

Міністерство освіти і науки України
Луцький національний технічний університет
Факультет комп'ютерних та інформаційних технологій
Кафедра інженерії програмного забезпечення

КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»

РОЗРОБКА АВТОМАТИЗОВАНОЇ СИСТЕМИ ОБЛІКУ МАТЕРІАЛІВ
НА СКЛАДІ НА БАЗІ
МОВИ ПРОГРАМУВАННЯ C#

DEVELOPMENT OF AN AUTOMATED MATERIAL ACCOUNTING
SYSTEM IN THE WAREHOUSE BASED ON THE C# PROGRAMMING
LANGUAGE

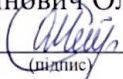
спеціальність 121 Інженерія програмного забезпечення
(шифр і назва спеціальності)

освітня програма «Інженерія програмного забезпечення»
(назва освітньої програми)

Виконав: здобувач вищої освіти
Групи ІПЗс-21
Веремейчик Євгеній Петрович


(підпис)

Керівник:
к.т.н., доцент
Суринович Олена Миколаївна


(підпис)

Кваліфікаційну роботу
допущено до захисту
« 8 » 06 2024 р.
к.т.н., доцент
Гарант освітньої програми:
Ліщина Наталія Миколаївна

(підпис)

Луцьк – 2024 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра інженерії програмного забезпечення

Ступінь вищої освіти: бакалавр

Галузь знань: 12 Інформаційні технології

Спеціальність: 121 Інженерія програмного забезпечення

Освітня програма: «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

М. М. Мичине
« 12 » 02 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Веремейчику Євгенію Петровичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: «Розробка автоматизованої системи обліку матеріалів на складі на базі мови програмування C#»

Керівник роботи: к.т.н., доцент, Суринович Олена Миколаївна

затверджені наказом закладу вищої освіти від «30» грудня 2023 р. № 477/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи «5» червня 2024 р.

3. Вихідні дані до роботи: технічне та програмне забезпечення ЕОМ, вимоги до розробки програмного забезпечення, ергономічні вимоги до функціонування програмного засобу.

4. Зміст розрахунково-пояснювальної записки (перелік питань, що потрібно розробити):

Аналіз сучасного стану проблеми, існуючих методів і засобів її розв'язання, аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення, опис функціонального наповнення об'єкта проектування, практична реалізація об'єкта проектування.

5. Перелік графічного матеріалу: 11 рисунків, з яких 2 діаграми та 2 схеми

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		Завдання видав	Завдання прийняв
Аналіз предметної області	Суринович О. М.		
Специфікації вимог до розробленої системи	Суринович О. М.		
Розробка об'єкта проектування	Суринович О. М.		
Нормоконтроль	Повстяна Ю. С.		
Гарант ОП	Ліщина Н. М.		
Показник запозичень тексту		10,2%	
Академічна доброчесність	Яцук А. А.		

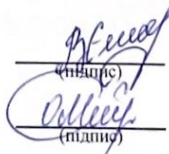
7. Дата видачі завдання «2» лютого 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ 3/п	Назва етапів кваліфікаційної роботи бакалавра	Строк виконання етапів роботи	Примітка
1	Огляд літературних джерел по темі кваліфікаційної роботи бакалавра	07.02.2024 р.	вик.
2	Аналіз проблеми розробки та впровадження об'єкту проектування	27.02.2024 р.	вик.
3	Обґрунтування вибору шляхів, технологій (алгоритмів) і засобів вирішення поставленого завдання	12.03.2024 р.	вик.
4	Розробка функціонально-структурної схеми роботи об'єкта проектування та проектування бази даних	17.04.2024 р.	вик.
5	Практична реалізація об'єкта проектування та розробка бази даних	18.05.2024 р.	вик.
6	Тестування та налагодження об'єкта проектування	01.06.2024 р.	вик.
7	Здача чистового варіанту кваліфікаційної роботи бакалавра на кафедрі	05.06.2024 р.	вик.

Здобувач вищої освіти

Керівник кваліфікаційної роботи


 (підпис)

 Веремейчик Є. П.
 (прізвище, ініціали)

 Суринович О. М.
 (прізвище, ініціали)

**Рецензія
на кваліфікаційну роботу**

Здобувач вищої освіти: Веремейчик Євгеній Петрович

Тема: «Розробка автоматизованої системи обліку матеріалів на складі на базі мови програмування C#».

Коротка характеристика кваліфікаційної роботи:

Кваліфікаційна робота бакалавра складається з трьох розділів, в яких проведено аналіз предметної області, здійснена чітка постановка завдань за темою роботи, проведено дослідження методів та засобів для розробки автоматизованої системи обліку матеріалів на складі, здійснено практичну реалізацію і проведено дослідження результату ефективності додатку, а також здійснено аналіз отриманих результатів.

Самостійні розробки і пропозиції автора:

Автором було самостійно розроблено автоматизовану систему обліку матеріалів на складі на базі мови програмування C#

Практичне значення роботи

полягає в ефективному використанні розробленої системи обліку матеріалів на складі для підприємств з продажем товарів.

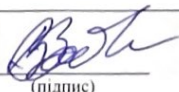
Недоліки: Суттєвих недоліків в роботі не виявлено.

Загальний висновок

Кваліфікаційна робота бакалавра виконана на достатньому рівні. Робота виконана з розумінням предмету. Матеріал викладено в чіткій та зручній формі. Вважаю, що кваліфікаційна робота відповідає вимогам до випускних робіт і заслуговує позитивної оцінки, а її авторів, студенту Веремейчику Євгенію Петровичу може бути присвоєний кваліфікаційний рівень "бакалавр" зі спеціальності «Інженерія програмного забезпечення»

Рецензент: Заблоцький В. Ю., к. т. н., доц., завідувач кафедри електроніки та телекомунікацій

Рецензент кваліфікаційної роботи


(підпис)

« 7 » серпня 202 4 р.

Міністерство освіти і науки України
Луцький національний технічний університет
Факультет комп'ютерних та інформаційних технологій
Кафедра інженерії програмного забезпечення

ВІДГУК
КЕРІВНИКА НА КВАЛІФІКАЦІЙНУ РОБОТУ

Здобувач вищої освіти Веремейчик Євгеній Петрович
(прізвище, ім'я, по батькові)
група ППЗс-21

Тема кваліфікаційної роботи: «Розробка автоматизованої системи обліку матеріалів на складі на базі мови програмування C#».

Керівник Суринович Олена Миколаївна

Актуальність теми Розробка автоматизованої системи обліку матеріалів на складі на базі мови програмування C# є важливим кроком для підвищення ефективності та точності складських операцій. Така система спрямована на зменшення людських помилок, оптимізацію ресурсів та підвищення швидкості обробки даних. Використання сучасних технологій та інструментів дозволяє створити надійне та масштабоване рішення, яке відповідає потребам сучасного бізнесу.

Об'єкт дослідження Автоматизована система обліку матеріалів на складі на базі мови програмування C#.

Характеристика теоретичного рівня, наявності самостійних розробок і практичної значимості роботи: Автор аналізує різні методи та засоби для розробки автоматизованої системи обліку матеріалів на складі на базі мови програмування C#. Визначає найкращі з них і обґрунтовує своє рішення. Зміст роботи відповідає обраній темі. Позитивними рисами кваліфікаційної роботи бакалавра є системність та послідовність викладення матеріалу, а також якісна його практична реалізація.

Зауваження та недоліки: Суттєвих недоліків в роботі не виявлено.

Загальний висновок: Кваліфікаційна робота бакалавра виконана на високому рівні. Робота виконана відповідно до вимог, логічно й послідовно описує хід виконання поставленого завдання. Пояснювальна записка відповідає стандартам до її оформлення. Дана робота заслуговує позитивної оцінки.

Керівник _____


(підпис)

(Суринович О. М.)
(прізвище, ім'я, по батькові)

« 2 » 06 2024 р.

АНОТАЦІЯ

Веремейчик Є. П. Розробка автоматизованої системи обліку матеріалів на складі на базі мови програмування C#. Рукопис.

Кваліфікаційна робота бакалавра ОП «Інженерія програмного забезпечення». Луцький національний технічний університет. Луцьк, 2024.

Кваліфікаційна робота бакалавра складається зі вступу, трьох розділів, висновків та списку використаних джерел.

У першому розділі здійснено аналіз предметної області і постановка завдань на кваліфікаційну роботу. В другому розділі визначено специфікацію вимог до розроблюваної системи. У третьому розділі описано етапи розробки автоматизованої системи обліку матеріалів на складі на базі мови програмування C#. У висновках узагальнено інформацію, відображену у попередніх частинах. Результати розробки демонструють можливості розробленої системи.

Ключові слова: C#, .NET, Windows API, автоматизована система, склад, облік, патерн, MVP.

ABSTRACT

Veremeichyk Ye. Development of an Automated Material Accounting System in the Warehouse Based on the C# Programming Language. Manuscript.

Bachelor's qualifying thesis of the educational program "Software Engineering". Lutsk National Technical University. Lutsk, 2024.

The bachelor's qualifying thesis consists of an introduction, three sections, conclusions and references.

In the first section, the analysis of the subject area and the setting of tasks for the qualification work were carried out. The second section defines the specification of requirements for the developed system. The third section describes the stages of development of an automated material accounting system in the warehouse based on the C# programming language. The conclusions summarize the information presented in the previous parts. The development results demonstrate the capabilities of the automated material accounting system.

Keywords: C#, .NET, Windows API, automated system, composition, accounting, pattern, MVP.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1 АНАЛІЗ АВТОМАТИЗОВАНОЇ СИСТЕМИ ОБЛІКУ МАТЕРІАЛІВ І ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ	9
1.1 Аналіз сучасного стану проблеми.....	9
1.2 Постановка завдання на кваліфікаційну роботу бакалавра.....	12
Висновки до розділу 1	12
РОЗДІЛ 2 СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ	13
2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення.....	13
2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання	15
Висновки до розділу 2	20
РОЗДІЛ 3 РОЗРОБКА АВТОМАТИЗОВАНОЇ СИСТЕМИ ОБЛІКУ МАТЕРІАЛІВ НА СКЛАДІ НА БАЗІ МОВИ ПРОГРАМУВАННЯ C#.....	21
3.1 Практична реалізація об'єкта проектування	21
3.2 Тестування та налагодження інформаційно-комп'ютерної системи ..	24
3.3 Розробка бази даних	27
Висновки до розділу 3	29
ВИСНОВКИ.....	31
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	34

ВСТУП

Актуальність роботи. Використання складів бере свій початок з часів розвитку землеробства та продажу різних благ. Виникла необхідність десь зберігати запаси, при цьому дотримуватись умов, за яких ці запаси зможуть зберегти свій початковий стан. У Стародавньому Єгипті сільськогосподарські товари зберігалися в зерносховищах і були єдиним різновидом складів на той час. Згодом було запроваджено складські операції, які вимагали нових виконавців.

Сам принцип зберігання полягає в правильному розташуванні та укладанні матеріалів на складі в окремих зонах складу. Раціональним розміщенням матеріалу вважається правильне використання простору в секторах зберігання, але при цьому варто врахувати нормальні умови роботи спеціальної техніки та механізмів. Раціональне використання складського простору дозволяє максимізувати прибуток та зменшити витрати на зберігання.

Ефективність підприємства нині безпосередньо залежить від правильної організації зберігання матеріальних цінностей на складі. Для обліку матеріалів на складі можливе використання паперової звітності. Даний варіант дозволяє заощадити час та гроші на розробку автоматизованої системи. У свою чергу, автоматизація дозволяє підвищити ефективність роботи, знизити втрати матеріалів та трудовитрати виконання операцій на складі.

Процес зберігання матеріалів на складі починається після приймання і переміщення товару на складі. Основною вимогою зберігання є забезпечення умов збереження цілісності матеріалу. Матеріали, розміщені на складі, сортуються за розміром, вагою, часом та умовою зберігання, а також особливостями, які не дозволяють їх утримувати з іншими об'єктами.

Головною метою кваліфікаційної роботи бакалавра є розробка автоматизованої системи обліку матеріалів на складі на базі мови програмування C#.

Для реалізації мети потрібно виконати наступні завдання:

- проаналізувати предметну область;
- визначити інструментальні засоби для розробки;
- спроектувати автоматизовану систему;
- реалізувати автоматизовану систему обліку матеріалів на складі на базі мови програмування C#;
- описати роботу розробленої системи.

Об'єкт кваліфікаційної роботи бакалавра – автоматизована система обліку матеріалів на складі на базі мови програмування C#.

Предмет кваліфікаційної роботи бакалавра – засоби розробки автоматизованої системи обліку матеріалів на складі.

Практична цінність кваліфікаційної роботи бакалавра полягає у використанні автоматизованої системи обліку матеріалів на складі в різноманітних приватних та державних організаціях.

РОЗДІЛ 1

АНАЛІЗ АВТОМАТИЗОВАНОЇ СИСТЕМИ ОБЛІКУ МАТЕРІАЛІВ І ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ

1.1 Аналіз сучасного стану проблеми

Склад – це спеціальне приміщення для зберігання матеріальних цінностей та організації складських послуг. На сьогоднішній день функціонування будь-якого складу можна розділити на три великі процеси, такі як:

- 1) надходження товару на склад;
- 2) зберігання та облік;
- 3) видача товару.

Перед надходженням товару на склад, постачальнику створюється замовлення з необхідною кількістю товару. Після угоди товар потрапляє на склад, де його потрібно прийняти, занести інформацію про товар у базу даних і розмістити товар на складі. Перед тим, як розмістити товар на складі, на нього наклеюється етикетка з характеристиками товару, а також місце зберігання на складі.

Зберігання та облік матеріалів на складі включає транспортування та інвентаризацію. У процесі інвентаризації відбувається перевірка відповідності між фактичними характеристиками товару з характеристиками, заявленими в документації. Товар може бути переміщений як усередині приміщення, так і на інший склад [1].

Видача товару включає різні операції відвантаження товарів, наприклад, відвантаження клієнту, списання, списання на внутрішні потреби. Видача товару відбувається лише після формування документа про видачу. У ньому зберігаються такі відомості про товар, як:

- найменування товару;
- номер товару у системі;
- кількість;

- ціна продажу;
- дата продажу;
- назва отримувача.

Сервіси автоматизованого обліку матеріалів на складі можна розділити на дві великі групи:

- 1) програми для купівлі та завантаження;
- 2) хмарні послуги [2].

Коротко опишемо аналоги, що належать до першої групи.

До прикладу, – «УкрСклад» – функціональна програма, призначена для ведення обліку товарів на складах або фірмах (рис. 1.1).

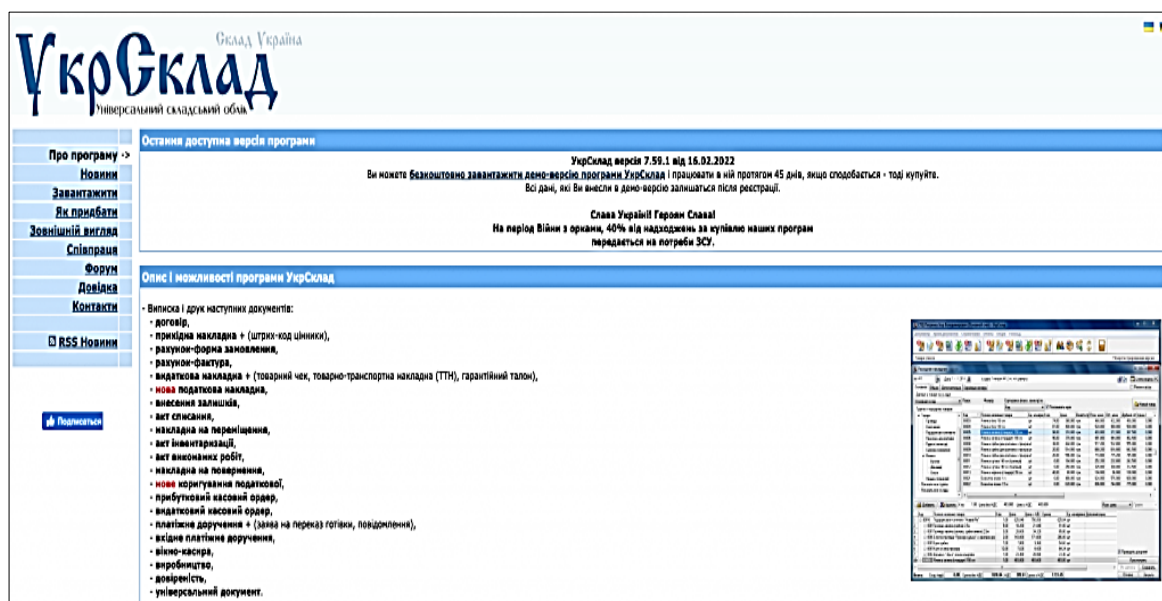


Рисунок 1.1 – Головна сторінка програми «УкрСклад» [3]

До основних функцій системи належать:

- ведення архіву різноманітної документації, створення нових документів з можливістю використовувати попередніх шаблонів, а також швидкий пошук, фільтрація та сортування у всіх документах і довідниках;
- можливість створення довідників для товарів, послуг, валют, клієнтів і активних компаній, а також наявність довідника банківських установ України.

Під час створення документів автоматично застосовується шаблонна нумерація для кожної компанії окремо;

– можливість багатокористувацької роботи в локальній мережі з функцією спеціального збереження, а також подальшого перенесення бланків на інші комп'ютери, де програма не встановлена.

Ще однією системою, що належить до першої групи є «IBS» (рис. 1.2).

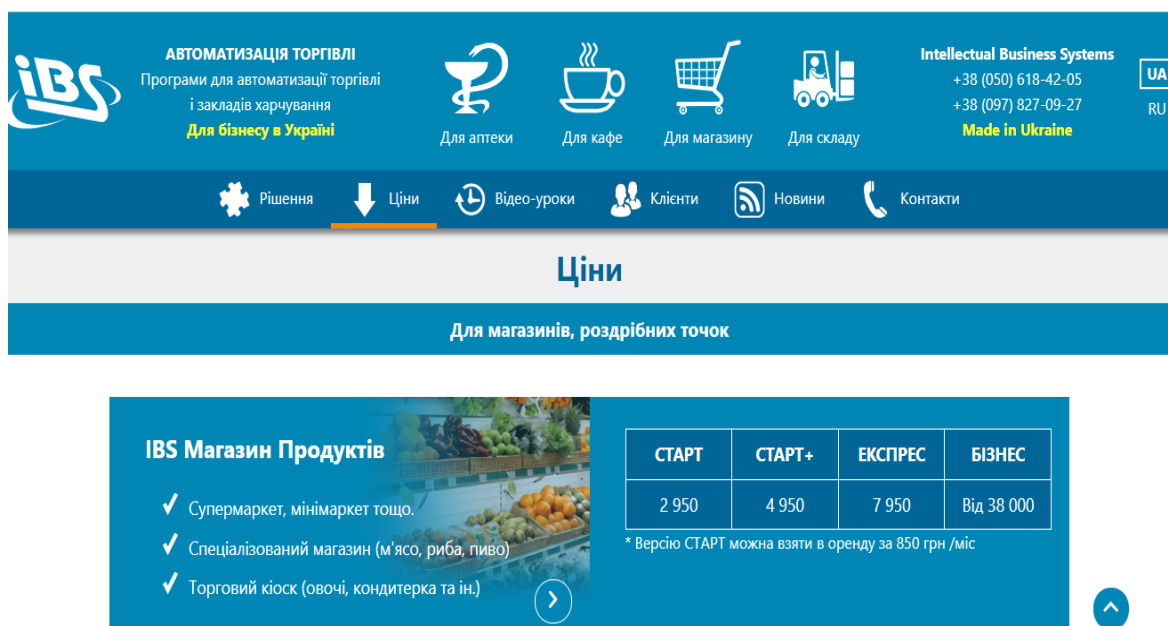


Рисунок 1.2 – Головна сторінка програми «IBS» [4]

До основних функцій програми належать:

– ведення єдиного довідника товарів з синхронізацією по всіх магазинах мережі;

– управління продажними цінами, акціями, знижками (ведення прайс-листів) з синхронізацією цін по мережі;

– управління товарообігом: закупівлі, переміщення між магазинами, повернення постачальникам.

Однак, постає завдання розробити автоматизовану систему обліку товарів на складі, що матиме ряд переваг у відповідності до аналогів [5].

1.2 Постановка завдання на кваліфікаційну роботу бакалавра

Використання автоматизованої системи обліку матеріалів має підвищити ефективність роботи, знизити втрати матеріалів та трудовитрати виконання операцій на складі.

Для цього ставимо перед собою ряд завдань та вимог:

- проаналізувати предметну область;
- визначити інструментальні засоби для розробки;
- спроектувати автоматизовану систему, що включатиме:

- 1) авторизацію у додатку;
- 2) швидкий та зручний інтерфейс для роботи користувача;
- 3) організоване зберігання матеріалів;
- 4) різні операції з матеріалами, що знаходяться на складі;
- 5) можливість автоматичного складання звіту;

– реалізувати автоматизовану систему обліку матеріалів на складі на базі мови програмування C#, де головними користувачами мають бути адміністратор та комірник;

- описати роботу розробленої системи.

Цілі, які необхідно досягти під час автоматизації, спрямовані на зменшення часу, що витрачається персоналом на складання різних документів, а також зменшення помилок через людський фактор.

Висновки до розділу 1

У першому розділі роботи було означено проблематику та описано актуальність розроблюваного продукту. Також було проаналізовано існуючі аналоги з означенням їх функцій. Окрім цього, було виокремлено завдання та вимоги до майбутньої автоматизованої системи обліку товарів на складі.

РОЗДІЛ 2

СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ

2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення

Для реалізації майбутньої системи опишемо ряд вимог, включаючи діаграму варіантів використання.

Діаграма варіантів використання – це форма подання концептуальної моделі проектованої системи, яка описує її функціональне призначення і надалі буде деталізована, і перетворена на моделі логічного та фізичного рівнів.

Розробка діаграми варіантів використання призначена для вирішення наступних завдань:

- на початкових етапах проектування системи слід визначити загальні межі та контекст предметної області;
- виявити основні вимоги до функціональної поведінки розроблюваної системи;
- розробити початкову концептуальну модель системи, яка згодом буде деталізована до логічного та фізичного рівня;
- підготувати початкову документацію для координації взаємодії між розробниками, замовниками та користувачами.

Діаграма будується на базі трьох основних компонентів: варіант використання (use case), дійова особа або актор (actor) і зв'язок або відношення (relationship).

«Актори» – це тип користувачів, які взаємодіють із системою. Варіант використання (прецеденти) – набір деяких дій, які система надає акторам, які призводять до певного результату.

Відносини – це семантичний зв'язок між окремими елементами моделі. Існують різні відносини між елементами діаграми, які показують взаємозв'язок між акторами та варіантами використання.

У мові UML є кілька стандартних видів відносин між акторами та варіантами використання:

- асоціація (Association) – позначає специфічну роль актора у його взаємодії з окремим варіантом використання;
- включення (Include) – відношення залежності між двома елементами моделі, де зміна одного елемента (незалежного) призводить до зміни іншого елемента (залежного);
- розширення (Extend) – визначає зв'язок базового варіанта використання з іншим варіантом, функціональна поведінка якого активується базовим лише при виконанні додаткових умов;
- узагальнення (Generalization) – два або більше акторів мають спільні властивості, тобто взаємодіють з одним і тим самим набором варіантів використання однаковим чином.

Опишемо UML-діаграму «UseCase Stock», що використовуватиметься у розробленій системі (рис. 2.1) [6].

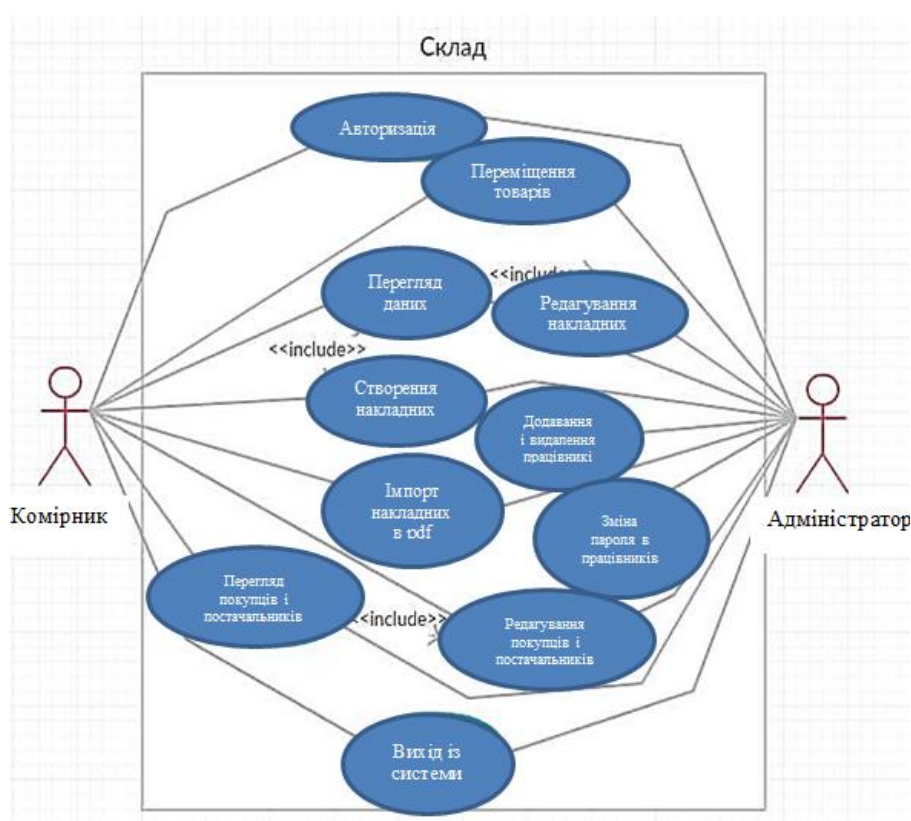


Рисунок 2.1 – UML-діаграма «UseCase Stock»

Основні функції, що будуть доступні «Адміністратору» та «Комірнику»:

- авторизація;
- переміщення матеріалу;
- перегляд накладних;
- створення накладних;
- імпорт накладних у PDF;
- редагування покупців та постачальників;
- перегляд покупців та постачальників;
- вихід із системи.

Крім основних функцій, «Адміністратор» має можливість:

- редагування накладних;
- додавання, видалення працівників;
- зміна пароля у працівників.

2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання

У 2000 році, завдяки зусиллям Андерса Хейлсберга, з'явилася мова програмування C#. Вона була створена на основі мови Delphi та середовища розробки Turbo Pascal. Наразі це потужна мова програмування, яка дозволяє створювати практично будь-які програми і підходить навіть для новачків у програмуванні.

C# є об'єктно-орієнтованою мовою. Об'єктно-орієнтований підхід допомагає вирішувати завдання з розробки великих, гнучких, масштабованих і розширюваних додатків. C# підтримує успадкування, поліморфізм, перевантаження операторів та статичну типізацію [7].

До основних можливостей даної мови належать:

- повна підтримка класів та об'єктно-орієнтованого програмування, включаючи успадкування інтерфейсів і реалізацій, віртуальні функції та перевантаження операторів;

- комплексний і чітко визначений набір основних типів;
- вбудована підтримка автоматичної генерації XML-документації;
- автоматичне звільнення динамічно-розподіленої пам'яті;
- можливість позначати класи та методи користувацькими атрибутами;
- повний доступ до бібліотеки базових класів .NET, а також легкий доступ до Windows API;
- вказівники та прямий доступ до пам'яті за потреби, хоча мова спроектована так, щоб зазвичай можна було обійтися без них;
- просте змінення ключів компіляції, що дозволяє отримувати виконувані файли або бібліотеки компонентів .NET, які можуть бути викликані іншим кодом.

Для розробки інформаційної системи автоматизації процесу обліку матеріалів на складі використовуватиметься об'єктно-орієнтована мова програмування C# [8].

Як інтегроване середовище розробки використовуватиметься Microsoft Visual Studio.

Microsoft Visual Studio – це лінійка продуктів від компанії Майкрософт, яка включає інтегроване середовище розробки програмного забезпечення та низку інших інструментальних засобів. Перша версія була випущена у 1997 році [9].

Основні компоненти Visual Studio включають редактор вихідного коду, вбудований відладчик, дизайнери класів, редактор форм для створення графічного інтерфейсу та інше.

Microsoft Visual Studio дозволяє розробляти як консольні програми, так і програми з графічним інтерфейсом, включаючи підтримку технології Windows Forms. Можливості Visual Studio можуть бути розширені за допомогою сторонніх плагінів.

Visual Studio містить редактор вихідного коду з підтримкою IntelliSense та можливістю легкого рефакторингу коду. Вбудований налагоджувач може працювати як з вихідним кодом, так і на рівні машинного коду. Серед інших

інструментів є редактор форм для полегшення створення графічних інтерфейсів, дизайнер класів, дизайнер схем баз даних і веб-редактор.

Для створення БД, і навіть різних операцій із нею було обрано СУБД MS SQL Server.

Вибір СУБД MS SQL Server обґрунтований зручністю роботи з ним в Microsoft Visual Studio (засоби для підключення, генерації та виконання запитів тощо).

MS SQL Server – це система управління базами даних (СУБД), яка використовує мову Transact-SQL, розроблену компанією Microsoft. Перша версія була випущена в 1987 році. Основним компонентом MS SQL Server, як і будь-якої СУБД, є база даних – сховище даних, організованих певним чином. Фізично база даних часто представлена у вигляді файлу на жорсткому диску, хоча це не завжди так [10].

MS SQL Server використовує реляційну модель організації баз даних, розроблену в 1970 році Едгаром Коддом і Крістофером Дейтом. У цій моделі дані зберігаються у вигляді таблиць, що складаються з рядків і стовпців. Кожен рядок представляє окремий об'єкт, наприклад, олівець, а стовпці містять атрибути цього об'єкта, наприклад, колір і розмір. Сьогодні MS SQL Server фактично є стандартом для організації баз даних. Однак творці реляційної моделі даних, Едгар Кодд і Крістофер Дейт, разом зі своїми прихильниками, стверджують, що SQL не відповідає реляційній моделі з низки причин.

Для ідентифікації кожного рядка в таблиці використовується первинний ключ (primary key), який може складатися з одного або кількох стовпців. Первинний ключ забезпечує унікальність кожного рядка в таблиці, тому два рядки не можуть мати однаковий первинний ключ. Використовуючи первинний ключ, можна однозначно посилатися на певний рядок у таблиці.

Ключі дозволяють встановлювати зв'язки між таблицями, тобто організовувати зв'язки між двома таблицями. Кожна таблиця може розглядатися як відношення (relation). Для взаємодії з базою даних використовується мова SQL (Structured Query Language). Клієнт (наприклад,

зовнішня програма) надсилає SQL-запит через спеціальний API. СУБД інтерпретує та виконує цей запит, а потім надсилає клієнту результат виконання.

MS SQL Server використовує версію SQL, що називається Transact-SQL (T-SQL). T-SQL включає додатковий синтаксис для збережених процедур і підтримує транзакції, що забезпечує взаємодію бази даних з керуючим додатком.

Тепер коротко опишемо архітектуру розроблюваної системи (рис. 2.2).

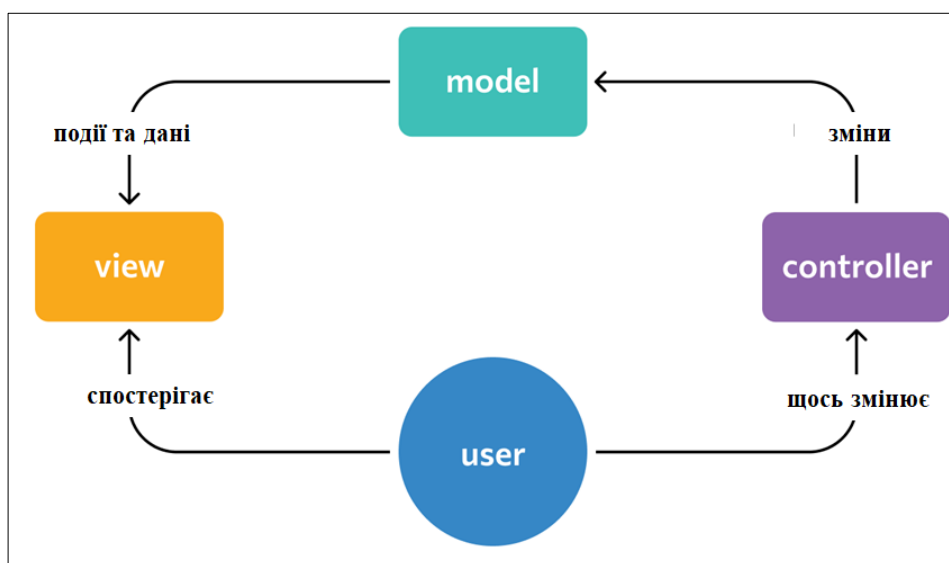


Рисунок 2.2 – Схема патерна MVC

Архітектура інформаційної системи – концепція, за якою взаємодіють компоненти інформаційної системи [11].

Model-View-Controller – патерн, у якому відбувається поділ даних програми, інтерфейсу та логіки на три окремих компоненти таким чином, що кожен компонент може бути реалізований незалежно від інших. Патерн був описаний Трюгве Рєєнскауг в 1978 році. Надалі, на основі MVC були розроблені й інші патерни, наприклад, MVVM, MVP та інше.

Model-View-Presenter – патерн, похідний від Model-View-Controller, який використовується в основному для побудови користувача інтерфейсу. Основна

ідея цього патерну в тому, що і контролер, і уявлення залежать від моделі, але модель не залежить від цих двох компонентів.

Елемент Model – це бізнес-логіка програми, тобто це набір даних та методів їх обробки. Модель нічого не повинна знати про існування Presenter та View. Model є повністю незалежною.

Елемент View забезпечує різні способи представлення даних. Він може бути шаблоном, який заповнюється даними. View може бути кілька різних видів, і він звертається до Presenter для оновлення даних. Елемент Presenter в даному шаблоні бере на себе функціональність посередника (аналогічно контролеру в MVC) і відповідає за керування подіями інтерфейсу користувача.

Цей підхід дозволяє створювати абстракцію уявлення. Реалізовано цей патерн зазвичай за допомогою створення інтерфейсів подання. У кожного представлення будуть інтерфейси з певними наборами методів та властивостей, необхідних Presenter. Presenter ж, у свою чергу, ініціалізується з даним інтерфейсом, підписується на події уявлення та по спрацюванні подій оновлює дані (рис. 2.3).

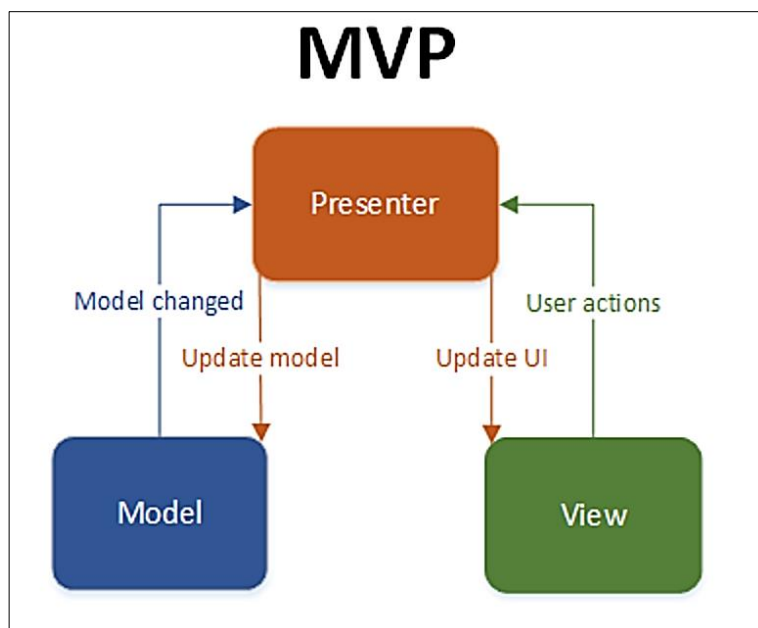


Рисунок 2.3 – Схема паттерна MVP [12]

Висновки до розділу 2

У другому розділі для реалізації майбутньої системи було описано ряд вимог, включаючи діаграму варіантів використання. Також розроблено та обґрунтовано UML-діаграму «UseCase Stock». Окрім цього, детально описано об'єктно-орієнтовану мову програмування С#, що використовувалася для розробки інформаційної системи автоматизації процесу обліку матеріалів на складі.

Додатково був також здійснений опис інтегрованого середовища розробки Microsoft Visual Studio.

Також коротко було описано архітектуру розроблюваної системи.

РОЗДІЛ 3

РОЗРОБКА АВТОМАТИЗОВАНОЇ СИСТЕМИ ОБЛІКУ МАТЕРІАЛІВ НА СКЛАДІ НА БАЗІ МОВИ ПРОГРАМУВАННЯ C#

3.1 Практична реалізація об'єкта проектування

Покроково опишемо етапи розробки продукту.

Model містить бізнес-логіку програми через Presenter отримує або передає дані на View. Для дотримання архітектурного патерну MVP необхідно створити інтерфейси для роботи з репозиторієм, а також їх реалізувати.

Інтерфейс «IuserRepository» визначає основні операції для роботи з користувачами, такі як отримання користувача за ідентифікатором, отримання всіх користувачів, додавання, оновлення та видалення користувачів. Клас «User» представлений частково і може бути розширений відповідно до потреб програми. Інтерфейс «IuserRepository» зображено у лістингу 3.1.

Лістинг 3.1 – Функція «IuserRepository»

```
using System;
using System.Collections.Generic;
using System.Threading.Tasks;
namespace YourNamespace
{
    public interface IUserRepository
    {
        // Отримати користувача за ідентифікатором
        Task<User> GetUserByIdAsync(Guid userId);
        // Отримати всіх користувачів
        Task<IEnumerable<User>> GetAllUsersAsync();
        // Додати нового користувача
        Task AddUserAsync(User user);
        // Оновити існуючого користувача
        Task UpdateUserAsync(User user);
        // Видалити користувача за ідентифікатором
        Task DeleteUserAsync(Guid userId);
    }
    // Приклад класу User
    public class User
    {
        public Guid Id { get; set; }
        public string Name { get; set; }
        public string Email { get; set; }
    }
}
```

```

        // Інші властивості користувача
    }
}

```

Кінець лістингу 3.1

Model є своєрідним фасадом до деякого джерела даних або шару доступу до даних. Її завдання – завантажувати та зберігати дані відповідно до завдань конкретного екрана.

Спочатку створюється рядок підключення до БД. Першим рядком є ім'я сервера, друга назва бази даних, третій деякі параметри безпеки. Далі створюється список класу «LoginModel», у якому зберігаються лише прізвище та пароль користувачів. Далі відбувається читання з БД за допомогою класу SqlConnection. Даний клас дозволяє інкапсулювати sql-вираз, який має бути виконано. Наприкінці відбувається збереження даних до списку «name». Ця конструкція «using» закриває з'єднання з базою даних самостійно.

На рисунку 3.1 зображено метод, який повертає «true» у разі збігу даних, які ввів користувач, з даними, що зберігаються в базі даних.

```

public bool CheckLogin(string login, string pass)
{
    List<LoginModel> loginmodel = getAll();
    foreach(LoginModel m in loginmodel)
    {
        if (m.Surname == login && m.Password == pass)
            return true;
    }
    return false;
}

```

Рисунок 3.1 – Метод перевірки

View відповідає за візуалізацію даних, отриманих від Model. View створюється за допомогою інтерфейсу програмування програм Windows Forms. У View реалізовані інтерфейси, за допомогою яких Presenter спілкується з

поданням. У проекті View реалізовано 7 представлень, а відповідно 7 Presenter та Model (рис. 3.2).

```
public interface ILogin
{
    string Login { get; }
    string Password { get; }
    bool CheckLogin { get; set; }
    event EventHandler LoginButtonClick;
}
```

Рисунок 3.2 – Інтерфейс класу «ILogin»

Перший метод є прив'язкою події. Якщо користувач натиснув кнопку «Увійти», то викликається даний метод і передає інформацію про натискання Presenter. Наступний метод – конструктор класу. Перший рядок ініціалізує компоненти, а наступний прив'язує подію до натискання кнопки. Останні дві реалізації дозволяють Presenter зчитувати логін та пароль, введений користувачем.

Presenter визначає поведінку та структуру програми, не залежить від конкретної реалізації інтерфейсу. Presenter переважно забезпечує реалізацію варіантів використання програми, а також координує взаємодію користувача з бізнес-логікою.

Елементи «_view» і «_model» необхідні для спілкування з View і Model. Далі реалізується конструктор класу.

У «Presenterlogin» так само, як і в View відбувається прив'язка до методу тієї самої події, яка відбувається при натисканні кнопки «Увійти». Метод, який спрацьовує при спробі користувача увійти до системи, викликає метод LoginCheck від View, який у свою чергу приймає bool значення, одержуване з методу CheckLogin від Model. Даний метод набирає значення рядків, які ввів користувач.

3.2 Тестування та налагодження інформаційно-комп'ютерної системи

На даному етапі розробки програми існує дві посади (користувачі):

- комірник;
- адміністратор.

Комірник зобов'язаний реєструвати всілякі переміщення товару на складі, а також осіб, які належать до цього товару.

Адміністратор, крім обов'язків комірника, має можливість змінювати деякі дані в системі.

Для роботи з системою комірника необхідно спочатку авторизуватися. Під час відкриття програми з'являється вікно входу. Логіном є прізвище комірника, пароль видає адміністратор. Вікно входу до програми зображено на рисунку 3.3.

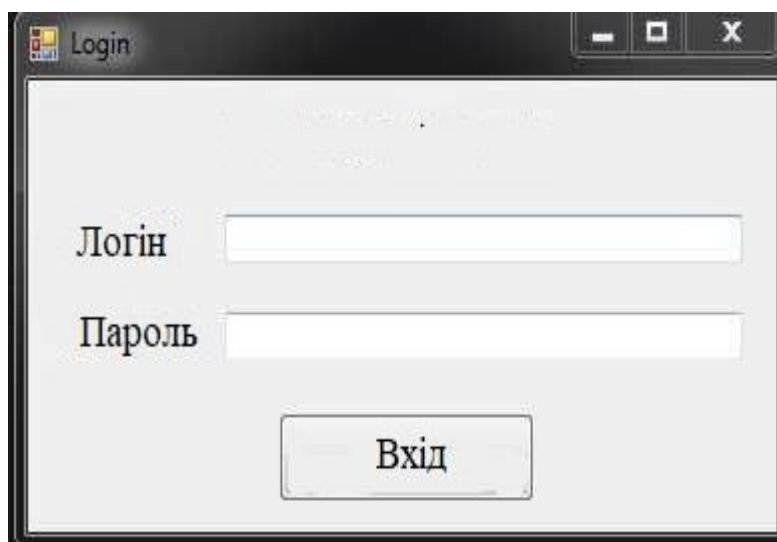


Рисунок 3.3 – Вікно входу до програми

Після правильного введення логіну та пароля відкривається головне меню програми. Головне меню представлено рисунку 3.4. У центрі вікна знаходиться список всіх товарів, розташованих на складі.

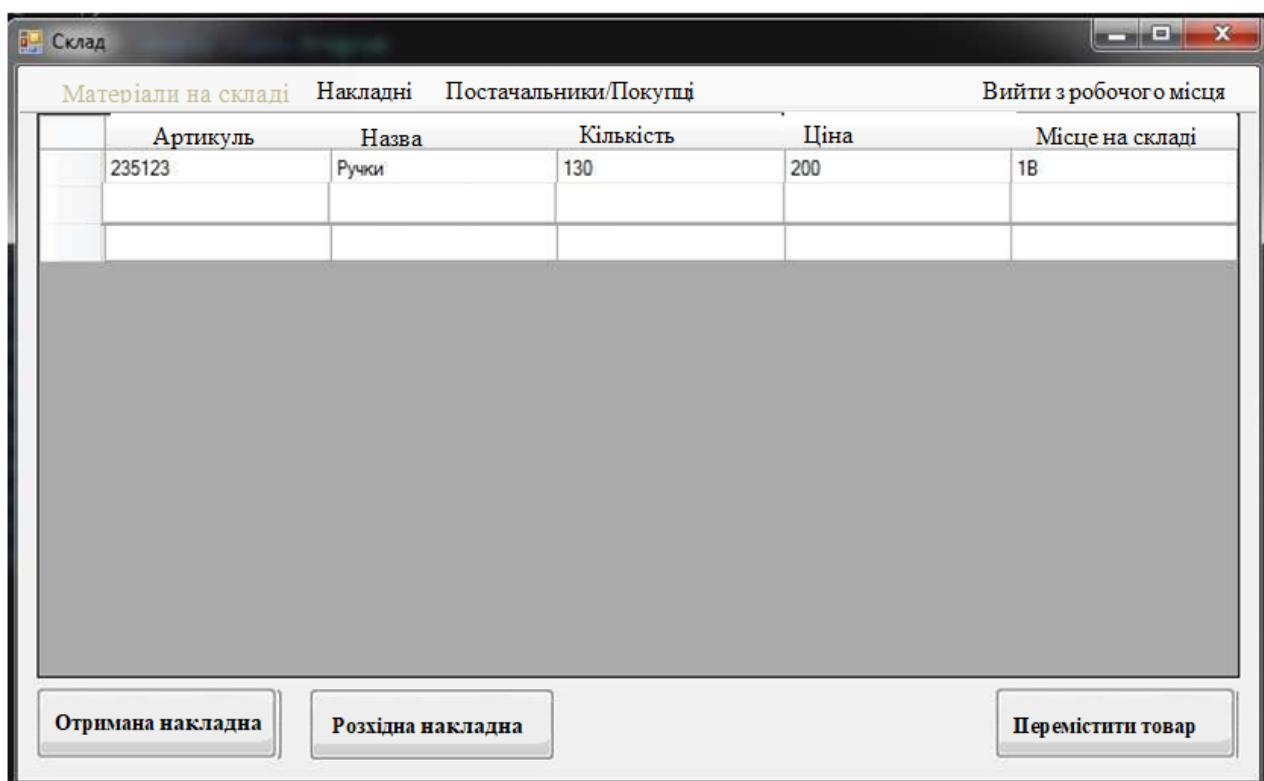


Рисунок 3.4 – Головне меню програми

Якщо потрібно перемістити товар на нове місце на складі, потрібно виділити необхідний товар у списку та натиснути кнопку «Перемістити товар». Відобразиться вікно з можливістю вибору місця зберігання. Після цієї зміни необхідно дати вказівку перемістити товар вантажникам складу із зазначенням нового місця зберігання.

При завершенні робочої зміни або необхідності вийти з робочого місця необхідно вийти із системи для запобігання доступу до даних. При натисканні кнопки «Вийти з робочого місця» головне меню програми закривається та з'являється вікно входу до системи.

При необхідності перегляду накладних необхідно натиснути кнопку «Накладні». Меню перегляду накладних зображено на рисунку 3.5. При натисканні кнопки «Імпорт у PDF» створюється PDF-файл із вибраним звітом.

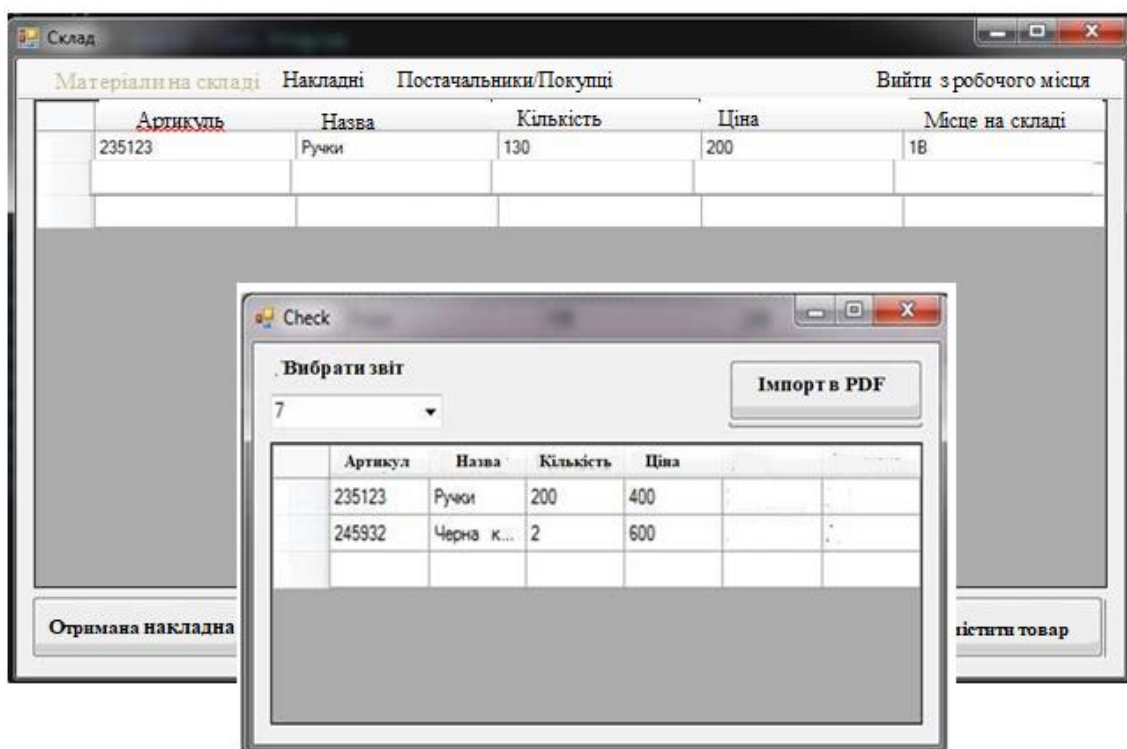


Рисунок 3.5 – Меню перегляду накладних

При натисканні кнопки «Постачальники / Покупці» відкривається вікно з можливістю перегляду та додавання постачальників та покупців (залежно від вибору у списку, що випадає. Для того, щоб додати дані, необхідно заповнити поля:

- назва компанії;
- ім'я представника;
- прізвище представника;
- телефон;
- додаткова інформація (необов'язкове поле).

Після введення інформації потрібно натиснути кнопку «Додати».

При необхідності видалення сутності необхідно виділити та натиснути кнопку «Видалити». Ця кнопка не видаляє назавжди постачальника або покупця із системи, всі дані про них зберігаються, але в програмі вони перестають відображатись.

Адміністратор, крім основних функцій, які доступні комірнику, має низку додаткових можливостей:

- змінювати інформацію або видаляти накладну у разі помилки;
- відновлювати віддалених постачальників чи покупців;
- додавати або видаляти працівників складу із системи;
- змінювати інформацію про працівників складу у системі.

При необхідності адміністратор може виконувати обов'язки комірника.

Щоб додати працівника необхідно заповнити поля даного вікна:

- ім'я;
- прізвище;
- по батькові;
- посада;
- пароль.

Після заповнення всіх полів потрібно натиснути кнопку «Додати».

Щоб видалити працівника складу, необхідно виділити необхідного працівника у списку і натиснути кнопку «Видалити». Після цих дій інформація про користувача все одно збережеться в системі, але переглянути інформацію про нього або відновити його буде вже неможливо. Для того, щоб змінити пароль користувачеві, необхідно в таблиці з користувачем вписати новий пароль.

3.3 Розробка бази даних

Опишемо проектування БД. Майже в кожній інформаційній системі є база даних. База даних дозволяє зберігати дані у структурованому вигляді. БД складається з таблиць – основних об'єктів бази даних, і навіть зв'язків з-поміж них. Таблиці складаються з полів – стовпці таблиці та записів – рядки таблиці.

Бази даних дозволяє створювати запити – це спеціальний засіб, за допомогою якого можна отримати інформацію з БД. Запити можуть бути відфільтровані за певними критеріями.

Кожна таблиця повинна містити стовпець або набір стовпців, які однозначно визначають кожен рядок таблиці. Таким унікальним записом може стати номер паспорта або код співробітника.

Таблиці у даних можуть бути пов'язані між собою, тобто, бути залежними. У реляційних базах даних існує три види зв'язків між таблицями:

- зв'язок один до одного;
- зв'язок один до багатьох;
- зв'язок багато хто до багатьох.

Використовувану БД зображено на рисунку 3.6.

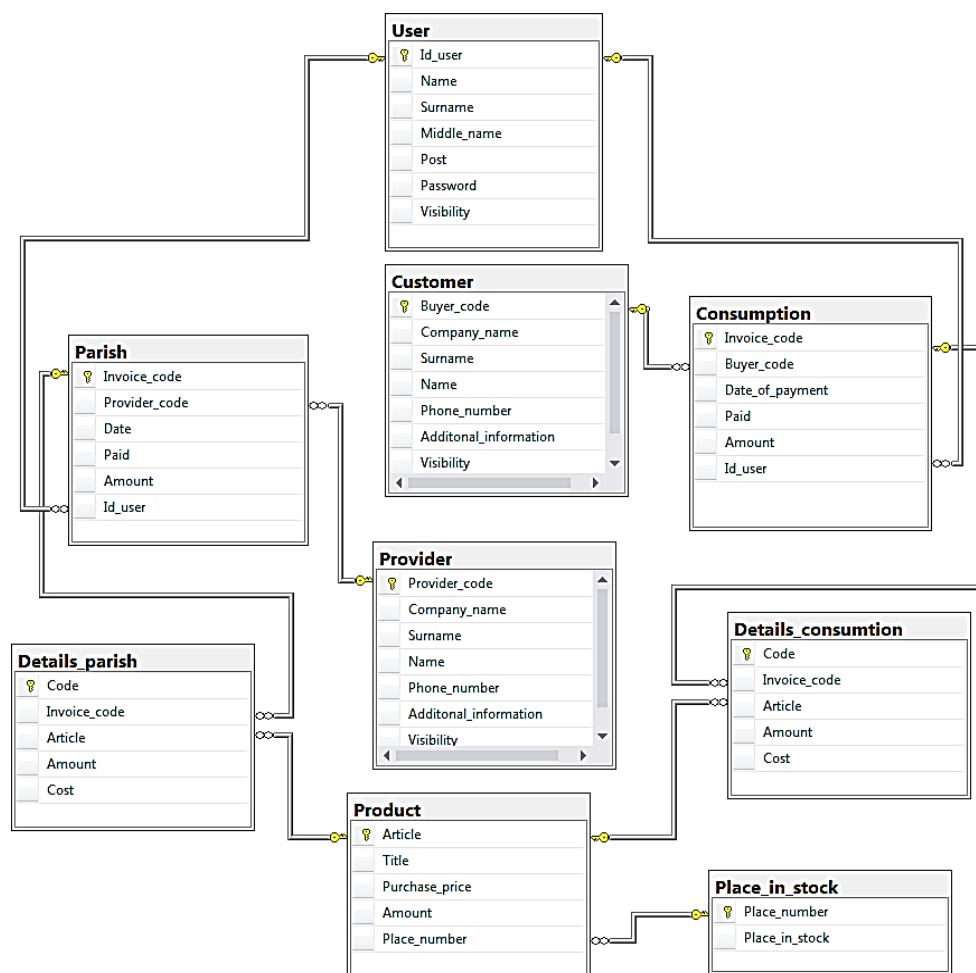


Рисунок 3.6 – Діаграма БД

На рисунку зображено 9 сутностей та зв'язки між ними:

- User – інформація про комірників та адміністраторів;

- Consumption – інформація про документ витрати;
- Details_consumption – інформація про витрату конкретного продукту;
- Product – інформація про продукт на складі;
- Parish – інформація про документ приходу;
- Details_parish – інформація про прихід конкретного продукту;
- Provider – дані про постачальника;
- Customer – дані про покупця;
- Place_in_stock – місце зберігання продукту.

Один із найрізкісніших зв'язків – це зв'язок друг до одного. Вона реалізується тоді, коли одному об'єкту таблиці відповідає один об'єкт іншої таблиці. Наприклад, людина та паспорт.

Для коректної роботи інформаційної системи необхідно передбачити авторизацію працівників складу. Також правильним рішенням буде відстеження фізичного місцезнаходження матеріалів на складі для швидкого доступу до них. При прийомі та видачі товару необхідно зберегти важливу інформацію, таку як дата видачі, кількість, вартість, ім'я працівника, який видав або прийняв товар, а також чи оплачена ця операція. Щоб реалізувати це, необхідно зберігати інформацію у базі даних.

Висновки до розділу 3

У даному розділі описано етапи розробки автоматизованої системи обліку товару на складі. Здійснено детальний розбір системи, де зазначено, що вона включає модулі для управління прийомом товарів, їх зберігання, інвентаризацію та відвантаження.

Наявний зручний інтерфейс користувача, що дозволяє легко додавати та редагувати записи. Передбачено функції звітності для моніторингу залишків і аналітики складських операцій.

Архітектура системи є модульною для полегшення майбутніх оновлень та масштабування. Інтеграція з іншими системами через API забезпечує безперервний обмін даними.

Детально описано етапи розробки бази даних.

Розроблена система обліку матеріалів на складі працює за принципом централізованого управління складськими операціями. Користувачі можуть вводити дані про надходження товарів, їх розміщення на складі та відвантаження замовлень через зручний інтерфейс. Система автоматично оновлює записи в базі даних, забезпечуючи актуальність інформації про запаси.

ВИСНОВКИ

Розробка автоматизованої системи обліку матеріалів на складі на базі мови програмування C# є важливим кроком для підвищення ефективності та точності складських операцій. Така система спрямована на зменшення людських помилок, оптимізацію ресурсів та підвищення швидкості обробки даних. Використання сучасних технологій та інструментів дозволяє створити надійне та масштабоване рішення, яке відповідає потребам сучасного бізнесу.

Аналіз предметної області показує необхідність інтеграції з іншими системами для забезпечення безперервності бізнес-процесів. Вибір правильних інструментальних засобів, таких як .NET, Visual Studio та SQL Server, є критично важливим для успішної реалізації проекту. Проектування та впровадження автоматизованої системи вимагає ретельного планування, тестування та навчання персоналу. Розроблена система повинна забезпечити зручний інтерфейс, високу продуктивність та безпеку даних. Загалом, впровадження такої системи сприяє підвищенню конкурентоспроможності компанії та ефективному управлінню складськими запасами.

Автоматизована система обліку матеріалів на складі є важливою для оптимізації складських операцій, включаючи прийом, зберігання, інвентаризацію та відвантаження товарів. Вона повинна забезпечити точне та своєчасне оновлення інформації про запаси, зменшуючи ризик людських помилок.

Аналіз предметної області виявив необхідність інтеграції з існуючими бізнес-системами, такими як ERP для підвищення ефективності. Важливими характеристиками системи є її здатність масштабуватися, враховуючи збільшення обсягів товарообігу. Система має зручний інтерфейс користувача для полегшення навігації та виконання завдань. Окрім цього, велике значення має безпека даних, щоб запобігти несанкціонованому доступу. Враховуючи всі ці аспекти, система відповідає вимогам сучасного складського господарства.

Для розробки автоматизованої системи обліку матеріалів на складі на базі мови програмування C# було використано платформу .NET, яка забезпечує стабільність і високу продуктивність. Visual Studio є оптимальним середовищем розробки завдяки своїм потужним інструментам для написання, тестування та налагодження коду. Для збереження даних використано Microsoft SQL Server, який забезпечує надійність та масштабованість баз даних. Використання цих інструментів сприяло створенню ефективної та надійної системи.

Проектування автоматизованої системи обліку матеріалів на складі починалося з визначення вимог та функціональних характеристик. Система включає модулі для управління прийомом товарів, їх зберігання, інвентаризацію та відвантаження. Наявний зручний інтерфейс користувача, що дозволяє легко додавати та редагувати записи. Передбачено функції звітності для моніторингу залишків і аналітики складських операцій. Архітектура системи є модульною для полегшення майбутніх оновлень та масштабування. Інтеграція з іншими системами через API забезпечує безперервний обмін даними. Забезпечення безпеки та конфіденційності даних є одним з ключових аспектів проектування.

Реалізація автоматизованої системи обліку матеріалів на складі на базі мови програмування C# включала кілька етапів. Спочатку створювалася база даних у Microsoft SQL Server для збереження інформації про товари, місцезнаходження та операції. Використовуючи Entity Framework, налаштовується зв'язок між базою даних та додатком. На основі WPF або ASP.NET розроблявся інтерфейс користувача, що забезпечує доступ до всіх необхідних функцій. Виконувалася імплементація основних модулів системи, таких як прийом товарів, інвентаризація, відвантаження та звітність. Після завершення розробки проводилося тестування та налагодження для виявлення та виправлення помилок.

Розроблена система обліку матеріалів на складі працює за принципом централізованого управління складськими операціями. Користувачі можуть вводити дані про надходження товарів, їх розміщення на складі та

відвантаження замовлень через зручний інтерфейс. Система автоматично оновлює записи в базі даних, забезпечуючи актуальність інформації про запаси. Інвентаризація здійснюється швидко і точно. Вбудовані інструменти звітності дозволяють відстежувати рух товарів та аналізувати ефективність складських операцій. Система також підтримує інтеграцію з іншими бізнес-додатками, що дозволяє забезпечити безперервність бізнес-процесів. Важливим аспектом є забезпечення безпеки даних через багаторівневу аутентифікацію та шифрування.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кащена Н. Б., Носач Н. М. Облік реалізації товарів в підприємствах торгівлі. Економіка і суспільство. Випуск # 7. URL: https://economyandsociety.in.ua/journals/7_ukr/154.pdf (дата звернення: 27.02.2024).
2. Пристемський О. С., Сакун А. Ж. Теоретико-методологічні аспекти обліку товарних операцій на підприємствах торгівлі. Збірник наукових праць Таврійського державного агротехнологічного університету імені Дмитра Моторного (економічні науки). Том 1 № 47 (2023). URL: <https://oj.tsatu.edu.ua/index.php/zbirnyk/article/view/539> (дата звернення: 29.02.2024).
3. Головна сторінка програми «УкрСклад». URL: <https://www.ukrsklad.com/> (дата звернення: 03.03.2024).
4. Головна сторінка програми «IBS». URL: <https://www.ibsystems.com.ua/ua/> (дата звернення: 03.03.2024).
5. Пристемський О. С. Організаційно-правові засади бухгалтерського обліку на підприємствах торгівлі. Проблеми обліково-аналітичного забезпечення управління підприємницькою діяльністю : матеріали IV Міжнар. наук.-практ. конф. (м. Полтава, 20-21 квітня 2022 р.). Полтава: ПДАУ, 2022. С. 183-185.
6. Stock Analysis System Use Case Diagram. URL: <https://creately.com/diagram/example/ip68h0hc2/stock-analysis-system-use-case-diagram-classic> (дата звернення: 25.03.2024).
7. Що має знати C# .net розробник у 2023 році. URL: <https://itvdn.com/ua/blog/article/cs-dnet-23> (дата звернення: 05.04.2024).
8. What's new in C# 11. URL: <https://learn.microsoft.com/en-us/dotnet/csharp/whats-new/csharp-11> (дата звернення: 05.04.2024).

9. Microsoft Visual Studio: матеріал Вікіпедії. URL: https://uk.wikipedia.org/wiki/Microsoft_Visual_Studio (дата звернення: 25.04.2024).

10. Microsoft SQL Server: матеріал Вікіпедії. URL: https://uk.wikipedia.org/wiki/Microsoft_SQL_Server (дата звернення: 05.05.2024).

11. Марченко А. В. Проектування інформаційних систем. URL: http://kist.ntu.edu.ua/textPhD/PIS_Marchenko.pdf (дата звернення: 15.05.2024).

12. MVP design pattern diagram. URL: https://www.researchgate.net/figure/MVP-design-pattern-diagram_fig3_348792766 (дата звернення: 25.05.2024).