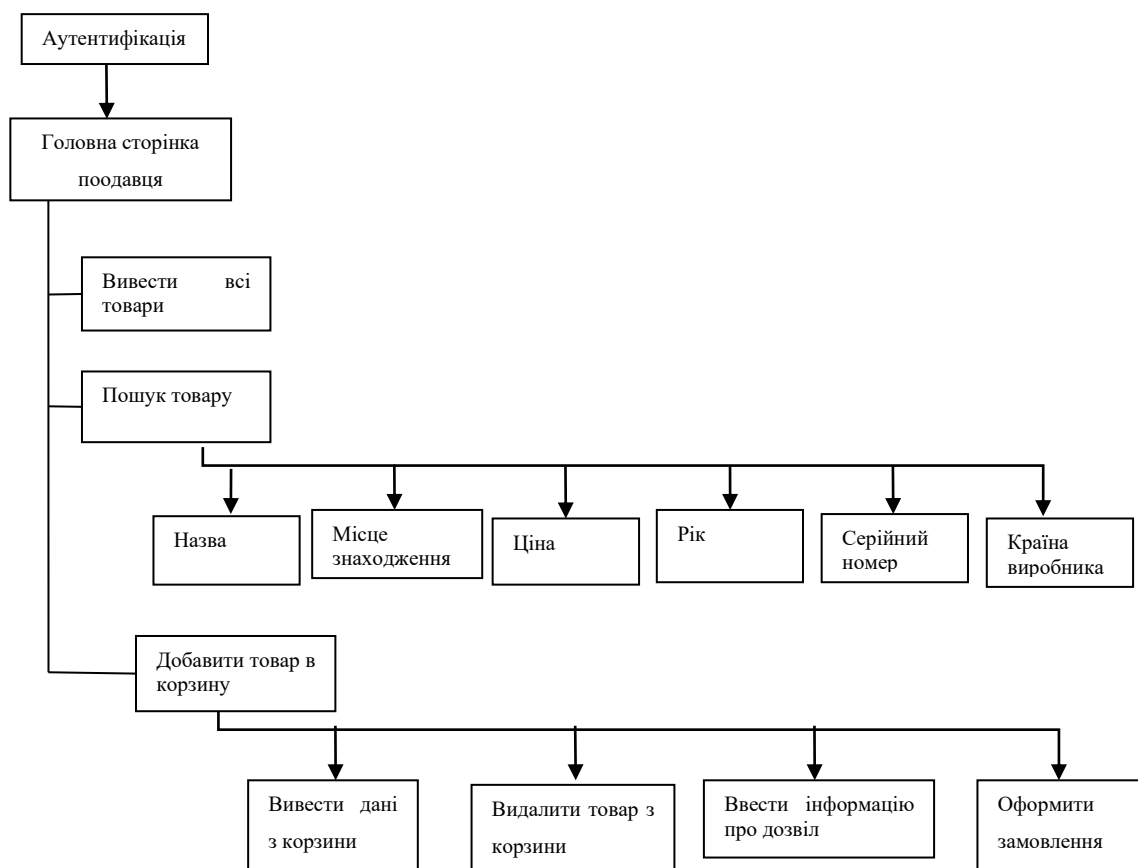


Рисунок 1.3 – Структура користувацької частини

Адміністративна частина програми відображається при вході адміністратора в систему попередньо ввівши логін і пароль після чого адміністратору системи надається можливості необхідні для керування програмною мережею, які представлені на рисунку 1.4, серед яких: додавання товару, видалення товару, додавання нового продавця, видалення продавця тощо.

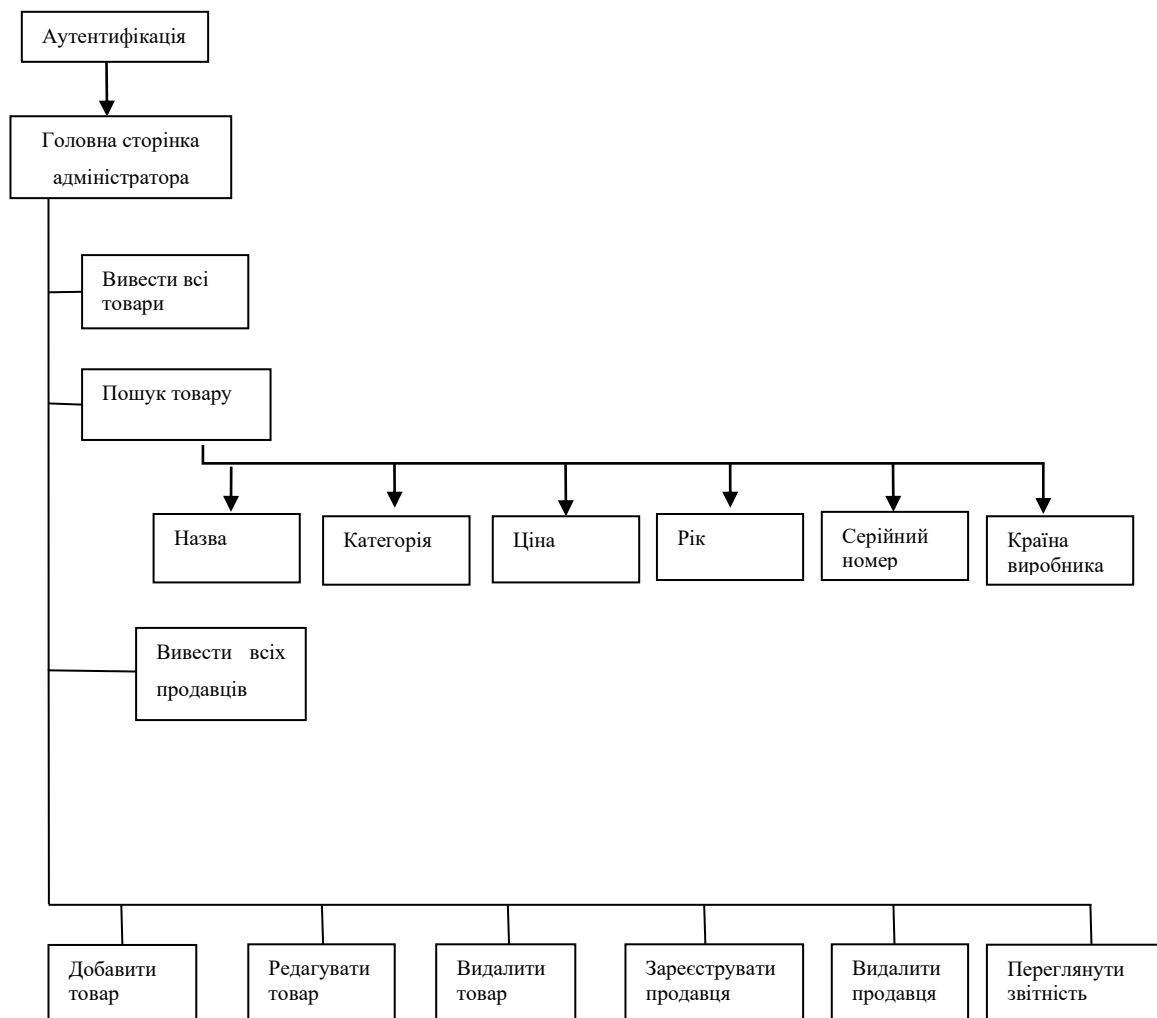


Рисунок 1.4 – Структура першої адміністративної частини

Адміністративна частина старшого адміністратора відображається при вході старшого адміністратора в систему попередньо ввівши логін і пароль після чого адміністратору системи надається можливості необхідні для керування програмною мережею, які представлені на рисунку 1.5 , серед яких : додавання адміністратора, видалення адміністратора, пошук проданого товару, перегляд звітності тощо.

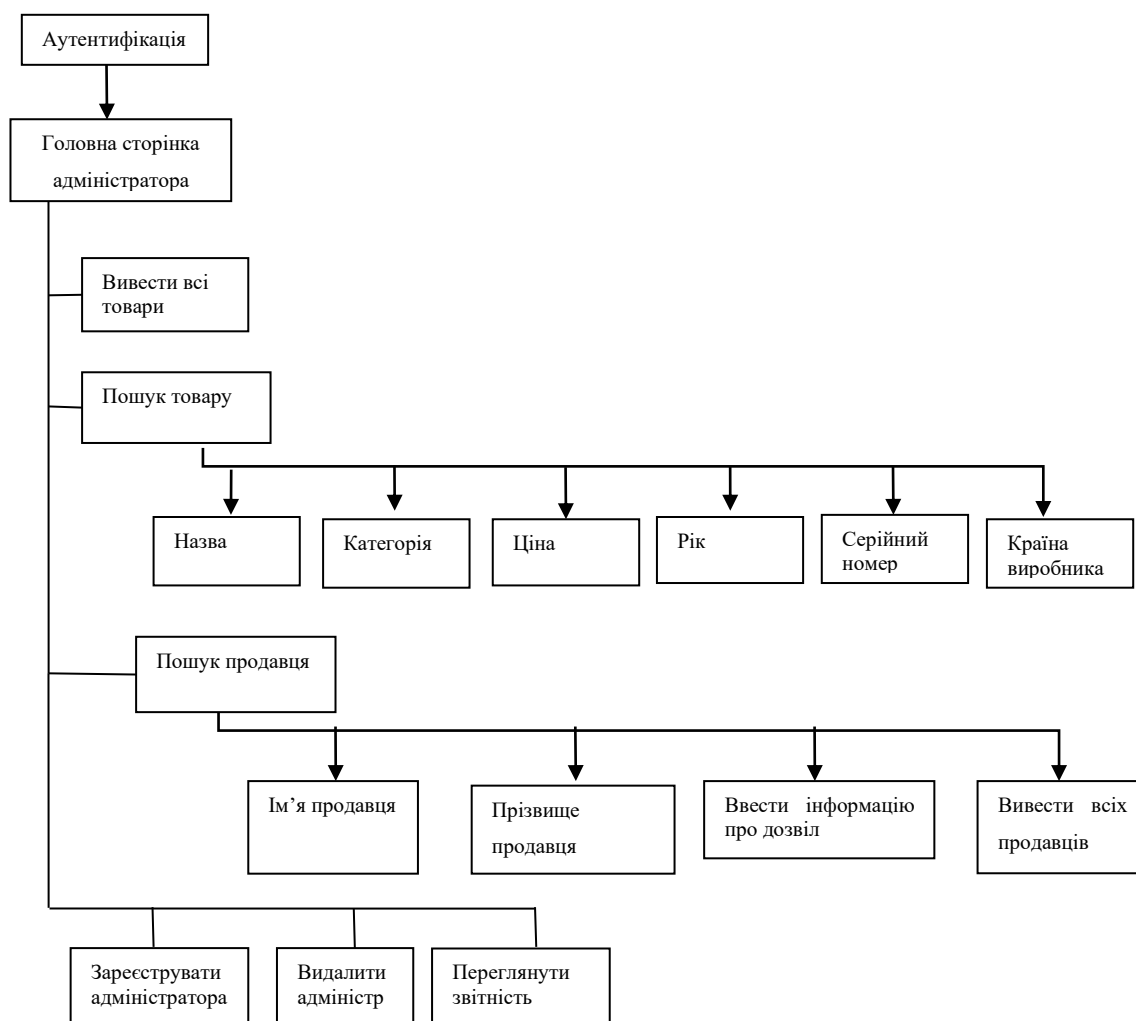


Рисунок 1.4 – Структура другої адміністративної частини

1.4.1 Побудова UML-діаграми залежності класів

Уніфіковану мову для моделювання, використовують в рамках парадигм об'єктно-орієнтованого програмування, як мову широкого профіля, з відкритим стандартом, в якій використовуються позначення графічного представлення призначенням яких є можливість створити абстрактну модель для проєктованої системи [32,33]. При створенні UML її призначенням планувалось визначення, візуалізація, проєктування та документування і стосувалось програмних систем в основному. Хоча і відомо, що призначенням уніфікованої мови для моделювання

не є безпосередньо мовою програмування, проте у засобах виконання моделей як проінтерпретованого коду є можливість згенерувати код.

Далі я використав UML для представлення діаграми класів, як статичного представлення структури самої моделі. На ній представлені статичні елементи, такі як основні класи, типи використаних даних та їхній зміст і наявні відношення. В цілому ж діаграма класів слугує для представлення саме моделі системи в статичній структурі в термінології та для класів з дотриманням парадигм об'єктно-орієнтованого програмування.

З використанням засобів UML, мною було спроектовано UML-діаграми залежності класів, яку я і представив на рис.4.1.

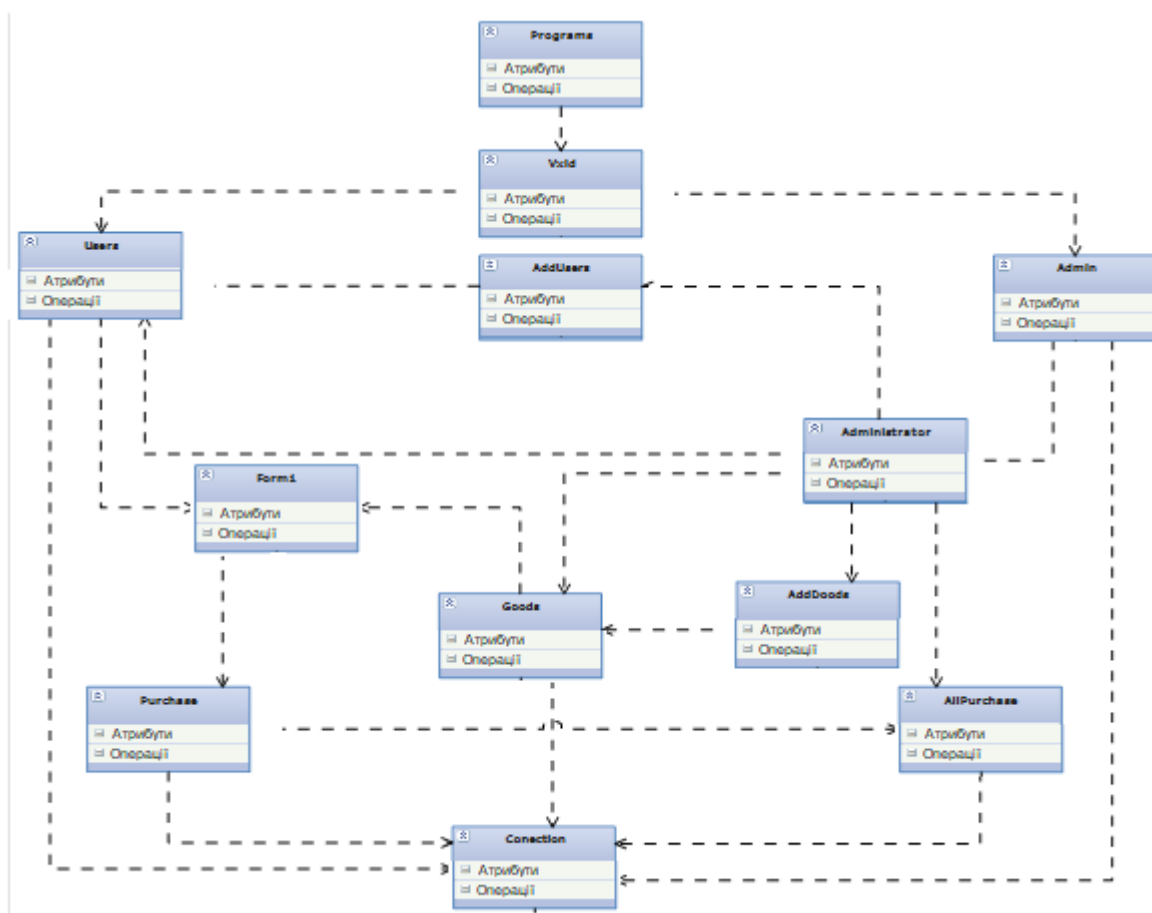


Рисунок 1.5 – UML діаграма залежності класів

1.5 Структура бази даних

База даних то є по суті впорядкований набір логічно взаємопов'язаних даних, які спільно використовуються, з основним призначенням, яке полягає в задоволенні інформаційних потреб користувача. Головним завданням їх є гарантоване, надійне зберігання значних обсягів інформації з можливістю отримати доступ до неї користувачу чи для прикладної програми.

Зазвичай, в базі даних містяться багато сутностей різних типів. Кожен із типів сутностей володіє своїм унікальним набором атрибутів, а кожна сутність в свою чергу володіє своїми власними значеннями для кожного із атрибутів. Ці властивості і визначають атрибути сутності якими і є властивості, якими буде володіти об'єкт, який розглядається. Крім цього база даних володіє набором відношень, які є логічно побудованими зв'язками в базі даних між сутностями.

Самі ж відношення задають при допомозі поєднання ключів, якими називають унікальне значення, яким дозволяється якимось способом ідентифікувати суть або частину суті предметної області. Таким чином ключем є значення певного атрибута чи атрибутів у кортежі відношення, який і представляє екземпляр суті для бази даних реляційного типу. Зазвичай, розглядають два типи ключів, це: первинний (мінімальну множину атрибутів, які є підмножиною заголовку данного відношення, складені значення яких унікально визначають кортеж відношення. Тобто, первинним ключем позначається поле чи група полів таблиці бази даних, значення якого (комбінація значень яких) використовують як унікальний ідентифікатор для запису цієї таблиці та зовнішні ключі); та зовнішній, який представляє собою поле таблиці, призначенням якого є збереження значення первинного ключа для другої таблиці із метою організувати зв'язок між цими таблицями.

Важливо, щоб таблиці були певним чином організовані, оскільки це дасть змогу забезпечити оптимальне використання інформації, зокрема, допоможе

уникнути повторів. З цією метою використовують нормальну форму, яка є фактично властивістю відношення в моделі даних реляційного типу, яка і характеризує його по питанню надмірності, виявлення якої може спровокувати логічно помилкові результати вибірки чи навіть зміни даних. Нормальна форма по суті своїй є сукупністю вимог, які і повинні задовольнятися відношенням.

1.5.1 Перелік таблиць для розроблюваної системи

Проаналізувавши інформацію про проєктовану систему я сформував наступні таблиці як в переліку в таблиці далі по тексту.

Таблиця 1.1– Перелік та призначення таблиць розроблюваної системи

Назва таблиці БД	Призначення
Administrator	Таблиця з обліковими записами адміністратора
Goods	Таблиця з переліком товарів
Purchase	Таблиця з переліком покупок
All purchase	Таблиця з тимчасовими записами які занесені в корзину
Users	Таблиця з обліковими записами користувачів
Super administrator	Таблиця з обліковими записами старшого адміністратора

З даної таблиці можна виявити наступні сутності:

- 1 адміністратор;
- 2 старший адміністратор
- 3 товари;
- 4 покупки;
- 5 корзина;
- 6 продавці.

На основі вище перелічених сутностей було створено ER – діаграму, яка представлена на рисунку 1.6 нижче.

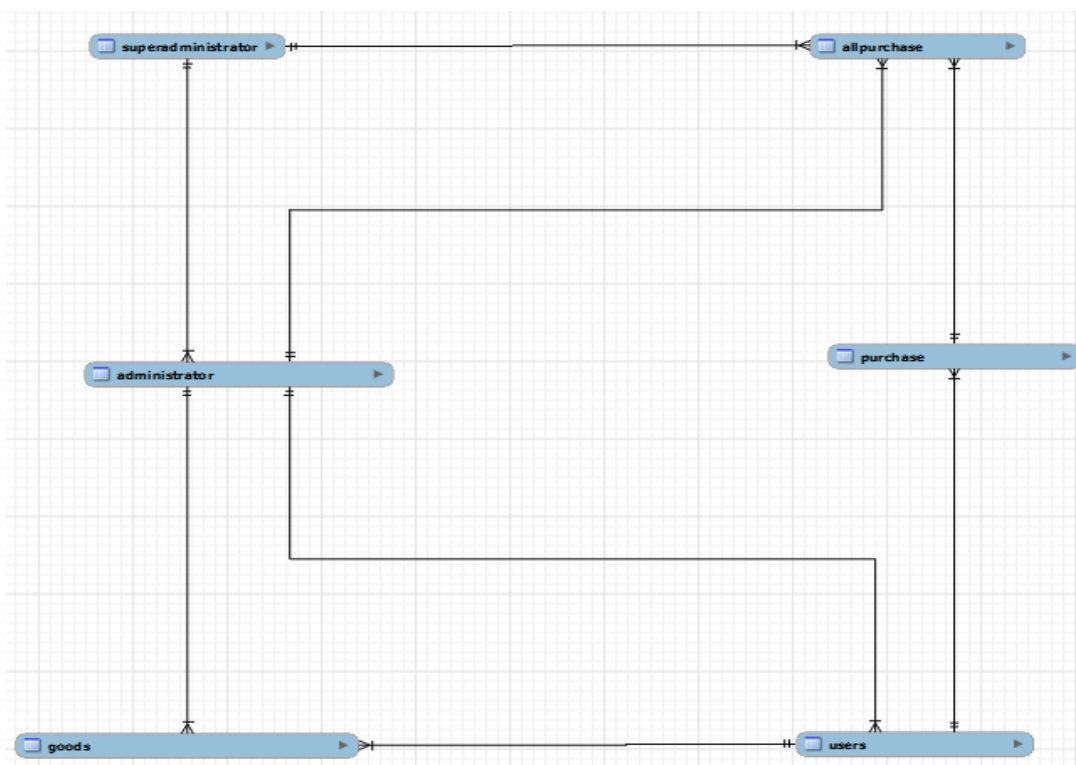


Рисунок 1.6 – ER – діаграма сутностей

1.5.2 Структура таблиць

Зважаючи на наведені вище структурні особливості програмного комплексу проведено проектування інформаційних таблиць для роботи з програмою . У таблиці 1.2 представлено структуру таблиці БД з інформацією про адміністраторів системи.

Таблиця 1.2 – Структура таблиці з інформацією про адміністраторів системи

Ім'я поля	Тип поля	Довжина	Призначення
idadministrator	int	11	Ідентифікатор адміністратора
Name	varchar	45	Ім'я адміністратора
Lname	varchar	45	Прізвище адміністратора
Login	varchar	45	Логін адміністратора (імя для входу)
Password	varchar	45	Пароль
Email	varchar	45	Емейл адміністратора

Таблиця має назву administrator, містить унікальний ідентифікатор адміністратора, логін адміністратора та пароль для входу в систему , ім'я адміністратора, прізвище адміністратора та його електрону пошту .

У таблиці 1.3 представлено структуру таблиці для збереження інформації про товари які знаходяться в системі.

Таблиця 1.3 – Структура таблиці з інформацією про товари системи

Ім'я поля	Тип поля	Довжина	Призначення
Idgoods	int	11	Ідентифікатор товару
Name	varchar	100	Назва товару
Price	double		Ціна товару
Year	date		Дата виготовлення
Location	text	11	Місце знаходження
description	text	11	Опис про товар

Продовження таблиці 1.3

Ім'я поля	Тип поля	Довжина	Призначення
serial_number	int	255	Серійний номер
Status	int	11	Статус товару
Category	text		Категорія товару
arrival_date	date		Дата прибуття

Таблиця має назву goods, містить унікальний ідентифікатор товару , назву товару , ціну товару , дату випуску товару , ,опис для детальної інформації про товар, серійний номер товару , статус .

Токаж потрібно спроектувати таблицю яка містить список товари які покупець бажає придбати.

Таблиця 1.4 – Структура таблиці з інформацією про товари які містяться в корзині системи

Ім'я поля	Тип поля	Довжина	Призначення
Idpurchase	int	11	Ідентифікатор товару
Name	int	11	Ім'я товару
Price	int	11	Ціна товару
year	date		Дата вироблення товару
Category	varchar	45	Категорія товару
serial_number	int	255	Серійний номер
Country	varchar	45	Країна виробника

Таблиця меє назву purchase , містить унікальний ідентифікатор корзини в якій містяться товари які занесені в неї для оформлення замовлення , а саме назва товару та його ціна , рік та унікальний ідентифікатор продавця для того щоб в корзині відображалися тільки ті товари які даний продавець додав, .

Таблиця 1.5 – Структура таблиці з відомостями стосовно покупки системи

Ім'я поля	Тип поля	Довжина	Призначення
idall_purchase	int	11	Ідентифікатор покупки
Idgoods	int	11	Ідентифікатор товару
Iduser	int	11	Ідентифікатор продавця
Date	date		Дата покупки
<u>goods_name</u>	varchar	100	Ім'я товару

Таблиця має назву all_purchase , міститиме унікальний ідентифікатор покупок та ідентифікатори товару і продавця , дату продажу товару , назву товару та його ціну для того щоб адміністратор відслідковував що коли і ким було продано.

Таблиця 1.6 – Структура таблиці з інформацією про продавців системи

Ім'я поля	Тип поля	Довжина	Призначення
Idusers	int	11	Ідентифікатор продавця
Name	varchar	45	Ім'я продавця
Lname	varchar	45	Прізвище продавця
Login	varchar	45	Логін продавця (ім'я для входу)
Password	varchar	45	Пароль
Email	varchar	45	Емейл продавця
Location	text		Місце знаходження

Таблиця має назву users, міститиме унікальний ідентифікатор продавця, логін продавця та пароль для входу в систему , ім'я та прізвище продавця і електрону пошту .

1.5.3 Побудова схеми реляційної бази даних

У попередньому розділі я виділив сутності та встановив відношення між цими сутностями.

Повна схема реляційної бази даних зі всіма представленнями, таблицями, полями та зв'язками я представив далі на рисунку 1.7.

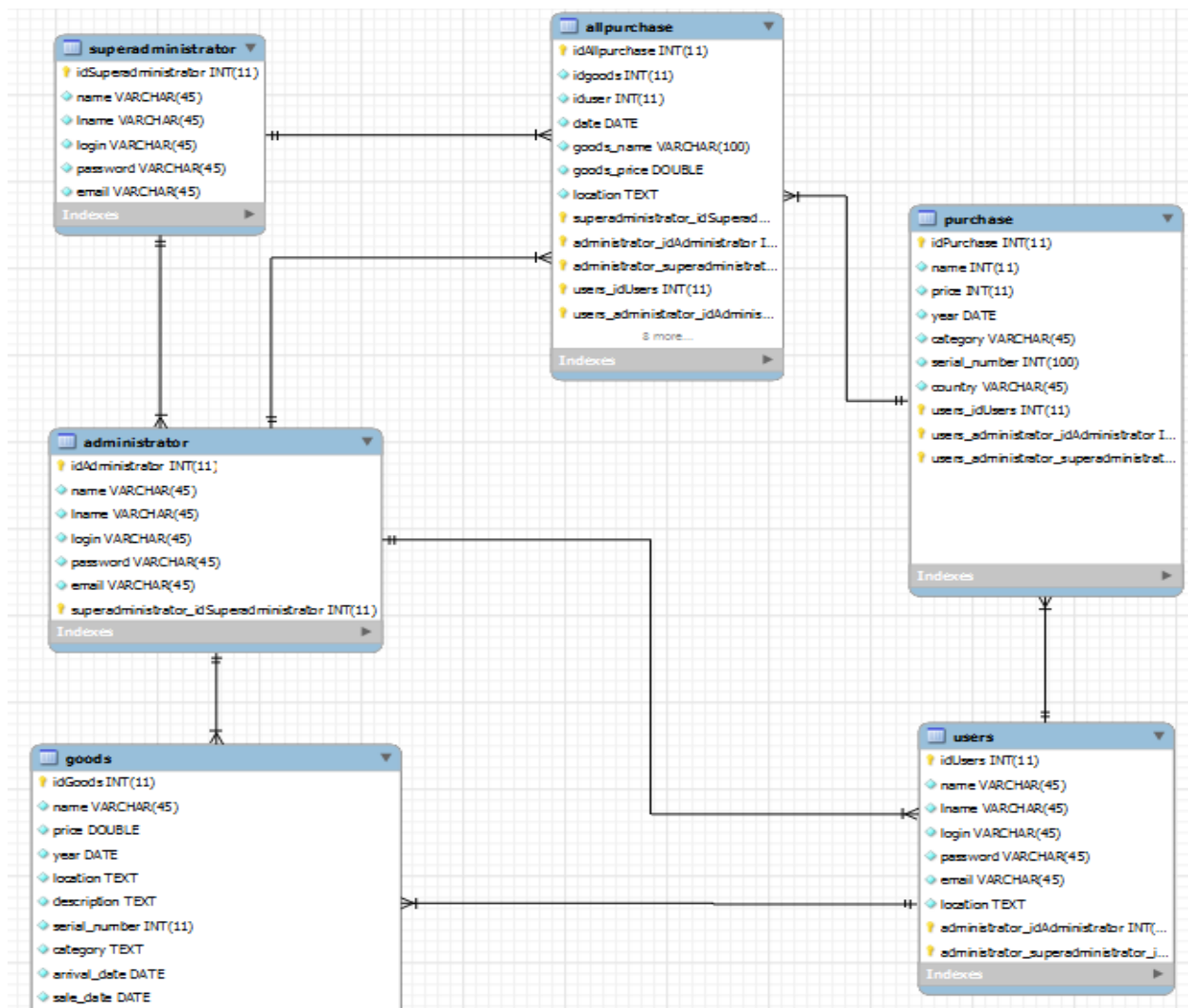


Рисунок 1.7 – Схема реляційної бази даних

1.6 Реалізація основних класів та методів

Класами призначеними для роботи із базою даних визначено слідуєчі класи: Superadministrator Admin, User, Goods, Purchars Connection. Які містять наступні методи:

- AddGood – отримує дані про товари і заносить їх до бази даних;
- UpdateGoods – відповідно до отриманих даних редагує таблицю Goods;
- Pos – отримує параметри пошуку і здійснює за ними вибірку із бази даних;
- DelGoods – отримує id товару яке потрібно видалити і видаляє його;
- AddUser – отримує дані про користувача і заносить їх до бази даних;
- Pos - отримує параметри пошуку і здійснює за ними вибірку із бази даних;
- DelUser – отримує id користувача якого потрібно видалити і видаляє його;
- Purchars – при додаванні товару до корзини дані про товари поміщаються в таблиці Purchars;
- Allpurchars – при оформленні замовлення дані про покупку поміщаються в таблицю Allpurchars;

Метод Connection використовується для з'єднання із базою даних. В даному методі оголошується стрічка MyConString в яку і занесуться необхідні параметри для з'єднання, а саме: в параметрі DRIVER містяться параметри драйвера бази даних, в SERVER міститься адрес сервера, DATABASE містить назву сервера в якому зберігаються таблиці, в параметрах UID і PASSWORD міститься відповідно ім'я користувача і пароль.

```
class Conection
{
```

```

        public static OdbcCommand Conn()
        {
            string MyConnectionString = "DRIVER={MySQL ODBC 3.51
Driver};" +
                "SERVER= host-5-58-80-83.la.net.ua;" +
                "DATABASE= magazin;" +
                "UID=root;" +
                "PASSWORD=1111111111;" +
                "OPTION=3";

            OdbcConnection MyConnection = new
OdbcConnection(MyConnectionString);
            MyConnection.Open();
            OdbcCommand MyCommand = new OdbcCommand("",
MyConnection);
            return MyCommand;
        }

```

Лістинг 1.1 – Підключення до бази даних

Метод Pos міститься в класі Goods, отримує параметри пошуку з форми і здійснює за цими параметрами вибірку безпосередньо із бази даних. Цей метод особливий тим, що він формує SQL запит відповідно до отриманих ним параметрів, тобто якщо один чи декілька параметрів були не задіяні в формі пошуку то SQL-запит не включатиметься. Це стало реалізовним за рахунок перевірок, лістинги яких я наведу нижче по тексту.

```

private void pos_good_Click(object sender, EventArgs e)
{
    dataGridView1.Rows.Clear();
    dataGridView1.Columns.Clear();

    OdbcDataReader MyDataReader;
    MyDataReader = Goods.Pos(textBox1.Text,
textBox2.Text, textBox3.Text, textBox4.Text, textBox8.Text,
textBox7.Text, textBox6.Text);
}

```

```

        foreach (string col in Goods.OutGood_Colums())
        {
            dataGridView1.Columns.Add("", col);
        }

        while (MyDataReader.Read())
        {
            dataGridView1.Rows.Add(MyDataReader.GetInt32(0),
MyDataReader.GetString(1), MyDataReader.GetString(2),
MyDataReader.GetString(3), MyDataReader.GetString(4),
MyDataReader.GetString(5), MyDataReader.GetString(4));

        }
    }
}

```

Лістинг 1.2 – Пошук товару

AddGood – отримання даних стосовно товару і заносить їх до бази даних. Даним методом підтримуються такі параметри, як: назву товару, ціну, дату виготовлення, країну виробника, відомості стосовно товару.

```

class Goods
{
    public static void AddGood(string name, string price,
string year, string location, string description, string
serial_number, string status, string arrival_date, string
sale_date, string country, string category)
    {
        OdbcCommand MyCommand;
        MyCommand = Conection.Conn();
        MyCommand.CommandText = "INSERT INTO
magazine.goods(name,price,year,
location,description,serial_number,status,arrival_date,sale_date,cou
ntry,category) VALUES('" + name + "','" + price + "','" + year +
 "','" + location + "','" + description + "','" + serial_number +

```

```

"', '" + status + "', '" + arrival_date + "', '" + sale_date + "', '" +
country + "', '" + category + "')";
        MyCommand.ExecuteNonQuery();
    }

```

Лістинг 1.3 – Додавання товару

UpdateGood – відповідно до отриманих даних редагує таблицю Goods. Даний метод отримує редаговані дані з активної стрічки таблиці користувача. І з отриманих даних формується SQL-запит на редагування всіх табличних полів. В результаті поля, які користувач змінював в своїй таблиці змінюються і в базі даних, а поля яких користувач не змінював також підлягають редагуванню але ці зміни не помітні для користувача.

```

public static void UpdateGood(string id, string name, string
price, string year , string location, string description, string
serial_number, string status, string arrival_date, string sale_date,
string country, string category)
{
    OdbcCommand MyCommand;
    MyCommand = Connection.Conn();
    MyCommand.CommandText = "UPDATE magazin.goods SET name
= '" + name + "', price = '" + price + "', year = " + year + ,
location = '" + location + "', description = '" + description +
"', serial_number = '" + serial_number + "', status = '" + status
+ "', arrival_date = '" + arrival_date + "', sale_date = '" +
sale_date + "', country = '" + country + "', category = '" + category
+ "' WHERE idgoods = " + id;
    MyCommand.ExecuteNonQuery();
}

```

Лістинг 1.4 – Редагування товару

DelUser – отримує id користувача якого необхідно видалити і видаляє його. Даний метод отримує id користувача з активної стрічки таблиці користувача. При

виклику цього метода реалізується SQL-запит на видалення стрічки з вказаним id. Дана властивість доступна адміністратору і для того щоб її застосувати потрібно спочатку вибрати користувача, і натиснути кнопку «видалити користувача».

```
public static void DelUser(string id)
{
    OdbcCommand MyCommand;
    MyCommand = Connection.Conn();
    MyCommand.CommandText = "DELETE FROM
magazin.users where idusers = " + id;
    MyCommand.ExecuteNonQuery();
}
}
```

Лістинг 1.5 – Видалення користувача

AddUsers – отримує дані про продавця і заносить їх в базу даних. Даний метод отримує такі параметри, як: ім'я продавця, прізвище продавця, логін і пароль, e-mail.

```
public static void AddUser(string name, string lname, string
login, string password)
{
    OdbcCommand MyCommand;
    MyCommand = Connection.Conn();
    MyCommand.CommandText = "INSERT INTO
magazin.users(name,lname,login,password) VALUES ('" + name +
"', '" + lname + "', '" + login + "', '" + password + "')";
    MyCommand.ExecuteNonQuery();
}
```

Лістинг 1.6 – Додавання нового продавця

OutGood – отримує дані про товар з бази даних і виводить ці дані на екран. Даний метод отримує такі параметри, як назва товару , ціна на товар, країну виробника , категорію товару , серійний номер.

```
private void OutTableGood()
{
    OdbcDataReader MyDataReader;
    MyDataReader = Goods.OutGood();
    foreach (string col in Goods.OutGood_Columns())
    {
        dataGridView1.Columns.Add("", col);
    }
    while (MyDataReader.Read())
    {
        dataGridView1.Rows.Add(MyDataReader.GetInt32(0),
                                MyDataReader.GetString(1),
                                MyDataReader.GetString(2),
                                MyDataReader.GetString(3),
                                MyDataReader.GetString(4),
                                MyDataReader.GetString(5),
                                MyDataReader.GetString(6),
                                MyDataReader.GetString(7),
                                MyDataReader.GetString(8),
                                MyDataReader.GetString(9),
                                MyDataReader.GetString(10),
                                MyDataReader.GetString(11));
    }
}
```

Лістинг 1.7 – Виведення товарів

OutUsers – отримує дані про користувача з бази даних і виводить ці дані на екран. Даний метод отримує такі параметри, ім'я продавця , прізвище продавця, логін, пароль, а також e-mail .

```
private void OutTableUser()
{
    OdbcDataReader MyDataReader;
    MyDataReader = User.OutUser();
```

```

        foreach (string col in User.OutUser_Colums())
        {
            dataGridView1.Columns.Add("", col);
        }
        while (MyDataReader.Read())
        {
            dataGridView1.Rows.Add(MyDataReader.GetInt32(0),
MyDataReader.GetString(1), MyDataReader.GetString(2));
        }
    }
}

```

Лістинг 1.8 – Виведення користувача

Форма для AddGood – містить поля для додавання нового товару. Даний метод отримує параметри про товар, його ціну, назву, країну виробника, категорію товару, а також серійний номер.

```

public partial class AddGood : Form
{
    public AddGood ()
    {
        Initialize Component ();
    }
    private void button2_Click(object sender, EventArgs e)
    {
        Goods.AddGood(textBox1.Text, textBox2.Text,
textBox3.Text, textBox4.Text, textBox5.Text, textBox6.Text,
textBox7.Text, textBox8.Text, textBox9.Text, textBox10.Text,
textBox11.Text, textBox12.Text);
        MessageBox.Show(" Товар успішно добавлено! ");
        this.Close();
    }
}

```

Лістинг 1.9 – Форма для додавання товару

```

public partial class Vxid : Form
{
    public Vxid ()

```



```

{
    InitializeComponent ();
}
private void button1_Click (object sender, EventArgs e)
{
    {
        OdbcDataReader MyDataReader;
        MyDataReader = User.OutUser();
        int q = 0;
        while (MyDataReader.Read())

        if (textBox1.Text == MyDataReader.GetString(3))
        {
            if (textBox2.Text == MyDataReader.GetString(4))
            {
                Form1 fl = new Form1();

                Hide();
                label1.Text = "";

                fl.ShowDialog();
                Show();
                q++;
            }
        }
    }
}

```

Лістинг 1.10 – Форма входу

2 ТЕСТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ

Якщо говорити в загальному, то тестування є процесом в ході якого здійснюється перевірка програмного продукту при використанні тестів. Метою є виявити помилки, неточності які могли бути допущені розробниками [34-36].

Розробка програмного продукту може розподілятися по етапам, кожен з яких важливо моніторити та, по можливості, корегувати процес з врахуванням змін, на предмет мінімізації можливих помилок, зрештою це стосується розробки коду програми; розробки інтерфейса програми; тестування програми в цілому.

До основних фаз процесу розробки програмного продукту процес тестування є важливим, характеризується достатньо великою цінністю при створенні продукту. З технічної ж точки зору суттю процесу тестування є відлагодження та оптимізація програмного коду.

Отже, підсумовуючи, тестування виконує дві основні задачі:

- а) демонстрація якості функціонування програмного забезпечення;
- б) знаходження і усунення помилок.

2.1 Вибір середовища тестування

Оскільки в дипломній роботі був розроблений програмний продукт призначення якого є керування сервісом торгівельної мережі, то його тестувати доцільно за допомогою автоматизованих тестів та вручну.

2.1.1 Тестування ПЗ

Тестування в ручному режимі є процесом виявлення дефектів роботи програми, при кому тестувальник перевіряє працездатність усіх компонент програми, начебто він є користувачем. Для точності перевірки, часто тестувальники можуть використовувати наперед розроблений план тестування, в якому відмічають найбільш важливі, на їхню думку, моменти роботи програми.

В цілому ж, для при тестуванні можуть використовуватись статичний (перевірка синтаксису та інших особливостей коду) та динамічний підходи (включає запуск програмного забезпечення).

Також тестування може бути функціональним (коли перевіряється робоча область програмного забезпечення) та не функціональним (коли перевіряється продуктивність, сумісність та безпека тестової системи).

Для опрацювання різного роду виключень я використовував так звані елементи «валідатори», які передбачають ситуацію заповнення неправильними даними форм користувачем.

Наприклад, форма входу в систему містить поля, які обов'язкові для заповнення і повинні містити ідентичну інформацію з даними які знаходяться в базі даних серед них логін і пароль користувача . Для того, щоб уникнути аварійного завершення або незрозумілих дій програмної системи у разі якщо користувач ввів невірний пароль чи логін, необхідно здійснити процес перехвату помилки яку зображено у лістингу 2.1

```
private void button1_Click(object sender, EventArgs e)
{
    OdbcDataReader MyDataReader;
```

```

MyDataReader = User.OutUser();

int q = 0;

while (MyDataReader.Read())

    if (textBox1.Text == MyDataReader.GetString(4))
    {
        if (textBox2.Text == MyDataReader.GetString(5))
        {
            Form1 f1 = new Form1();
            Hide();
            label1.Text = "";

            f1.ShowDialog();
            Show();
            q++;
        }
    }

MyDataReader = Admin.OutAdmin();

    while (MyDataReader.Read())
        if (textBox1.Text == MyDataReader.GetString(3))
if (textBox2.Text == MyDataReader.GetString(4))
    {
        Administrator f2 = new Administrator();
        Hide();
        label1.Text = "";

        f2.ShowDialog();
        Show();
        q++;
    }
}

```

```

        if (q == 0) { label1.Text = "Даної комбінації  

лоніна і пароля неіснує!"; }

    }

```

Лістинг 2.1 – Перехват помилки

Завдяки перехвату помилки якщо користувач системи введе неправильні дані для перевірки, програма не буде некоректно працювати і користувач отримає відповідне повідомлення, приклад якого зображено на рисунку 2.1

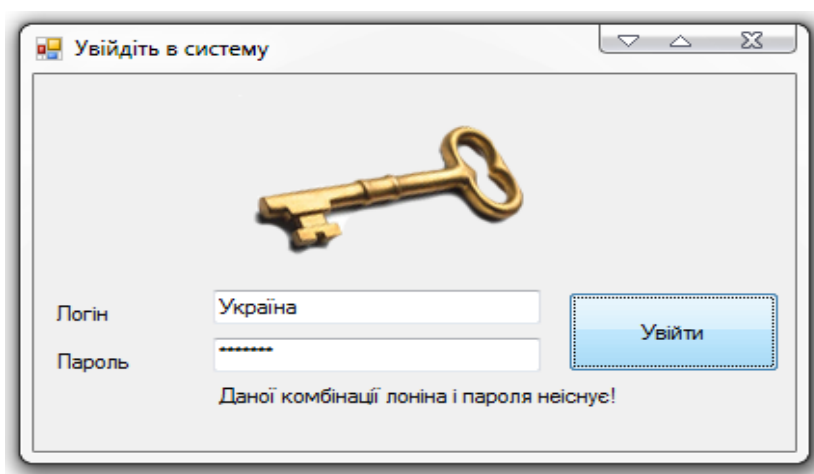


Рисунок 2.1 – Повідомлення про помилку

Виконуючи послідовно дії, були отримані наступні результати:

- при спробі увійти в систему адміністрування був неправильно введений пароль. У відповідь на це було видане повідомлення: Невірні логін/або пароль;
- після чого ввівши повторне введення, який не містив помилкових даних і адміністратор потрапив на сторінку Адміністрування, після чого був вибраний розділ додавання товару;
- було проведено тестування на додавання нового товару, по закінченні тестування було досліджено що при додаванні товару слід заповнити всі поля, при додаванні нового товару було незаповнене поле назва товару і після натиснення кнопки додати програма аварійно закривалася, для усунення

цієї помилки здійснити процес перехвату помилок і при незаповненні поля користувачу програми виведеться повідомлення заповніть всі поля;

- на наступній функції переглянути звітність, відображаються товари які були продані продавцем;
- була перевірена можливість видалення товару , редагування товару , додавання нового подавця , видалення продавця замовлення.

Таким чином, в режимі адміністрування були перевірені всі функції адміністратора , аналогічно провели тестування функціоналю продавця серед яких :

- пошук товару;
- додавання товару в корзину;
- видалення товару з корзини;
- оформити замовлення.

Визляд головного вікна старшого адміністратора та головного вікна адміністратора зображено на рисунку 2.2 і 2.3

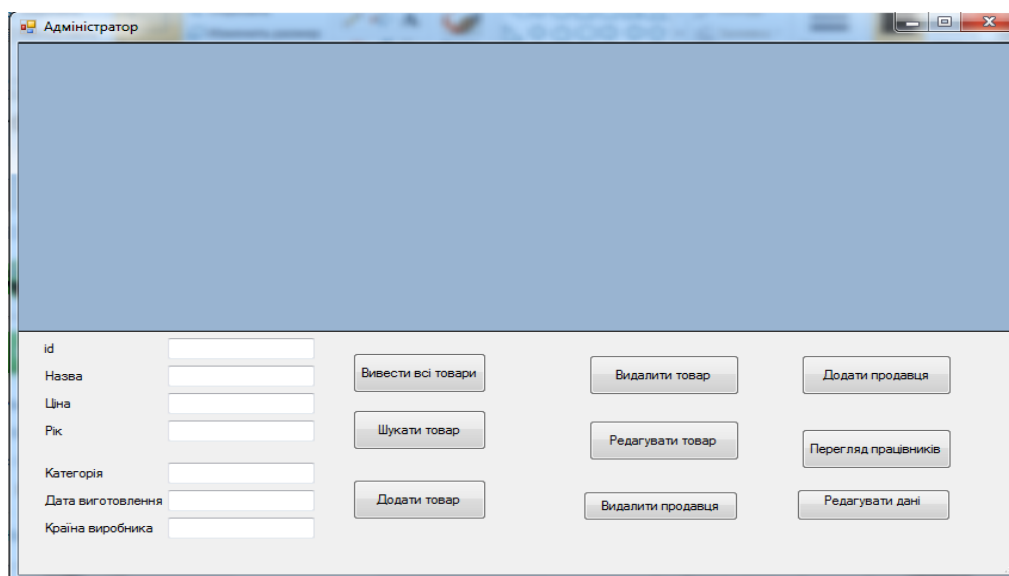


Рисунок 2.2 – Головне вікно адміністратора

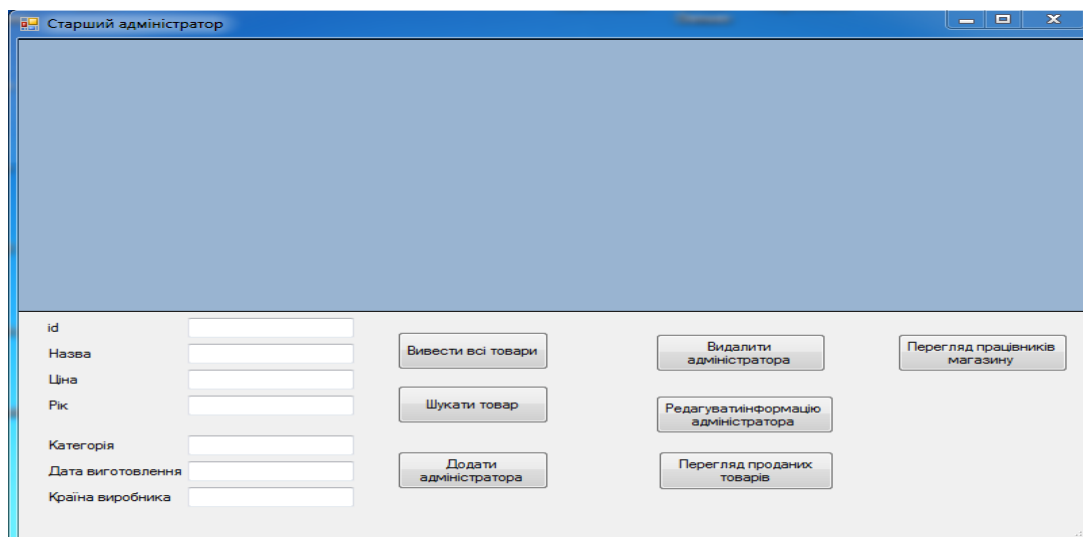


Рисунок 2.3 – Головного вікно адміністратора

При перевірці функціоналу продавця всі функції були протестовані . Таким чином при тестуванні виявлені помилки були усунені , тестування пройдено успішно.

2.1.2 Автоматизоване тестування програмного продукту

Відомо про два основні підходи для автоматизації тестування, а саме тестування на рівні кода (модульне тестування) і тестування інтерфейса користувача (імітування дій користувача при допомозі спеціальних тестових фреймворків).

Найбільше розповсюджена форма автоматизації – це тестування програм засобами графічного інтерфейсу користувача. Популярність цього вида тестування можна пояснити двома факторами: 1 - програму тестують тим самим способом, яким буде користуватись користувач; 2 – можливе тестування програмиине володіючи, при цьому, доступом до вихідного коду.

Одна з головних проблем автоматизованого тестування полягає в його трудоемкості. Не дивлячись на те, що такий вид тестування дає можливість

усувати частину рутинних операцій, а також прискорити виконання тестів, також значні ресурси можуть затрататись для оновлення безпосередньо самих тестів. І важливо, що це стосується обидвох видів автоматизації.

Однак, проблемою є те, що автоматичні тести точно не зможуть в повній мірі замінити ручне тестування. Проте, із-за того, що автоматизація всіх випробувань є достатньо дорогим процесом, тому слугує лиш як доповнення до ручного тестування.

Для автоматизованого тестування я прийняв рішення обрати performance and load testing/тестування навантаженням за допомогою якого можна визначити масштабовність додатків під навантаженням. Під час цього тестування :

- виконується замір часу потрібного для здійснення обраних операцій за певних інтенсивностей їх виконання;
- визначається кількість користувачів, які одночасно працюють із додатком;
- визначаються прийнятні межі продуктивності при умовах збільшення навантаження, тобто при підвищенні інтенсивності виконання операцій;
- досліджується продуктивність при високих та граничних, а також стресових (екстримальних) навантаженнях.

Для проведення тесту потрібно відкрити майстер тестового навантаження, вибрати крок навантаження, змінити шаблон навантаження на покрокове, щоби, з часом, поступово збільшити кількість користувачів.

Далі, після створення тесту, потрібно його додати до списку виконуваних тестів.

Після завершення роботи з майстром тестового навантаження, тест продуктивності додаю в навантажувальний тест, що і відображається засобами редактора тестового навантаження. при виконанні тесту є можливість побачити, що тест дійсно перебуває в черзі на обробку.

Проходження процесу виконання тесту відображається на графіку продуктивності. всі помилки можна переглядати, обравши «відображення

докладних відомостей, попередження та інші відомості» що стосуються виконання навантажувального тесту. також на екрані можна побачити звіт з тестування та переглянути віртуального користувача (рис. 2.1).

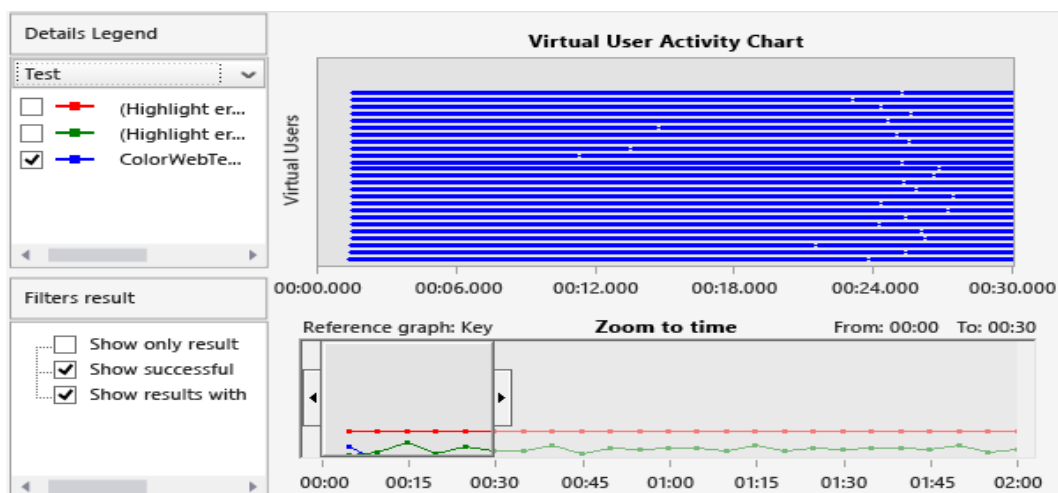


Рисунок 2.1 – Вигляд результату тестування 1-го користувача

Після завершення тестування навантаженням було проведено автоматизований тест навантаження на сервер в якому розміщено базу даних, яке я пройшов успішно.

5 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

5.1.Безпека життєдіяльності

Внаслідок надзвичайних ситуацій, не залежно від виду та напрямку діяльності, можуть виникати руйнування будинків, споруд, шляхів сполучення, зараження місцевості радіоактивними та хімічними речовинами, затоплення, пожежі тощо. Люди можуть опинитися у завалах, у пошкоджених, підтоплених або палаючих будинках, інших не передбачуваних ситуаціях. У зв'язку з цим необхідні заходи з рятування людей, надання їм допомоги, локалізації аварій та усунення пошкоджень.

5.1.2 Ліквідація надзвичайних ситуацій

При вирішенні проблем, при ліквідації надзвичайних ситуацій, виходять з того, що в осередках ураження і районах лиха будуть проводитися не тільки суто рятувальні роботи, а й деякі невідкладні, не пов'язані з рятуванням людей.

Рятувальні та інші невідкладні роботи (РіНР) проводяться з метою:

- порятунку людей та надання допомоги ураженим;
- локалізації аварій та усунення пошкоджень;
- створення умов для наступного проведення відновлювальних робіт.

При проведенні РіНР велике значення має дотримання таких умов, як своєчасне створення угруповань, сил, що залучаються для проведення РіНР; своєчасне ведення розвідки; швидкий рух і введення сил в осередок ураження; безперервне проведення РіНР до їх повного завершення; тверде й оперативне управління силами, що залучаються до проведення РіНР; всебічне забезпечення їх діяльності.

Рятувальні роботи включають:

- розвідку маршрутів висування формувань і об'єктів робіт;
- локалізацію і гасіння пожеж на маршрутах висування і на ділянках робіт;
- пошук уражених і витягування їх з пошкоджених та палаючих будинків, загазованих, затоплених, задимлених приміщень, із завалів;
- розкриття зруйнованих, пошкоджених, завалених споруд та рятування людей, які там знаходяться;
- подання повітря в завалені споруди з пошкодженою фільтровентиляційною системою;
- надання першої долікарської допомоги ураженим та евакуація їх до лікарсь-ких установ;
- виведення (вивезення) населення з небезпечних зон у безпечні райони;
- санітарну обробку людей, ветеринарну обробку сільськогосподарських тва-рин, дезактивацію та дегазацію техніки, засобів захисту, одягу, продовольства, води, фуражу.

Навіть в умовах мирного часу можуть виникати осередки масового ураження надзвичайного характеру, котрі призводять до порушення нормальної діяльності населення, загибелі людей, руйнування та знищення матеріальних цінностей.

Щорічно в Україні виникає близько 1000 важких НС природного та техногенного характеру, котрі призводять до загибелі тисяч людей, а матеріальні збитки сягають кількох мільярдів гривень і складають від 3,2 до 4% внутрішнього валового продукту.

За таких умов поряд з іншими завданнями щодо ліквідації наслідків НС важливе значення мають медичні завдання.

Медичний захист населення, котре мешкає в районі розташування АЕС, складає важливу частину цілого комплексу захисних заходів, котрі реалізуються штабами та службами ЦО у випадку виникнення радіаційне небезпечної аварії на такій станції.

Основною метою таких заходів є зведення до мінімуму кількості опромінених людей та доз їх опромінення, зумовленого знаходженням на радіаційне забрудненій території.

Заходи щодо захисту населення плануються та реалізуються на підставі закону України "Про захист людини від впливу іонізуючих випромінювань" для використання в роботі з питань ЦО та НС за №54/09-1.

Планування вказаних заходів здійснюється органами управління ЦО згідно з вимогами "Типового змісту плану заходів по захисту населення у випадку загальної радіаційної аварії на АЕС".

Згідно з планом захисту населення розробляється "План медичного забезпечення населення у випадку загальної аварії на атомній станції", в якому більш детально висвітлюються завдання місцевих сил цивільної оборони, відповідальність посадових осіб та порядок взаємодії з іншими службами, органами управління ЦО, медично-санітарною частиною, яка забезпечує АЕС.

У випадку виникнення загальної аварії на АЕС, основними заходами місцевих сил цивільної оборони є:

- участь в контролі за рівнями радіації на місцевості;
- порядок видачі хворим та персоналу медичних закладів ЗІЗ та використання ПРУ;
- організація санітарного нагляду за радіаційною безпекою різних груп населення, а також осіб, що приймають участь у ліквідації наслідків аварій на АЕС;
- організація медичного обстеження населення, яке зазнало дію іонізуючого випромінювання та диспансерний нагляд за ними;
- загальні санітарно-гігієнічні та протиепідеміологічні заходи. З усіх заходів, які здійснює МСЦО стосовно населення, яке зазнало дії іонізуючого випромінювання внаслідок аварії на АЕС, найбільш важливим в початковий період після її виникнення є йодна профілактика, яка є достатньо ефективним методом захисту щитовидної залози від дії радіаційних ізотопів, що надходять в організм людини

інгаляційним шляхом. Засоби йодної профілактики в таблетках, а через 1,5—2 місяці після аварії для виведення J132, J134 – ферроцин у вигляді порошків.

Одним з основних завдань цивільної оборони України є організація життєзабезпечення населення у разі виникнення надзвичайної ситуації. Заходи життєзабезпечення здійснюються центральними та місцевими органами державної виконавчої влади структурними підрозділами у їх складі, що безпосередньо відповідають за захист населення, адміністраціями підприємств, установ і організацій з метою задоволення життєвих потреб громадян, які потерпіли від наслідків НС. Головними заходами є:

- надання житла;
- організація харчування;
- забезпечення одягом, взуттям та товарами першої необхідності;
- медичне обслуговування і т. п.

Організація життєзабезпечення населення в екстремальних умовах є комплекс заходів, спрямованих на створення і підтримання нормальних умов життя, здоров'я і працездатності людей.

Всі ці заходи організуються державною виконавчою владою областей, районів, міст, районів у містах, селищ і сіл, органами управління цивільної оборони при чіткому погодженні між ними заходів, що проводяться. Керівники підприємств, установ і організацій є безпосередніми виконавцями цих заходів. Заходи розробляються завчасно, відображаються в планах ІДО і виконуються в період загрози та після виникнення надзвичайної ситуації.

Тимчасове розселення громадян у безпечних районах передбачає максимальний захист людей від радіоактивного забруднення, хімічного ураження при аваріях або катастрофах на радіаційне або хімічно небезпечних об'єктах, а також запобігає загибелі людей у випадках катастрофічного затоплення районів його проживання. В місцях розселення звільняються приміщення для розміщення евакуйованих громадян, готуються (при необхідності) колективні засоби захисту. Якщо сховищ недостатньо, то організовується їх додаткове будівництво,

пристосування існуючих підвалів, гірських виробок, для чого залучається усе працездатне населення, в тому числі і евакуйовані.

Виключно велике значення має забезпечення в місцях розселення евакуйованого населення продуктами харчування, надання їм побутових послуг і медичне обслуговування.

Забезпечення населення продуктами харчування і предметами першої необхідності здійснюється службою торгівлі і харчування ІДО сільського району (району куди евакуйоване населення).

Перші дві доби люди повинні харчуватися запасами продуктів, привезених з собою. При їх відсутності харчування здійснюється через мережу (їдалень) громадського харчування, або в сім'ях, куди вони підселяються.

Харчування особового складу невоєнізованих формувань ЦО здійснюється пересувними пунктами харчування і пересувними пунктами продовольчого постачання служби торгівлі і харчування. Безпосередньо організовують харчування особового складу формувань заступники командирів формувань по матеріально-технічному постачанню.

Особовий склад НФ забезпечується гарячою їжею до виходу в осередках ураження, а потім після санітарної обробки і виводу у відведені місця відпочинку. При відсутності часу (неможливості) на приготування гарячої їжі видається сухий пайок у захисній тарі.

Видача гарячої їжі та сухих пайків проводиться по установленій нормі. Приготування та прийом гарячої їжі організується на незараженій території, або у місцевості з рівнем радіації, що не перевищує 1 рентген на годину (1 Р/год). При радіації від 1 до 5 Р/год приготування їжі і прийом її здійснюється тільки в закритих приміщеннях. При рівні радіації понад 5 Р/год особовому складу НФ замість гарячої їжі видається сухий пайок у герметичній упаковці.

Для приготування їжі в зонах радіоактивного зараження використовуються в основному консервовані і концентровані продукти у захисній (герметичній) тарі і упаковці, які не потребують складної кулінарної обробки. В районах, заражених отруйними речовинами дозволяється готувати і приймати їжу тільки у

спеціальних приміщеннях, обладнаних фільтровентиляційними установками. В районах, заражених бактеріальними засобами, приготування і вживання їжі дозволяється тільки після ретельної дезінфекції території, кухонь і обладнання, а також повної санітарної обробки потерпілих. Продовольство і воду, заражені радіоактивними речовинами, отруйними речовинами, і бактеріальними (біологічними) засобами використовувати в їжу заборонено. Допустимі рівні сумарного вмісту радіонуклідів у продуктах харчування і питній воді встановлені головним Державним лікарем з урахуванням радіаційної обстановки можуть, мінятися.

Комунально-побутове обслуговування в районах розміщення покладають на місцеві комунально-побутові служби. Кількість їх при необхідності може бути збільшена за рахунок розгортання комунально-побутових служб евакуйованим населенням.

5.2 Основи охорони праці

При виборі теми кваліфікаційної роботи на здобуття освітнього ступеня «бакалавр» я обрав актуальний на сьогодні напрямок, а саме розробку системи прийняття рішень з використанням сучасних інформаційних технологій. При використанні ПЗ, яке є результатом даної розробки, як і при використанні будь-якого іншого ПЗ, необхідно дотримуватися вимог з охорони праці при роботі з ПК. Розгляну основні нормативні документи, в яких зазначені вимоги до робочих місць та приміщень при використанні ПК [40,41].

Насамперед розглянемо вимоги до приміщень, де буде знаходитися робоче місце. Ці вимоги описані в ДСанПін «Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» та НПАОП «Правила охорони праці під час експлуатації електронно-обчислювальних машин».

У відповідності до вимог, площа для одного робочого місця при використанні розробленого ПЗ повинна бути не менше 6,0 м², а об'єм – не менше 20,0 м³, при цьому робочі місця не можуть бути розміщені у підвальних приміщеннях чи на цокольних поверхах. З метою збереження функції зору, в приміщенні повинне бути рівномірне комбіноване освітлення. Згідно ДБН В, природне освітлення повинно здійснюватися через світлові прорізи, які орієнтовані на північ чи північний схід, і забезпечувати коефіцієнт природного освітлення не менше 1,5%. Прорізи повинні мати жалюзі або завіски. Штучне освітлення мають забезпечувати люмінесцентні лампи. Покриття підлоги повинне мати незначний коефіцієнт відбиття, який знаходиться в межах 0,3-0,5. Температура, відносна вологість та рухливість повітря повинні відповідати СН. Також у цих приміщеннях потрібно щодня робити вологе прибирання.

Щодо організації робочого місця, то воно також повинне відповідати вимогам. Насамперед, конструкція робочого місця повинна забезпечувати підтримання оптимальної робочої пози, а конструкція робочого столу має забезпечувати оптимальне розміщення ПК та периферійних пристроїв.

Висота робочого столу, на якому розміщений ПК має знаходитися в межах 680...800 мм, а ширина і глибина – 600...1400 мм і 800...1000 мм відповідно. Стіл також повинен мати достатній простір для ніг, що забезпечить зручну осанку користувача ПК.

Робочий стілець повинен бути підйомно-поворотним, регульованим за висотою, за кутом і за нахилом сидіння та спинки. Регулювання за кожним із параметрів має здійснюватися незалежно, легко і надійно фіксуватися. Висота поверхні сидіння стільця має регулюватися в межах 400...500 мм, а ширина і глибина становити не менше ніж 400 мм. Кут нахилу сидіння – до 15 град. вперед і до 5 град. назад. Висота спинки стільця має становити 300...320 мм, ширина – не менше ніж 380 мм, радіус кривизни горизонтальної площини - 400 мм. Кут нахилу спинки має регулюватися в межах 1...30 град. від вертикального положення. Відстань від спинки до переднього краю сидіння має регулюватися в межах 260...400 мм.

Монітор ПК має розташовуватися на відстані 600...700 мм від очей користувача. Розташування монітору має забезпечувати зручність зорового спостереження у вертикальній площині під кутом +30 град. до нормальної лінії погляду працівника.

Пристрої вводу (клавіатуру та мишку) слід розташовувати на поверхні столу на відстані 100...300 мм від краю поверхні робочого столу. У конструкції клавіатури має бути опорний пристрій (виготовлений із матеріалу з високим коефіцієнтом тертя, що перешкоджає мимовільному її зсуву), який дає змогу змінювати кут нахилу поверхні клавіатури у межах 5...15 град. Висота середнього рядка клавіш має не перевищувати 30 мм. Поверхня клавіатури має бути матовою з коефіцієнтом відбиття 0,4.

Вимоги безпеки при роботі з ПК визначено в НПАОП 0.00-1.28-10. Згідно вимог електробезпеки, ПК повинні підключатися до електромережі тільки за допомогою справних штепсельних з'єднань. Не допускається підключати ПК до звичайної двопровідної електромережі, в тому числі з використанням перехідних пристроїв. Електромережі штепсельних з'єднань та електророзеток для живлення ПК потрібно виконувати за магістральною схемою. При організації робочих місць електромережу штепсельних розеток для живлення ПК у центрі приміщення прокладають у каналах або під знімною підлогою в металевих трубах або гнучких металевих рукавах.

Щодо безпеки при роботі з ПК, щодня перед початком роботи необхідно очищати монітор від пилу та інших забруднень. Після закінчення роботи з ПК, він та периферійні пристрої повинні бути відключені від електричної мережі. У разі виникнення певної аварійної ситуації необхідно негайно відключити ПК від електричної мережі. Не допускається виконувати обслуговування, ремонт та налагодження ПК безпосередньо на робочому місці.

Під час користування розробленим ПЗ, як і при будь-якій іншій роботі за ПК, необхідно робити перерви по 15 хвилин через кожну годину. Якщо такі перерви неможливо проводити, то тривалість роботи з ПК повинна бути не більше, ніж чотири години.

Основні вимоги до пожежної безпеки вказані в НАПБ «Правила пожежної безпеки в Україні». Згідно цих вимог, у приміщеннях, де знаходиться ПК, обов'язково повинні бути переносні вуглекислотні або аерозольно водопінні вогнегасники. Підходи до засобів пожежогасіння повинні бути вільними. Також у приміщенні, де розміщені ПК, робочі місця повинні бути обладнані системою автоматичної пожежної сигналізації з димовим пожежним сповіщувачем.

ВИСНОВКИ

В записці представлено основні етапи виконання кваліфікаційної роботи на тему «Розробка системи прийняття рішень структурної одиниці комерційної мережі з підвищеною продуктивністю роботи з врахуванням парадигм ООП, C#, MVS»

Об'єктом розроблення було обрано систему реалізації з підвищеною продуктивністю роботи структурної одиниці мережі з врахуванням парадигм об'єктно-орієнтованого підходу.

Предметом дослідження виступив ґрунтовний аналіз вибору підходу розробки, зокрема розглянуто об'єктно-орієнтований та процедурний підходи, проаналізовано переваги та їхні недоліки. Для створення схеми бази даних було виокремлено основні сутності, сконструйовано концептуальну модель бази даних, UML діаграми класів, обґрунтовано вибір мови програмування.

Ціллю роботи вважаю використання сучасних ІТ технологій при розробці безпечної системи, яка б оптимізувала та автоматизувала роботу окремої структурної одиниці мережевого підприємства, тим самим вдосконаливши і покращивши результати роботи всієї мережі, та врахування в розробці можливості вирішувати конкретні завдання відповідно до вимог замовника.

В представлений роботі реалізовано програмну систему, яка взаємодіє із базою даних. На основі аналізу предметної області показано застосування та мету створення даного проєкту. Далі описано його функціональні можливості. Створена СУБД була продумана та структурована, а також нормалізована.

Розробка програми здійснювалася з допомогою використання можливостей мови програмування C#. Розробку програми було розбито на створення інтерфейсу, який призначений для взаємодії користувача з системою, та розробку функціонала програми, який відповідатиме за роботу безпосередньо програмної системи. Враховано, необхідність, щоб програмний інтерфейс та програмний функціонал взаємодіяли між собою .

Як засіб для написання програмного продукту використано мову програмування C#, основною перевагою якої є простота розробки таких програм та зручна взаємодія із базою. І як наслідок в якості середовища розробки використано Microsoft Visual Studio.

В якості СУБД використано MySQL Server, оскільки її досить легко можна налаштувати і підключити до системи.

Отже, в результаті роботи програмний продукт пройшов всі етапи розробки. Зокрема можна відмітити етапи аналізу вимог, проектування та розробки. Також був пройдений етап тестування програми. Поставлені завдання були виконані, однак в системі залишаються можливості до вдосконалення і покращення. Зокрема можлива реалізація в майбутньому додаткових функцій, впровадження ідентифікації покупців (клієнтські карти знижок). В майбутнього проєкт можна розширити, збільшивши базу даних та додавши новий функціонал програми такий як інструкція користувача системи.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Дистанційний курс «Кваліфікаційні роботи бакалаврів» сайту дистанційного навчання ТНТУ [Електронний ресурс]. – Режим доступу: URL: <https://dl.tntu.edu.ua/bounce.php?course=5329>.
2. Авраменко В.С., Авраменко А.С. Проектування інформаційних систем: навчальний посібник / В.С. Авраменко, А.С. Авраменко. – Черкаси: Черкаський національний університет ім. Б. Хмельницького, 2017. – 434 с.: іл.
3. Платформа цифрової дистрибуції [Електронний ресурс] – Режим доступу: URL: <https://uk.wikipedia.org/wiki/Origin>
4. Системи штучного інтелекту: навч. посіб. / Н. Б. Шаховська, Р. М. Камінський, О. Б. Вовк. — Львів: Вид-во Львівської політехніки, 2018. — 392 с. — ISBN 966-941-197-6.
5. Системи штучного інтелекту: навч. посіб. / Ю. В. Нікольський, В. В. Пасічник, Ю. М. Щербина; за наук. ред. В. В. Пасічника; М-во освіти і науки, молоді та спорту України. — 2-ге вид. — Львів: Магнолія-2006, 2013. — 279 с.: іл.— ISBN 978-617-57-40-11-4
6. Розробка автоматизованої інформаційної системи обробки даних / Казмірчук В.М., Цуприк Г.Б./ Матеріали VIII Науково-технічної конференції «Інформаційні моделі, системи та технології» – Тернопіль 9-10 грудня 2020. — Т. : ТНТУ, 2020. — С. 148.
7. Богуш Б.М., Довидков О .А. Проектування захищених інформаційних систем і мереж. -К.: ДУІКТ, 2006. - 414 с.
8. Дискретні структури (Алгебраїчні та числові системи, комбінаторний аналіз) : навчально-методичний посібник для студентів спеціальності 121 «Інженерія програмного забезпечення», аспірантів та викладачів вищих навчальних закладів / Укладач : Бойко І.В., Петрик М.Р., Цуприк Г.Б. – Тернопіль : Тернопільський національний технічний університет імені Івана Пулюя, 2017 – 64 с.

9. Моделювання та видобуток даних (висопродуктивні обчислення у великих алгебраїчних та числових системах, комбінаторному аналізі): навчальний посібник. Тернопіль: ТНТУ 2019 – 62 с.

10. Інформаційні технології видобутку даних (Data mining, високопродуктивні обчислення у складних системах): навчальний посібник ІВ Бойко, МР Петрик, Г Цуприк – 2020

11. Цуприк Г.Б. Розробка розумної системи обліку роботи комп'ютерної мережі підприємства в середовищі Embarcadero RAD Studio XE / Тимченко Є, Цуприк Г. /Матеріали X Науково-технічної конференції «Інформаційні моделі, системи та технології» – Тернопіль 7 грудня 2022. — Т. : ТНТУ, 2022. — С. 133.

12. Цуприк Г.Б. Розробка програмної підсистеми алгоритмічного перетворення даних з використанням різномірних обчислювальних засобів мовою С / О.В. Михайлівський, Г.Б.Цуприк / Матеріали XI Науково-технічної конференції «Інформаційні моделі, системи та технології» – Тернопіль 13-14 грудня 2023. — Т. : ТНТУ, 2023. — С. 216.

13. Розробка системи забезпечення безпечного обміну інформацією з використанням технологій веб програмування / Доскоч Н., Цуприк Г. / Матеріали X Науково-технічної конференції «Інформаційні моделі, системи та технології» – Тернопіль 7 грудня 2022. — Т. : ТНТУ, 2022. — С. 114.

14. Середовище розробки Microsoft Visual Studio [Електронний ресурс] – Режим доступу: URL: <https://learn.ztu.edu.ua/mod/page/view.php?id=9974>

15. [Електронний ресурс] – Режим доступу: URL: <https://software.com.ua/uk/programuvannya-microsoft/109-microsoft-visual-studio-professional-2022.html>

16. Microsoft visual studio [електронний ресурс] – режим доступу: url: <https://macrosoft.store/uk/blog/post/29-для-чого-потрібна-microsoft-visual-studio>

17. [Електронний ресурс] – Режим доступу: URL: «Visual Studio .NET» та інші продукти для розробників

18. [Електронний ресурс] – Режим доступу: URL: Розділ «Microsoft Visual Studio» на сайті Майкрософт

19. Бйорн Страуструп . Мова програмування C++. Короткий курс - 2019. - 320 с. - ISBN 978-5-907144-12-5 .
20. Бйорн Страуструп . Програмування: принципи та практика з використанням C++. - 2016. - 1328 с. - ISBN 978-5-8459-1949-6 .
21. [Електронний ресурс] – Режим доступу: URL: https://uk.wikipedia.org/wiki/C_Sharp
22. Bruce Eckel (2006). Thinking in Java (4th Edition). Prentice Hall PTR. ISBN 978-0131872486.
23. Joshua Bloch (2008). Effective Java (2nd Edition). Prentice Hall PTR. ISBN 978-0321356680.
24. James Gosling; Bill Joy, Guy Steele, Gilad Bracha (2005). The Java Language Specification, Third Edition. Addison-Wesley. ISBN 0-321-24678-0.
25. Silberschatz, Abraham; Sudarshan, S. (2011). Database system concepts (вид. 6). New York: McGraw-Hill. ISBN 9780073523323. OCLC 436031093.
26. [Електронний ресурс] – Режим доступу: URL: <https://www.mysql.com/products/workbench/>
27. [Електронний ресурс] – Режим доступу: URL: <https://www.mysql.com/>
28. [Електронний ресурс] – Режим доступу: URL: Підручник з Umbrello UML Modeller
29. Shaw, Mary; DeLine, Robert; Klein, Daniel V.; Ross, Theodore L.; Young, David M.; Zelesnik, Gregory (1995-04). Abstractions for Software Architecture and Tools to Support Them (PDF). IEEE Trans. Softw. Eng. **21** (4): 314—335. doi:10.1109/32.385970
30. Foundations for the study of software architecture // Software Engineering Notes. — ACM SIGSOFT^[en], 1992. — Vol. 17, iss. 4 (5 June). — P. 40-52.
31. [Електронний ресурс] – Режим доступу: URL: <https://www.artofba.com/uk/post/main-types-of-software-architecture>
32. James Rumbaugh, Ivar Jacobson, Grady Booch (1999). The unified modeling language reference manual (англ.) . Addison Wesley Longman Inc. ISBN 0-201-30998-X.

33. Fowler (2003) UML Distilled: A Brief Guide to the Standard Object Modeling Language

34. [Електронний ресурс] – Режим доступу: URL: Текст лекцій до курсу «Технології розробки і тестування програм» Дідковська М. В.

35. [Електронний ресурс] – Режим доступу: URL: Quality Assurance Group — Навчання, Перекваліфікація, Підвищення рівня кваліфікацій в сфері Забезпечення Якості, Контролю Якості та Тестування ПЗ

36. [Електронний ресурс] – Режим доступу: URL: Львівська Школа тестування Програмного Забезпечення

37. Михайлюк В.О., Халмурадов Б.Д. Цивільна безпека: Навчальний посібник. – К.: Центр учбової літератури, 2012, - 158 с.

38. Мохняк С.М., Дацько О.С., Козій О.І., Романів А.С., Петрук М.П., Скіра В.В., Васійчук В.О., Безпека життєдіяльності. Навчальний посібник. Львів. Видавництво НУ "Львівська політехніка", 2012.- 264 с.

39. Осипенко С.І., Іванов А.В. "Організація функціонального навчання у сфері цивільного захисту". Навчальний посібник. – К., 2014. – 286с.

40. Дистанційний курс «Основи охорони праці» сайту дистанційного навчання ТНТУ [Електронний ресурс]. – Режим доступу: URL: <http://dl.tntu.edu.ua/index.php>

41. Про затвердження Правил охорони праці під час експлуатації ЕОМ [Електронний ресурс]. – Режим доступу: URL: <http://zakon4.rada.gov.ua/laws/show/z0382-99>

42. Методичні вказівки до виконання дипломної роботи освітнього рівня —бакалавр студентами усіх форм навчання для напряму підготовки 121 —Інженерія програмного забезпечення// Укладачі : Петрик М.Р., Михалик Д.М., Кінах Я.І., Гладь С.В., Цуприк Г.Б. – Тернопіль : Вид-во ТНТУ імені Івана Пулюя, 2016 – 28 с.

ДОДАТКИ

Додаток А Апробація результатів роботи



Міжнародна науково-технічна конференція
**"Фундаментальні та прикладні проблеми
сучасних технологій"**
присвячена 180-річчю з дня народження Івана Пулюя
та 65-річчю з дня заснування
Тернопільського національного технічного університету
імені Івана Пулюя.
ТЕРНОПІЛЬ 2025



І.П. Вовк, канд. екон. наук, доц., А.Й. Матвійшин, канд. техн. наук, доц., Ю.Я. Вовк, канд. техн. наук, доц. ЦИФРОВІЗАЦІЯ ТА СТІЙКІСТЬ ЛАНЦЮГІВ ПОСТАЧАННЯ В УМОВАХ ГЛОБАЛЬНОЇ НЕСТАБІЛЬНОСТІ: ПОЄДНАННЯ ПІДХОДІВ ПРОЄКТНОГО АНАЛІЗУ ТА УПРАВЛІННЯ ЛАНЦЮГАМИ ПОСТАЧАННЯ	184
Р.Л. Голосинський ПРОБЛЕМАТИКА ПРОГРАМНИХ РІШЕНЬ ДЛЯ КЕРУВАННЯ БЕЗПІЛОТНИКАМИ У ВІЙСЬКОВИХ КОНФЛІКТАХ	186
В.І.Гураль, Г.Б.Цуприк, к.т.н., доц. ВИКОРИСТАННЯ СУЧАСНИХ ВЕБ-ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ ІНТЕРНЕТ-МАГАЗИНУ КОМП'ЮТЕРНИХ АКСЕСУАРІВ	188
Т. Деніга – ст. гр. СП-41, д. ф-м. н. проф. М.Р.Петрик РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ НАДАННЯ ФРІЛАНС ПОСЛУГ З ВИКОРИСТАННЯМ ФРЕЙМВОРКІВ REACT ТА NODE.JS	189
Д.А.Дячишин, Г.Б.Цуприк, к.т.н., доц.. ВИКОРИСТАННІ СУЧАСНИХ ІТ ТЕХНОЛОГІЙ ПРИ РОЗРОБЦІ СИСТЕМ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ОБРОБКИ ДАНИХ КОМЕРЦІЙНОГО ЗНАЧЕННЯ	190
Р.З. Золотий, к.т.н., доцент, Д.А. Коломийчук ЕФЕКТИВНІСТЬ АТАК МЕРЕЖ МАШИННОГО НАВЧАННЯ	192
Д.В. Коваль, Г.Б. Цуприк, к.т.н., доц.. МОДУЛЬНА РЕАЛІЗАЦІЯ ФУНКЦІОНАЛУ ФІНАНСОВОГО КОНТРОЛЮ НА ANDROID ІЗ ЗАСТОСУВАННЯМ ПРИНЦИПІВ ІНВЕРСІЇ УПРАВЛІННЯ ТА ШАБЛОНУ MVVM	193
А.М. Ковтко; В.Б. Савків, к.т.н., доц.; Г.В. Козбур, к.т.н., доц.; І.Р. Козбур АНАЛІЗ ЕФЕКТИВНОСТІ МУТАЦІЙНОГО ТЕСТУВАННЯ	195
А. Конончук – ст. гр. СП-41, док. філософії. І. Мудрик СТВОРЕННЯ УКРАЇНСЬКОЇ МОВНОЇ ГРИ НА ОСНОВІ EMBEDDING-МОДЕЛЕЙ	197
О.І. Крамар, к.ф.-м.н., доц.; Т.О. Крамар МОЖЛИВОСТІ ТРАНСФОРМАЦІЇ ЦИФРОВОГО МУЗЕЮ ІВАНА ПУЛЮЯ В КОНТЕКСТІ ВИКОРИСТАННЯ VR-ГАРНІТУРИ META QUEST	198
Т.О. Крамар, А.В. Мельник ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ СТВОРЕННЯ 3D-МОДЕЛЕЙ: ПЕРСПЕКТИВИ ТА СТАН ДОСЛІДЖЕНЬ	200
В.В. Круцяк, Г.Б. Цуприк, к.т.н., доц.. РОЗРОБКА ПРИКЛАДНОГО СПЕЦІАЛІЗОВАНОГО ЗАСТОСУНКУ ДЛЯ ОПТИМІЗАЦІЇ ОПРАЦЮВАННЯ МАСИВІВ ДАНИХ З ЗАСТОСУВАННЯМ ПАРАДИГМ ТА ТЕХНОЛОГІЙ PYTHON, PYQT5 ТА SQLITE	202
Я. Курко, доцент, О. Босюк ст. викладач, Н. Вальчак ст. викладач, З. Кульчицький, ст. викладач, І. Казмірчук, ст. викладач. ЗАСТОСУВАННЯ КОМП'ЮТЕРНОЇ ПРОГРАМИ "REACTION-TEST" ДЛЯ ВИЗНАЧЕННЯ СТАРТОВОЇ РЕАКЦІЇ СПОРТСМЕНІВ	204
Т. А. Липак, аспірант ІНТЕГРАЦІЯ МЕТОДІВ ШТУЧНОГО ІНТЕЛЕКТУ В РОЗРОБКУ ОРІЄНТОВАНИХ НА КОРИСТУВАЧА ІНТЕРФЕЙСІВ	206

УДК 004.41

Д.А.Дячишин, Г.Б.Цуприк, к.т.н., доц..

Тернопільський національний технічний університет імені Івана Пулюя, Україна

▲ **ВИКОРИСТАННІ СУЧАСНИХ ІТ ТЕХНОЛОГІЙ
ПРИ РОЗРОБЦІ СИСТЕМ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ОБРОБКИ ДАНИХ
КОМЕРЦІЙНОГО ЗНАЧЕННЯ**

D.A.Dyachyshyn, H.B.Tsupryk, Ph.D., Assoc. Prof.

**USING MODERN IT TECHNOLOGIES IN DEVELOPING SYSTEMS TO INCREASE
THE EFFICIENCY OF COMMERCIAL DATA PROCESSING**

Вибір теми роботи пов'язаний саме з тим, що сьогодні відбувається суттєве переформатування та фактично зміна вимог до такого роду програмних продуктів. Пов'язано це з різними причинами і перш за все з тривалим терміном дистанціювання, від ковіду і аж до сьогодні коли вже четвертий рік триває повномасштабне вторгнення

Сьогодні можна спостерігати значним поширенням закладів мульти торгівлі, в яких можна придбати найрізноманітніші товари (Аврора, Та-да, Prostor та інші: від їжі й одягу до товарів для риболовлі та дрібної техніки і т.д.). Такий підхід дещо ускладнює роботу, адже для

забезпечення безперебійної і злагодженої роботи таких закладів потрібна одночасна кропітка робота багатьох працівників, які забезпечують облік товарів та аналіз, вчасне і рівномірне наповнення складу, здійснення швидке оформлення. Крім цього, враховуючи виклики сьогодення, слід враховувати і можливість швидкого перенесення закладу. Не дивлячись на ніби то простоту, така діяльність вимагає заздалегідь продуманих, вчасних, однозначних та зрозумілих розпоряджень від керівництва, компетентності працівників та їх кваліфікованість, досвіду, та й контролю в цілому.

З огляду на вищесказане метою обраної реалізації варта означити пошук, обґрунтування та реалізацію можливості для оптимізації та автоматизації такої з даними та можливості контролю та прийняття рішень навіть віддалено.

Аналізуючи існуючі рішення варта зазначити, на виробництві вже вкрай актуальна та швидко розвивається автоматизація процесів з використанням ІТ. Таку ж мету можна реалізувати і в комерційного спрямування закладах за рахунок автоматизованих інформаційних систем з можливим врахуванням специфіки їх діяльності. В системах такого типу часто застосовують бази даних, в якій міститься уся необхідна інформація. Зв'язок з базою даних по мережі дозволяє одночасну та безперебійну роботу кількох користувачів такої системи, що в свою чергу покращує комунікацію та ефективність роботи. Завдяки автоматизованим процесам, можна відмовитися від значної кількості персоналу, а також суттєво скоротити затрати на обробку значної кількості документів. Збільшується швидкість операції продажу товарів, що дозволяє бачити стан речей «в цілому».

Така система дозволить працівникам виконувати свої функції, зосереджуючись лише на їх безпосередньому завданні. В цілому такі програми є продуктивні і широко поширюються у своїх сферах діяльності. Саме з цієї причини ефективніше замовити таку розробку, акцентуючи увагу на вимоги конкретного замовника, ніж використовувати та пробувати їх допрацьовувати вже існуючі.

Ціллю роботи вважаю використання сучасних ІТ технологій при розробці безпечної системи, яка б оптимізувала та автоматизувала роботу окремої структурної одиниці мережевого підприємства, тим самим вдосконаливши і покращивши результати роботи всієї мережі, та врахування в розробці можливості вирішувати конкретні завдання відповідно до вимог замовника.

Також важливим критерієм такого типу системи повинен бути захист інформації від стороннього втручання. В розроблюваній програмній системі такий захист забезпечується авторизацією, за допомогою якої працівник зможе увійти в систему під своїм логіном і паролем чи при необхідності змінювати його так часто, наскільки це потрібно, таким чином забезпечуючи надійність інформаційної системи.

Розробка програмного продукту здійснювалася за допомогою використання можливостей мови програмування C#. Розробку програми було розбито на створення інтерфейсу, який призначений для взаємодії безпосередньо користувача із самою системою, та розробку функціонала програми, який відповідатиме за роботу безпосередньо програмної системи. Враховано необхідність, щоб програмний інтерфейс та програмний функціонал взаємодіяли між собою.

Як засіб для написання програмного продукту використано мову програмування C#, основною перевагою якої є простота розробки таких програм та зручна взаємодія із базою. І як наслідок в якості середовища розробки використано Microsoft Visual Studio. В якості СУБД використано MySQL Server, оскільки її досить легко можна налаштувати і підключити до системи.

Література

1. Бойко І.В., Петрик М.Р., Цуприк Г.Б. Інформаційні технології видобутку даних (Data mining, високопродуктивні обчислення у складних системах): навчальний посібник. Тернопіль: : ТНТУ 2020 – 62 с.

2. Бойко І.В., Петрик М.Р., Цуприк Г.Б. Моделювання та видобуток даних (високопродуктивні обчислення у великих та числових системах, комбінаторному аналізі): навчальний посібник. Тернопіль: : ТНТУ 2019 – 62 с.

3. Information System for Design of Thin Multilayer Film Processes Parameters Management based on Diffusion M.R., Petryk, Mykhaylo R., V., Chyzh, Vitalii, H.B., Tsupryk, H. B., O.Y., Petryk, Oksana Yu ITTAP'2024: 4th International Workshop on Information Technologies: Theoretical and Applied Problems, October 23- 25, 2024, Ternopil, Ukraine, Opole, Poland, 2024, pp. 486-493 (<https://ceur-ws.org/Vol-3896/>)

4. Methods of constructing algorithms for comparative test statistical verification of mathematical models of bioobject responses to low-intensity stimuli / Bohdan Yavorsky, Evhenia Yavorska, Halyna Tsupryk, Roman Kinash // Scientific Journal of TNTU. — Tern.: TNTU, 2023. — Vol 112. — No 4. — P. 82–90.

Додаток Б Слайди презентації

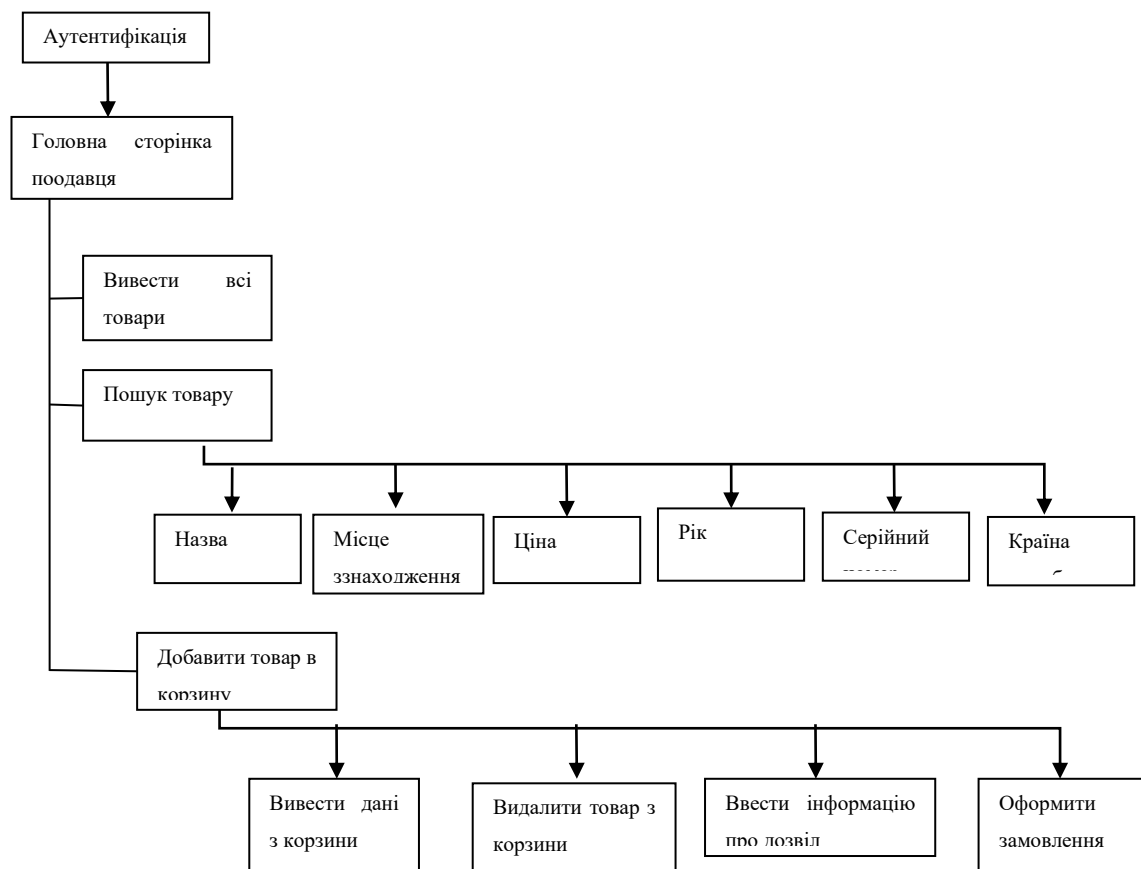


Представлення UML діаграми варіантів використання

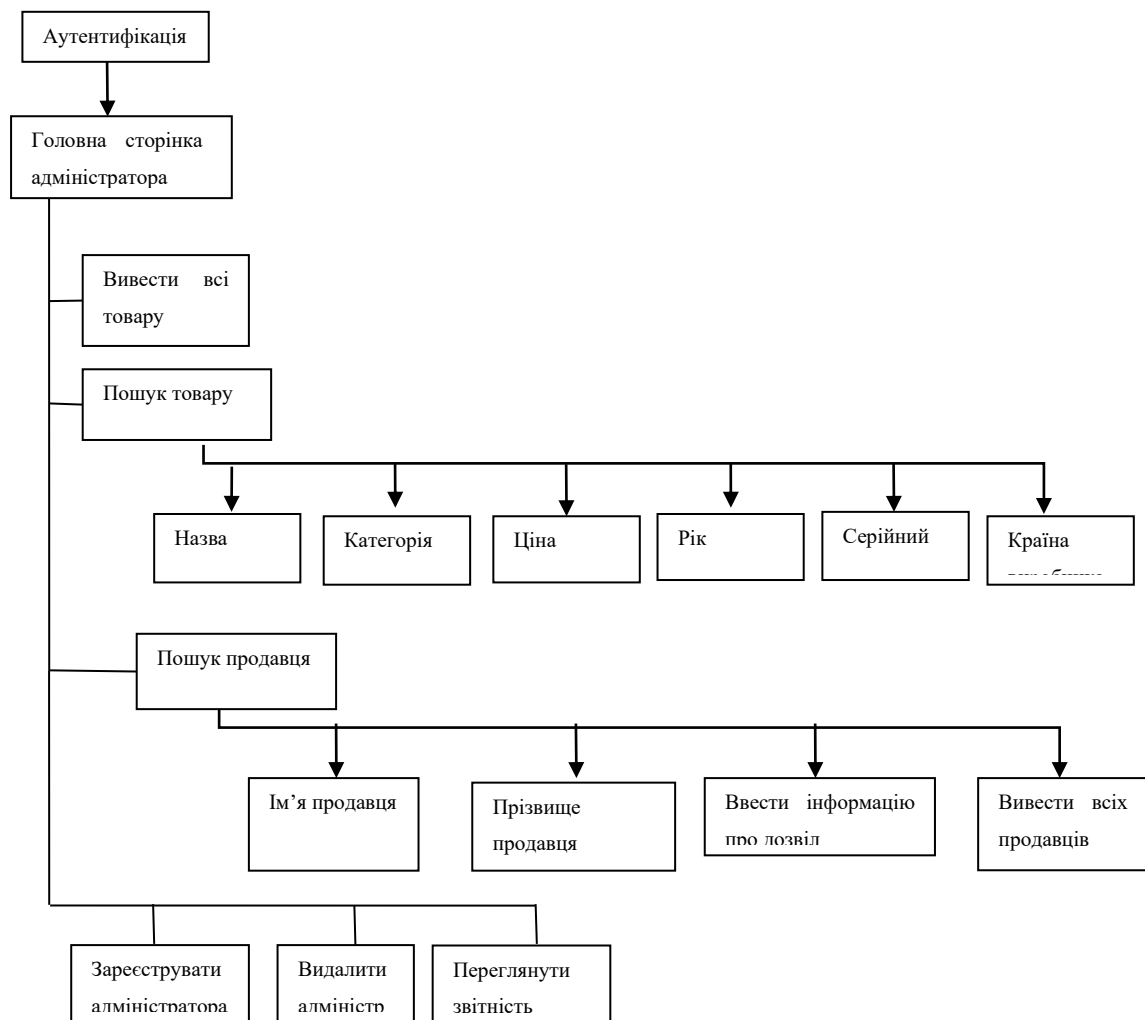


Представлення UML діаграми варіантів використання
(для адміністратора)

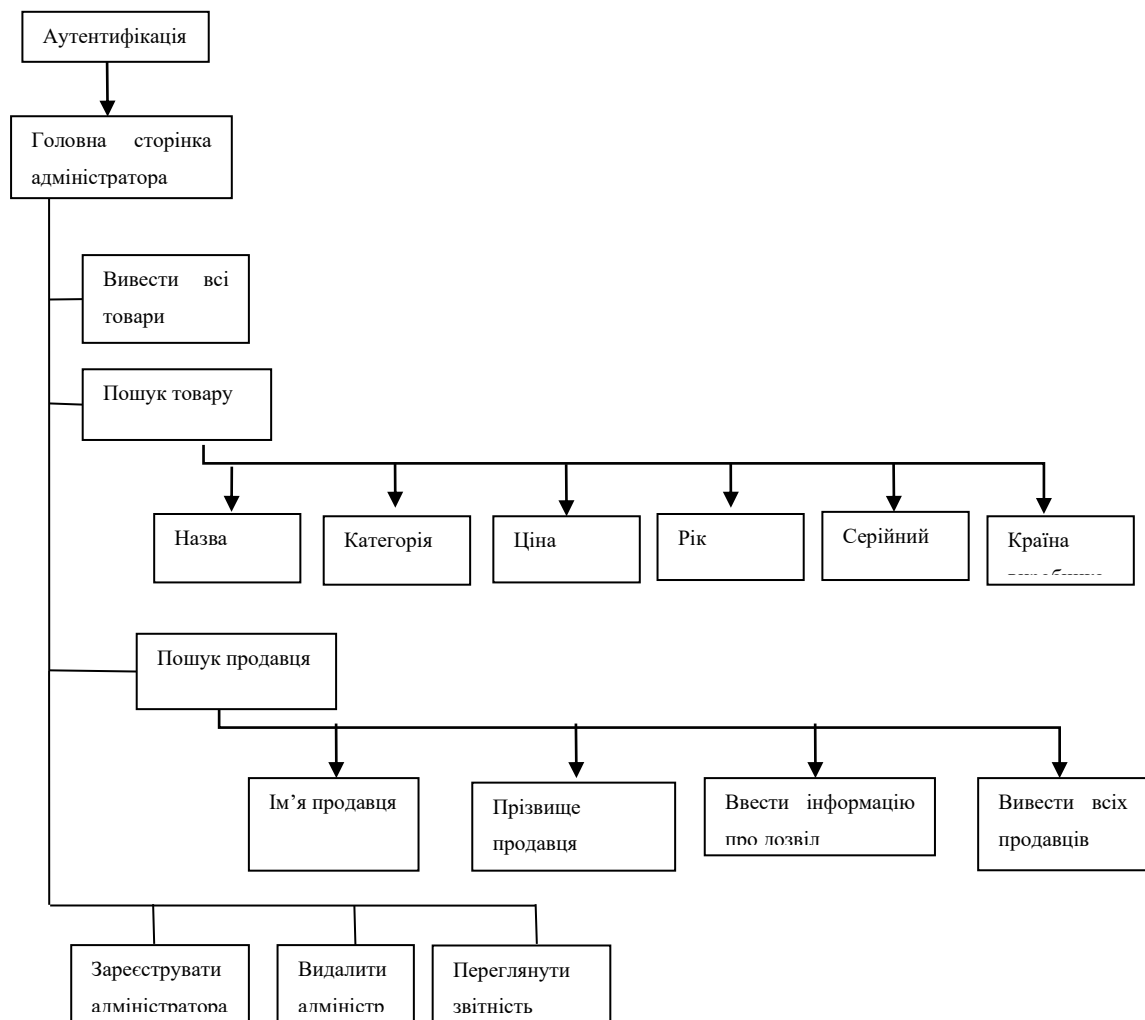
Представлення UML діаграми варіантів використання
(для старшого адміністратора)



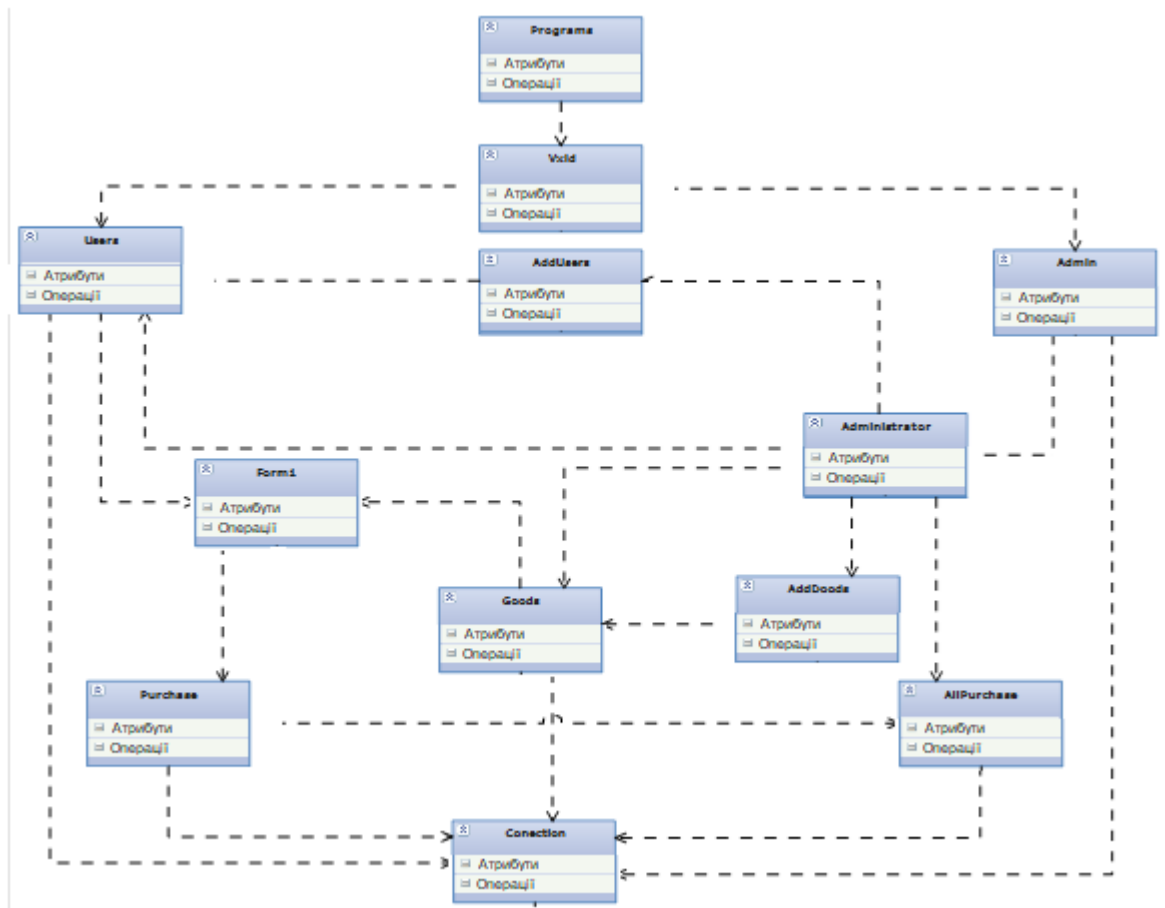
Структура користувацької частини



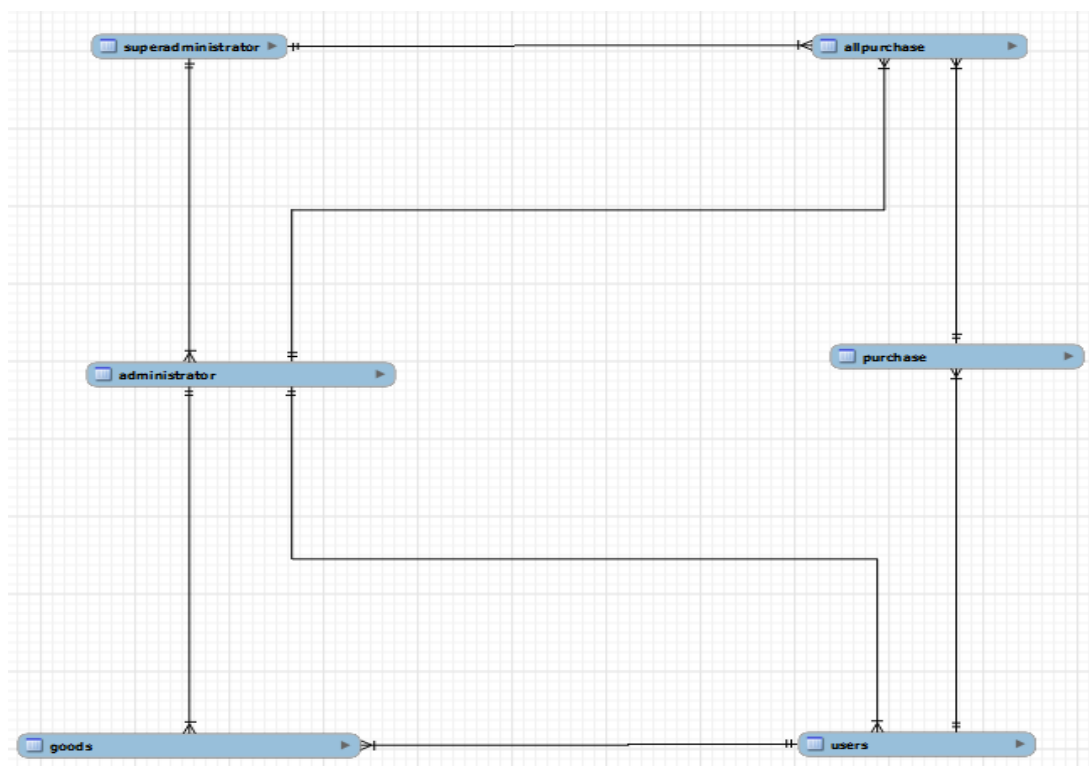
Структура першої адміністративної частини



Структура другої адміністративної частини



UML діаграма залежності класів



ER – діаграма сутностей

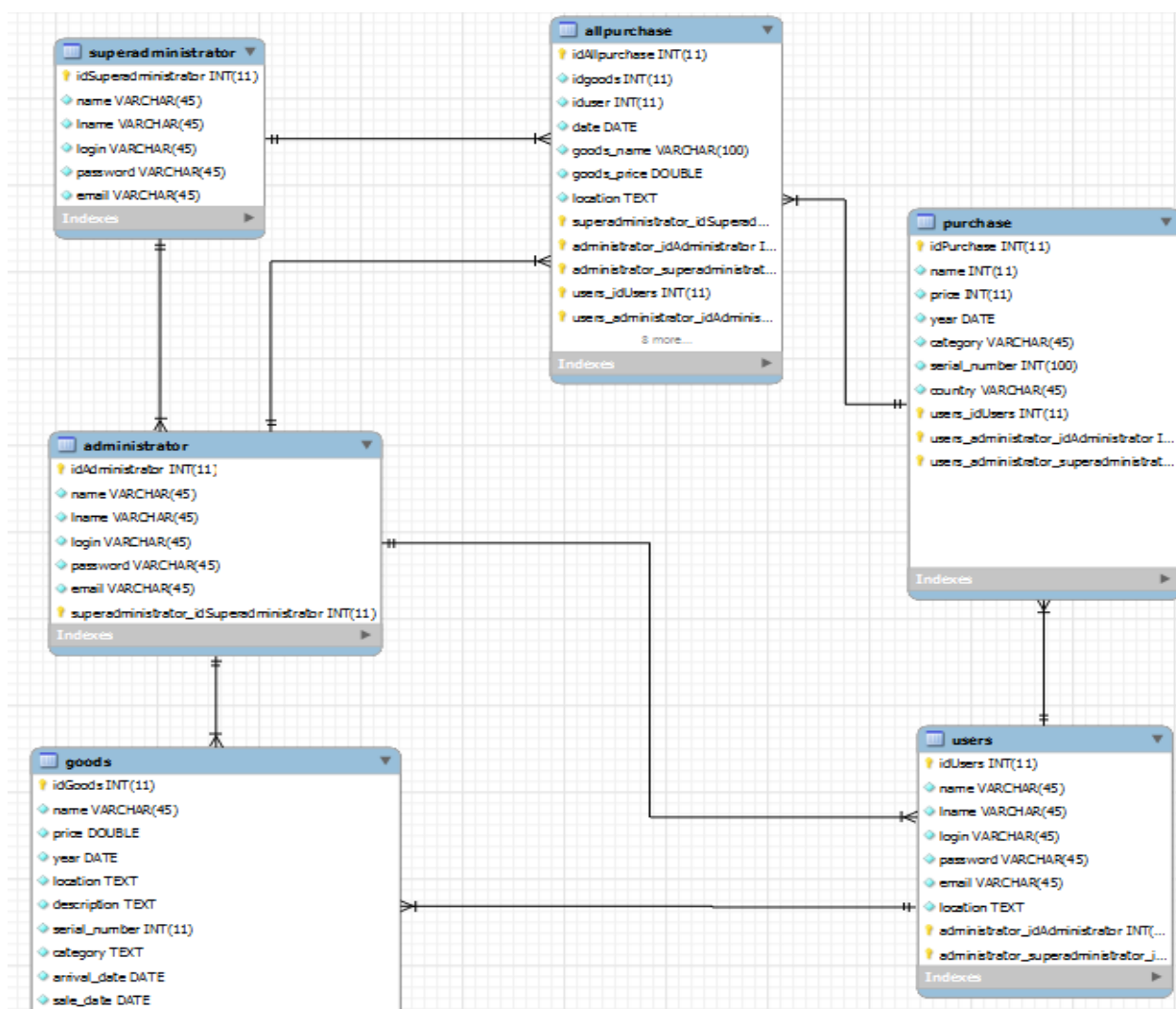
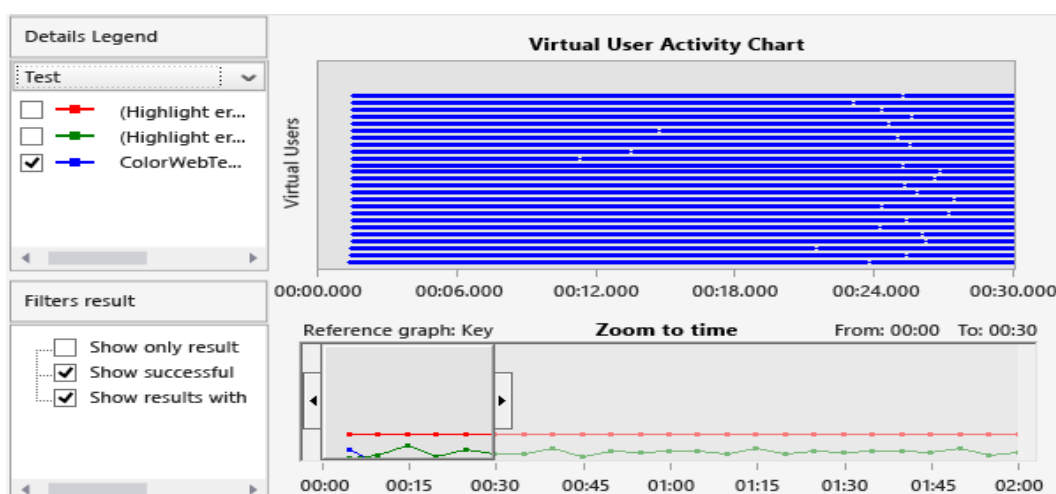


Схема реляційної бази даних



Вигляд результату тестування 1-го користувача

Додаток Г Диск з роботою