

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) Автоматизації і комп'ютерних систем

Кафедра Інформаційних технологій, штучного інтелекту і кібербезпеки

Освітній ступінь бакалавр

Спеціальність 122 Комп'ютерні науки

Освітньо-професійна програма Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інформаційних технологій, штучного інтелекту і кібербезпеки

Сергій ГРИБКОВ

“ 20 ” листопада 2023 року

З А В Д А Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА

Олійнику Вячеславу Валерійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення системи інформаційної підтримки діяльності підприємства з виготовлення металоконструкцій

керівник роботи М'якшило Олена Михайлівна, доц., к.т.н.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від 14 листопада 2023 року №924-кв

2. Строк подання здобувачем роботи 20 січня 2024 р.

3. Вихідні дані до роботи

Статут підприємства. Відомості про надходження матеріалів.

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

Системний аналіз діяльності підприємства; розрахунок очікуваного економічного ефекту від впровадження системи; технічне завдання; розроблення інтерфейсу ІС; розроблення інструкції користувача; опис заходів з охорони праці та техніки безпеки.

5. Перелік графічного матеріалу

1. Організаційна структура підприємства

2. Функціональна модель діяльності підприємства

3. Логічна та фізична моделі бази даних

4. Приклади роботи ІС (інтерфейс користувача)

5. Фрагменти коду

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	к.т.н., доц. М'якшило О.М.		
2	к.т.н., доц. М'якшило О.М.		
3	к.т.н., доц. М'якшило О.М.		

7. Дата видачі завдання _____ 20 листопада 2023 року _____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів виконання кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Системний аналіз діяльності підприємства та постановка задачі на проектування	27.11.2023	Виконано
2	Розроблення функціональної моделі	02.12.2023	Виконано
3	Аналіз систем-аналогів розробки	05.12.2023	Виконано
4	Створення технічного завдання	07.12.2023	Виконано
5	Проектування БД	11.12.2023	Виконано
6	Створення ІС	13.12.2023	Виконано
7	Написання інструкцій користувача	23.12.2023	Виконано
8	Оформлення пояснювальної записки	04.01.2024	Виконано
9	Оформлення презентації	26.01.2024	Виконано

Здобувач _____
(підпис)

Керівник роботи _____
(підпис)

Вячеслав ОЛІЙНИК _____
(прізвище та ініціали)

Олена М'ЯКШИЛО _____
(прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота «Розроблення системи інформаційної підтримки діяльності підприємства з виготовлення металоконструкцій».

Автор проекту: Олійник В.В.

Обсяг – 86 сторінок, 22 рисунків, 11 таблиць, 2 блок-схеми, 7 формул, 7 додатків, 12 джерел.

Мета кваліфікаційної роботи: розробка програмного забезпечення для підтримки діяльності підприємства з виготовлення металоконструкцій.

У кваліфікаційна робота проведено аналіз інформаційного забезпечення предметної області та інформаційних потоків, розроблена схема бази даних. Для розробки програмного продукту використано мову С# та Microsoft Access database. За допомогою цих засобів розроблено програмне забезпечення для автоматизації роботи фірми.

Впровадження розробленого програмного продукту дозволяє автоматизувати діяльності персоналу підприємства.

Ключові слова: VISUAL STUDIO, С#, Microsoft Access database.

ANNOTATION

Diploma project «Development of a system of information support for the activity of the enterprise for the production of metal structures».

Author of the project: Oliynyk V.V.

Volume - 86 pp., 22 Fig., 11 table., 2 block diagram, 7 formulas, 7 appendices, 12 sources.

The purpose of the thesis project is to develop software to support the activities of a metal fabrication company.

The thesis project analyzed the information support of the subject area and information flows, and developed a database scheme. The C# language and Microsoft Access database were used to develop the software product. These tools were used to develop software to automate the company's work.

The implementation of the developed software product allows you to automate the activities of the company's personnel.

VISUAL STUDIO, C#, Microsoft Access database.

ЗМІСТ

ВСТУП	8
1 СИСТЕМНИЙ АНАЛІЗ ОБ'ЄКТА ДОСЛІДЖЕННЯ ТА ВИЯВЛЕННЯ ЗАДАЧ АВТОМАТИЗАЦІЇ	10
1.1 Опис предметного середовища	10
1.1.1 Загальна схема організаційної структури	11
1.2. Розроблення функціональної моделі та аналіз існуючих процесів	15
1.3 Аналіз існуючих аналогів	18
1.3.1 М.Е.ДОС	18
1.3.2 Дебіт Плюс	19
1.3.3 Облік SaaS	20
1.3.4 Dilovod	22
1.4 Порівняльна таблиця	23
1.5 Розрахунки економічної частини	26
1.5.1 Економічна доцільність проекту	33
2 ТЕХНІЧНЕ ЗАВДАННЯ	34
2.1 Постановка задачі	34
2.2 Вимоги до системи	34
2.3 Склад і зміст робіт зі створення системи	35
2.4 Порядок контролю і приймання системи	36
2.5 Вимоги до складу та змісту робіт з підготовки об'єкта автоматизації до введення системи в дію	36
2.6 Вимоги до документування	37
2.7 Вимоги до функцій	37
2.8 Джерела розробки	37

3 ОПИС КОМПЛЕКСУ ЗАДАЧ АВТОМАТИЗАЦІЇ.....	39
3.1 Вибір програмного та апаратного забезпечення	39
3.1.1 C#	39
3.1.2 WPF	40
3.1.3 .NET Framework	41
3.1.4 .NET Core	41
3.2 Проектування власного програмного додатку	42
3.2.1 Вікно авторизації користувача.....	47
3.2.2 Головне вікно програми	49
3.2.3 Вікно редагування даних	58
3.2.4 Вікно керування адміністраторами.....	58
3.3 Інструкція користувача	62
4. ОХОРОНА ПРАЦІ	64
ВИСНОВКИ	75
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	76
ДОДАТКИ	

ВСТУП

Кваліфікаційна робота є невід'ємною складовою освітньо-професійної програми підготовки студентів. Вона становить важливу та обов'язкову ланку в підготовці висококваліфікованих спеціалістів до майбутньої діяльності за фахом. Кваліфікаційна робота спрямований на закріплення теоретичних знань отриманих студентами за час навчання, та набуття, вдосконалення і розширення практичних навичок і умінь в роботі за обраною спеціальністю.

Таке поняття як бази даних та програмне забезпечення для їх розробки використовується на персональних комп'ютерах досить давно. База даних – це сукупність даних, організованих відповідно до концепції, яка описує характеристику цих даних і взаємозв'язки між їх елементами. За допомогою такої системи можна зберігати різні дані наприклад: номери телефонів, прізвища, імена, листи або адреси. Один з типів баз даних являється текстовими документами, які набрані за допомогою текстових редакторів і об'єднанні в групи. Другий тип – файли електронних таблиць, які об'єднанні в групи по характеру їх використання. Для того, щоб управляти такими даними, які розкидані по сотням таблиць і файлів, які потрібні для використання базами даних потрібна система керування базами даних. Однією з таких систем являється Microsoft Access database.

Тема кваліфікаційної роботи – створення програмного додатку для підтримки діяльності підприємства з виготовлення металоконструкцій.

Microsoft Access database — полегшена реляційна система керування базами даних. Втілена у вигляді бібліотеки, де реалізовано багато зі стандарту SQL-92. Сирцевий код Microsoft Access database поширюється як суспільне надбання, яке входить у пакет Microsoft Office 365.

Для кваліфікаційної роботи вибрана система Microsoft Access database, через її простоту у використанні та легкі функції при створенні таблиць, форм, запитів та макросів, саме це дає змогу легко та швидко створювати і ефективно оновлювати дані. Усі дані з різних джерел зберігаються в окремій таблиці і при

роботі з ними створюються зв'язки. Для додавання нових даних або видалення їх використовується форма. Для пошуку та відбору створюються запити.

При створенні бази даних потрібно звернути увагу на систему збереження і оновлення даних, а також обґрунтувати зв'язки між даними та забезпечити їх узгодженість. Бази даних постійно оновлюються з врахуванням вимог основних користувачів бази даних.

На сьогоднішній день, більшість організацій та різних компаній використовують бази даних. Це дає змогу значно пришвидшити роботу підприємств.

Програмне забезпечення, яке буде керувати базою даних буде створене на мові програмування C#, використовуючи середовище розробки програмного забезпечення Visual Studio 2017.

Microsoft Visual Studio — серія продуктів фірми Майкрософт, які містять інтегроване середовище розробки програмного забезпечення та низку інших інструментальних засобів. Ці продукти дають змогу розробляти як консольні програми, так і програми з графічним інтерфейсом, включно з підтримкою технології Windows Forms, а також веб-сайти, веб-застосунки, веб-служби як в рідному, так і в керованому кодах для всіх платформ, що підтримуються Microsoft Windows, Windows Mobile, Windows Phone, Windows CE, .NET Framework, .NET Compact Framework та Microsoft Silverlight.

Visual Studio включає один або декілька з наступних компонентів:

Visual Basic .NET, а до його появи — Visual Basic.

Visual C++.

Visual C#.

Visual F# (входить до складу Visual Studio 2010).

Visual Studio Debugger.

1 СИСТЕМНИЙ АНАЛІЗ ОБ'ЄКТА ДОСЛІДЖЕННЯ ТА ВИЯВЛЕННЯ ЗАДАЧ АВТОМАТИЗАЦІЇ

1.1 Опис предметного середовища

Фірма для якої розробляється програмний продукт називається VESCO.

Підприємство є підрядною організацією та займає лідируючі позиції в розробці та застосуванні прогресивних проектних рішень, нових технологій, металокопструкцій та спеціальної техніки в енергетичному та транспортному будівництві.

Підприємство здійснює виготовлення та монтаж металокопструкцій для енергетичних комплексів (АЕС, ГАЕС), захисних дамб, мостів та автомобільних доріг. Підприємство має в управлінні щебеневий кар'єр.

Зберігаючи та розвиваючи традиції енергетичного та транспортного будівництва, VESCO веде активну діяльність у різних регіонах країни, а саме: закінчення виготовлення та монтажу металокопструкцій Ташлицької ГАЕС, Канівської ГЕС, за що отримало відповідний сертифікат, реконструкції автомобільної дороги в обхід Ташликського водосховища. Крім того, продовжує роботи з виготовлення і монтажу металокопструкцій для Дністровської ГАЕС та ряду інших об'єктів.

VESCO гарантує задоволення вимог замовників по якості проектування, виготовлення та монтажу металокопструкцій, надійності та довговічності збудованих споруд.

Високий професійний рівень інженерно-технічних працівників та робочих кадрів дозволяє підприємству здійснювати масштабні проекти – від зародження ідеї до її повної реалізації.

При цьому основа стратегії VESCO залишається незмінною: відмінна якість, дотримання строків за рахунок високопродуктивної праці.

ПРИНЦИПИ ТА НАПРЯМИ ДІЯЛЬНОСТІ

- Фірма VESCO захищає інтереси національного виробника - компаній з українським чи іноземним капіталом, які створюють в Україні робочі місця, виробничі потужності та матеріально-технічну базу, сплачують податки в Україні, підтримають професійну освіту, реалізують важливі соціальні проекти.
- Фірма VESCO є незалежною від державних органів, але має на меті створення конструктивного діалогу із державою для пошуку спільних ефективних рішень у напрямках розвитку ринку з урахуванням інтересів держави, споживачів та учасників ринку.
- Фірма VESCO не просуває корпоративні інтереси будь-кого зі своїх членів, але захищає національного виробника у випадку безпідставного та необґрунтованого тиску на нього із боку державних структур.
- Фірма VESCO є неприбутковим об'єднанням і не надає комерційних послуг.
- Фірма VESCO проводить роз'яснювальну роботу серед громадськості та ЗМІ, спрямовану на правдиве висвітлення проблематики сфери дорожнього будівництва.
- Позиція Асоціації формується на основі консенсусу її членів.

КЕРІВНИЦТВО АСОЦІАЦІЇ

Березовський Микола Володимирович - Президент

Бойко Олександр Леонідови - Віце-президент

Варивода Олег Миколайович - Виконавчий директор

1.1.1 Загальна схема організаційної структури

На підприємстві використовується лінійно-функціональна структура управління. На вершині ієрархії знаходиться директор (рисунок 1.1). Він здійснює загальне керівництво підприємством і несе повну відповідальність за його діяльність.

Підрозділ маркетингу очолює начальник відділу маркетингу. Йому підпорядковуються працівники відділу маркетингу.

Бухгалтерію очолює головний бухгалтер. Йому підпорядковуються бухгалтери та помічник головного бухгалтера.

Технічний відділ очолює керівник. Йому підпорядковуються інженери та координатор проектів.



Рисунок 1.1 Верхній рівень організаційної структури VESCO

Кожен працівник майстерні має свої обов'язки залежно від посади.

Менеджер закупок відповідає за закупівлю всіх необхідних металодеталей для виготовлення конструкцій. Він також забезпечує майстерню всіма необхідними інструментами, обладнанням та іншими важливими для діяльності підприємства товарами.

Інженер з виготовлення проводить усі необхідні операції, які безпосередньо стосуються виготовлення конструкцій з металу. До цих робіт належать: розробка проектів конструкцій, пошук необхідних металодеталей, виготовлення конструкцій, перевірка якості виготовлення.

Бухгалтер здійснює контроль за грошовими потоками підприємства, забезпечує ведення фінансової звітності та бухгалтерського обліку, відповідає за нарахування зарплати співробітникам.

Робота менеджера закупок та інженерів є ключовими у діяльності майстерні з виготовлення металоконструкцій, оскільки вони виконують основну

роль підприємства – реалізацію надання послуг виготовлення конструкцій клієнтам.

У таблиці 1.1 представлені завдання та функції на фірмі.

Таблиця 1.1 – Завдання і функції на фірмі

№	Задачі	Функції
1	Забезпечення якісного проведення виготовлення металоконструкцій	Контроль за дотриманням чистоти, правил і норм охорони праці, техніки безпеки при виконанні робіт з виготовлення металоконструкцій.
	Вибір відповідного обладнання та інструментів для виготовлення металоконструкцій.	Вибір матеріалів та металодеталей для виготовлення металоконструкцій.
2	Забезпечення якісної комунікації з клієнтами	Оформлення квитанції про прийом металоконструкцій.
3	Організація оплати за виготовлені металоконструкції	Розрахунок загальної вартості виготовлених металоконструкцій.

У таблиці 1.2 представлено взаємодія підрозділів на фірмі.

Таблиця 1.2 – Взаємодія підрозділів в фірмі

№	Підрозділ	Одержання	Надання
1	Директор	– звітів про фінансової майстерні; – необхідні відомості, пов'язані з діяльністю фірми.	– наказів по усіх напрямках майстерні; – функцій і завдань структурних підрозділів;

			– посадових інструкцій, виробничих інструкцій, інших документів.
2	Бухгалтер	<ul style="list-style-type: none"> – фіскальний звітний чек; – заявку на підбір необхідних спеціалістів; – таблиць виходу на роботу; – підсумків інвентаризації матеріальних ресурсів. 	<ul style="list-style-type: none"> – заробітної плати та відпускних; – документів щодо нарахування та виплати зарплати; – графіків відпусток; – відібраних спеціалістів із зазначеними вміннями в поданій заявці.
3	Менеджер заcuпок	<ul style="list-style-type: none"> – розрахунків потреб і заявок на матеріально-технічні ресурси; – звітів про використання матеріально-технічних ресурсів; – відомостей про залишки матеріально-технічних ресурсів. 	<ul style="list-style-type: none"> – відомостей про наявність матеріально-технічних ресурсів на складі; – планів постачання матеріально-технічних ресурсів; – звітів про виконання планів матеріально-технічного постачання.
4	Інженер	<ul style="list-style-type: none"> – замовлення. – управління працівниками. 	– звітність по виконаному ремонту техніки.



Рисунок 1.2 – Логотип компанії

Контактна інформація:

03150, М.КИЇВ, ВУЛ.АНТОНОВИЧА, 30

+380 (97) 362 97 92

<https://www.faceMaterials.com/NADU.com.ua>

office.nadu@gmail.com

1.2. Розроблення функціональної моделі та аналіз існуючих процесів

Головною метою створення функціональної моделі процесу виготовлення металоконструкцій є точний опис всіх функцій, здійснюваних в рамках процесу вищого рівня ієрархії, а також характеру взаємозв'язків між ними. Функціональна модель дозволяє чітко визначити розподіл ресурсів між операціями ділового процесу, що дає можливість оцінити ефективність їх використання.

Мета моделювання: Визначити функції, завдання та процеси виготовлення металоконструкцій, що потребують автоматизації.

Межі моделювання: Починаючи від отримання замовлення на виготовлення металоконструкцій до їхнього відвантаження замовнику та оформлення звітної документації.

Використовуючи BPWin (AllFusion ProcessModeler7) можна створити детальні діаграми щодо всіх важливих аспектів бізнес-процесу, а саме необхідних операцій, способу впровадження та контролю необхідних для цього ресурсів, а потім візуалізувати отриману інформацію.

Для аналізу та дослідження видів робіт, які виконуються під час процесу виготовлення металоконструкцій, було розроблено функціональну модель, яка представлена на рисунку 1.4.



Рисунок 1.3 – Діаграма верхнього рівня

Після декомпозиції загальної моделі бізнес-процесів процес виконання замовлення розбивається на етапи, як показано на рисунку 1.4.

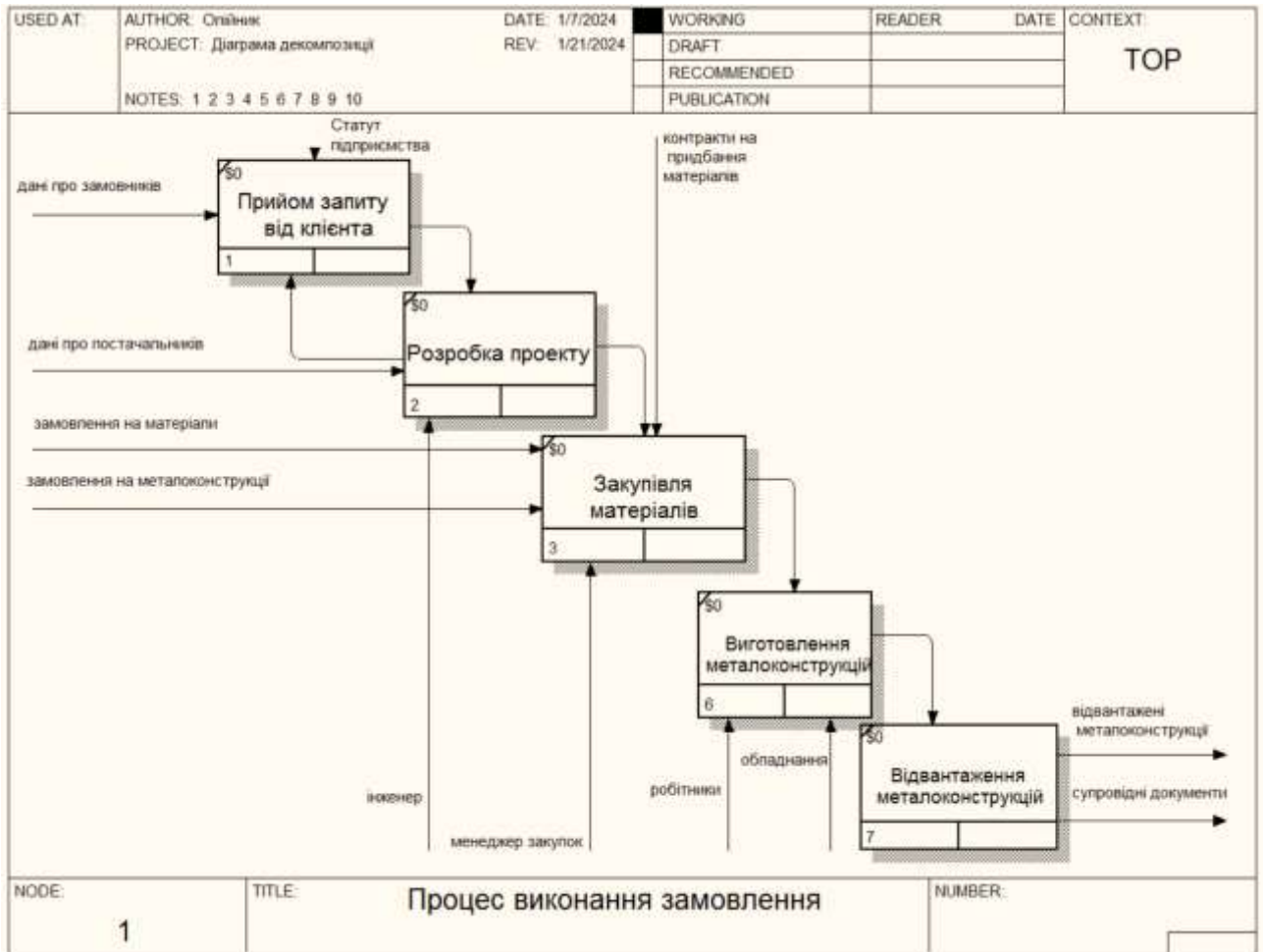


Рисунок 1.4 – Діаграма декомпозиції

Весь процес виконання замовлення описаний так:

Приєм запити від клієнта

- Консультування клієнта
- Реєстрація замовлення
- Приєм даних про металокопструкції

Розробка проекту

- Розробка технічного завдання
- Розробка конструкторської документації
- Узгодження проекту з клієнтом

Закупівля матеріалів

- Визначення потреб у матеріалах
- Закупівля матеріалів

Виготовлення металокопструкцій

- Розкрій матеріалів

- Зварювання
- Обробка металоконструкцій
- Контроль якості

Відвантаження металоконструкцій

- Формування комплекту документів
- Відвантаження металоконструкцій клієнту

1.3 Аналіз існуючих аналогів

1.3.1 M.E.DOC

My Electronic Document, часто називана Медок. Ця українська бухгалтерська програма дозволяє легко обмінюватися документами з контрагентами та подавати звітність контролюючим органам. Давайте розглянемо її функціональні можливості:

Функціонал My Electronic Document (Медок):

- **Універсальність:** Медок придатний для використання підприємцями будь-якого виду діяльності та різною системою оподаткування.
- **Модульна структура:** Програма має модульну структуру, що дозволяє налаштувати індивідуальний функціонал залежно від потреб підприємства. Можливості включають в себе облік кадрів, розрахунок зарплати, підключення СОТА (Системи обов'язкового та централізованого обліку та звітності), реєстрацію акцизних накладних та транспортних товарних накладних (ТТН), а також проведення повного циклу роботи з розрахунковими даними та податковими накладними.
- **Електронний обмін:** Медок спрощує процес обміну електронними документами з контрагентами та підтримує онлайн-подання звітності.

Вартість та передплата:

- **Вартість встановлення та обслуговування програми** залежить від її конфігурацій і може становити до 12 000 гривень на рік при першому зверненні.
- **Тривалість передплати:** Користувачі можуть обирати передплату на квартал, півроку або рік.

Електронна звітність:

Медок дозволяє подавати звітність онлайн до всіх контролюючих органів.

Ця програма може бути привабливою альтернативою для підприємств, які шукають заміну вигнаної 1С та бажають зберігати ефективний облік і звітність.

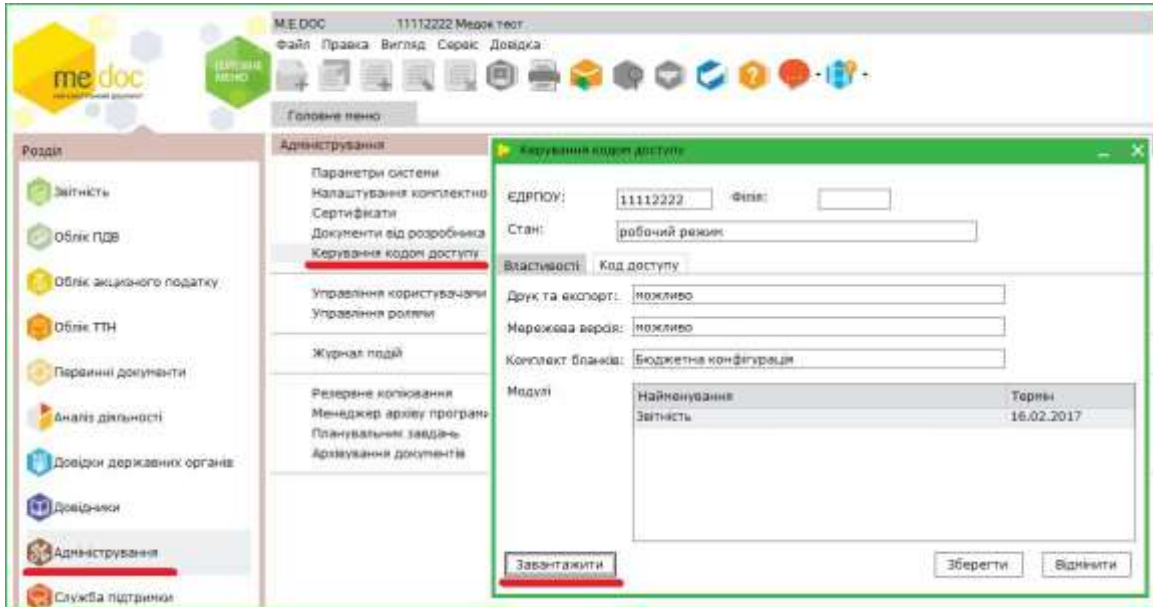


Рисунок 1.5 – My Electronic Document

1.3.2 Дебіт Плюс

Ця система пристосована до постійних змін у законодавстві, і вона легко налаштовується навіть користувачами без великого досвіду - для цього не потрібний програміст. Навчання роботі з Дебіт Плюс є простим і не забирає багато часу, оскільки воно включає в себе відеоуроки, які входять до комплексу.

Функціонал:

- Робота з оплатою праці, тарифами, автоматичними доплатами.
- Формування довідок та експорт до банку.
- Робота з матеріальними цінностями, запасами, залишками.
- Оборотно-сальдові відомості та картки обліку.
- Робота з бухгалтерським балансом та формування форм, виписок, головної данихи.

- Робота з основними засобами, інвентарними картками, актами, нарахуванням зносу.
- Робота з касовими та банківськими операціями, журналом реєстрації, авансовими звітами, касовими ордерами.

Тарифи:

Вартість розраховується індивідуально в залежності від конфігурації системи.

Тривалість передплати:

Ви можете обирати зручний спосіб оплати: щомісяця, раз на півроку, щорічно або одноразово безстроково.

Електронна звітність:

Ця система не підтримує електронну звітність

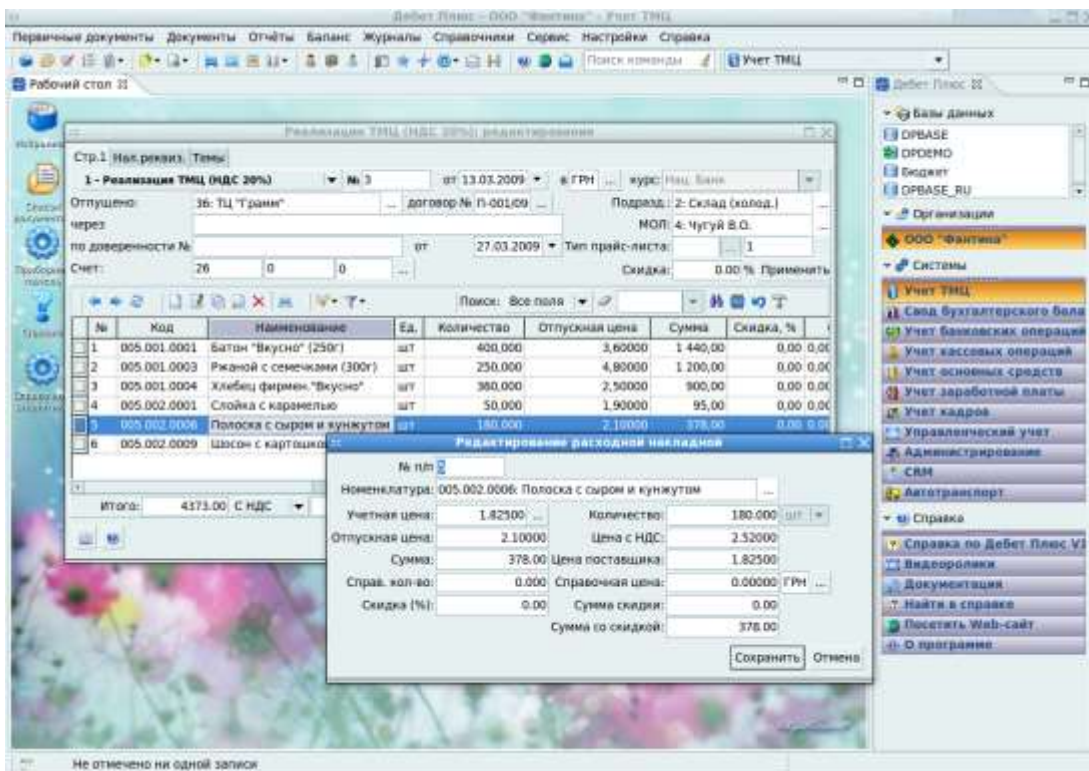


Рисунок 1.6 – Дебет Плюс

1.3.3 Облік SaaS

Цей хмарний сервіс призначений для бухгалтерів і надає можливість вести різні види обліку в єдиному місці. Функціональність програми залежить від обраної версії, яка підходить для різних видів підприємств.

- Стандартна версія: Відмінно підходить для малих та середніх підприємств.
- Корпоративна версія: Стане незамінним помічником для великих корпорацій.
- Галузева версія: Містить спеціалізований функціонал для нішевих підприємств.

Програма дозволяє вести облік діяльності компаній з України, Росії та Казахстану.

Функціонал:

- Інтеграція з CRM: Облік SaaS інтегрується з CRM, що дозволяє вести облік подвійним записом.
- Консолідована зведена звітність: У програмі можна вести консолідовану зведену звітність.
- Зашифрована передача даних: Дані SSL передаються зашифрованим протоколом, забезпечуючи безпеку і конфіденційність інформації.
- Різні комерційні операції: В обліку SaaS можна проводити різні комерційні операції, включаючи касові та банківські операції, а також розрахунок виплат співробітникам та роботу з клієнтами.

Тарифи:

- Малий бізнес: Розробники пропонують безкоштовну версію.
- Інші підприємства: Вартість становить 300 гривень на місяць.

Тривалість передплати: Рахунок поповнюється раз на місяць.

Електронна звітність: Система не підтримує електронну звітність

Счет	Назначение счета	Валюта	Начальный сальдо		Оконч. сальдо	
			Дебет	Кредит	Дебет	Кредит
0.001.001	Розрахунок з отриманими послугами роботи поштою	UAH	1 070.00	1 000.00	70.00	
0.002.010	Розрахунок з отриманими послугами поштою по Україні	UAH	1 070.00		1 070.00	
0.001.010	Розрахунок з отриманими послугами за кордоном	EUR		300.00		300.00
0.001.002	Розрахунок з отриманими послугами за кордоном	USD		1 250.00	1 000.00	250.00
		EUR		23 019.41		23 019.41
		USD		70.00		70.00
		UAH	4 803.89	880.00		4 803.89
		EUR		100.00		100.00
		USD		1 070.00		1 070.00
		UAH	1 070.00			1 070.00
		EUR	4 423.89	41 908.19	2 080.00	36 327.19
		USD				
		UAH	4 075.00	40 718.19	1 140.00	37 977.19

Рисунок 1.7 – Облік SaaS

1.3.4 Dilovod

Розробники стверджують, що їх програма Dilovod - це не просто конкурент 1С, але і інструмент, який дозволяє підприємцям легко і без особливих знань у галузі бухгалтерії вести звітність. Подивимося ближче на головні характеристики цієї програми:

Функціонал Dilovod:

- Об'єднання бухгалтерського та управлінського обліків, контроль товарообігу, реєстрація платників єдиного внеску (пРРО) та можливість подавати електронну звітність - усі ці можливості в одній програмі.
- Можливість ведення обліку кількох ФОП і юридичних осіб в одній базі даних.
- Автоматичне архівування копій документів і відображення всього грошового обігу в системі.
- Важливий бонус - автоматична перевірка наявності помилок у звітності.

Вартість та передплата:

Вартість програми Dilovod варіюється від 450 до 1050 гривень на місяць, залежно від обраного функціоналу.

Тривалість передплати: Передплата оформлюється на місяць.

Електронна звітність:

Можливість подавати звітність онлайн, якщо є доступ до Інтернету.

Dilovod може стати привабливою альтернативою для підприємців, які шукають зручний та функціональний інструмент для бухгалтерського обліку та звітності без значних зусиль на навчання бухгалтерським навичкам.

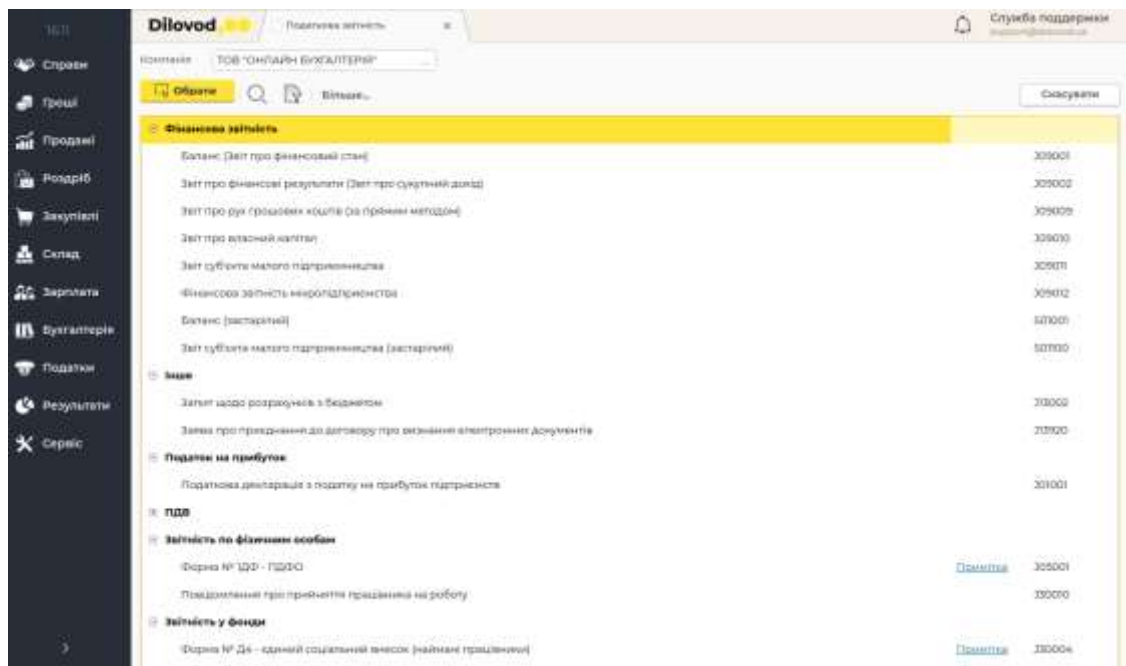


Рисунок 1.8 – Dilovod

1.4 Порівняльна таблиця

Таблиця 1.3 – Порівняльна таблиця існуючих аналогів

Характеристика	М.Е.ДОС	Дебіт Плюс	Облік SaaS	Dilovod
Ціна	Від 1000 до 10 000 грн/міс	Від 249 до 1299 грн/міс	Від 150 до 250 грн/міс	Від 199 до 299 грн/міс
Функціональність	Фінансовий облік, управління продажами, управління закупівлям	Фінансовий облік, управління продажами, управління закупівлям	Фінансовий облік, управління продажами, управління	Фінансовий облік, управління продажами, управління

	управління персоналом , CRM	управління персоналом , HRM	закупівлям и	закупівлям и
Мобільна версія	€	€	€	€
Онлайн-допомога	€	€	€	€
Відгуки користувачів	Позитивні	Позитивні	Позитивні-середні	Позитивні

Висновок

Всі чотири програми мають схожий функціонал та інтерфейс. Вони є доступними за ціною та мають мобільні версії. Відгуки користувачів про всі програми є позитивними.

Вибір програми залежить від індивідуальних потреб та вимог користувача.

Якщо для вас важлива ціна, то Dilovod або Облік SaaS будуть хорошим вибором. Якщо ви шукаєте програму з широким функціоналом, то М.Е.ДОС або Дебіт Плюс будуть більш відповідними варіантами.

Особливості кожної програми

М.Е.ДОС - це комплексна бізнес-система, яка включає в себе широкий спектр модулів для управління різними аспектами діяльності підприємства. М.Е.ДОС має широкий функціонал, але його також досить складно освоїти.

Дебіт Плюс - це проста та інтуїтивно зрозуміла програма для фінансового обліку. Дебіт Плюс підходить для малого та середнього бізнесу.

Облік SaaS - це хмарна програма для фінансового обліку. Облік SaaS є доступним за ціною та має зручний інтерфейс.

Dilovod - це програма для фінансового обліку та управління продажами. Dilovod є простим та інтуїтивно зрозумілим.

1.5 Розрахунки економічної частини

Визначення доцільності розробки проекту з економічної точки зору є досить важливою стадією, оскільки розробка програмного забезпечення має бути корисним для VESCO. Даний програмний додаток має на меті збільшити зручність ведення списку матеріалів, шляхом економії витрат.

Для цього, потрібно провести наступні обчислення:

Таблиця 1.4 - Кошторис витрат на придбання обладнання, необхідного розробки програмного продукту введення обліку VESCO.

Назва обладнання, програм	Кількість	Ціна за одиницю, грн.	Загальна вартість, грн.	Примітка
1	2	3	4	5
Ноутбук Lenovo ideapad 320-15ikb	1	17472,00	17472,00	Придбано раніше
Windows 10	1	3600,00	3600,00	Придбано раніше
Visual Studio 2017	1	13756,00	13756,00	Придбано раніше
Microsoft Office 2016	1	3000,00	3000,00	Придбано раніше
Всього:	-	—	37 828,00	—

Розробка додатку складається з багатьох послідовних, цілеспрямованих, взаємозв'язаних та взаємообумовлених етапів, на які можна поділити весь часовий відрізок, що відводиться на розробку проекту . Такі етапи приведені у таблиці 1.5.

Таблиця 1.5 – Перелік етапів та робіт по розробці ПЗ.

Найменування		Вид роботи	Виконавець, посада
стадії	етапу		
1	2	3	4

1 Підготовча стадія	Вивчення питання	Дослідження проблеми	Програміст
		Огляд та аналіз існуючих аналогів	Програміст
	Вивчення питання	Економічне обґрунтування доцільності виконання проекту	Бухгалтер
		Аналіз технічного завдання	Узгодження завдання із зацікавленими сторонами
			Складання плану та розрахунок розвитку
	2 Технічна пропозиція	Аналіз та техніко-економічне обґрунтування	Доведення техніко-економічного обґрунтування
Доведення та уточнення загального обсягу робіт, строків виконання та витрат			Програміст, Бухгалтер
3 Теоретична розробка	Теоретичне вивчення задачі	Дослідження технічних особливостей проекту	Програміст
		Розробка структури проекту	Програміст
4 Практична реалізація	Проектування та покращення проекту	Розробка додатку	Програміст

Продовження таблиці 1.5

1	2	3	4
4 Практична реалізація	Проектування та покращення проекту	Проектування та покращення проекту	Програміст
	Налаштування системи та реалізація під інші платформи	Аналіз проекту та його налаштування під інші платформи	Програміст
5 Тестування, доробка системи та підготовка документації	Тестування всього проекту	Тестування проекту у реальних умовах	Програміст
		Доробка проекту з урахуванням результатів тестування	Програміст
	Підготовка документації по системі	Підготовка звіту про розробку проекту	Програміст
6 Заключна стадія	Ознайомлення та зацікавлення осіб з проектом	Підготовка презентації	Програміст
		Демонстрація	Програміст

На основі даного переліку визначається кількість виконавців, тривалість виконання робіт в годинах, що виконуються. Такі дані приведені у таблиці 1.6.

Таблиця 1.6 – Зведені результати тривалості та трудомісткості розробки програмного додатку.

Найменування роботи	Виконавець, посада, спеціальність	Кількість виконавців	Тривалість виконання роботи, год	Годинна тарифна ставка, грн	Оплата праці, грн
1	2	3	4	5	6
Дослідження проблеми, вибір платформи для реалізації проекту .	Програміст	1	5	60,55	302,75
Економічне обґрун-я доцільності виконання проекту , узгодження завдання із зацікавленими сторонами, складання плану та розрахунок розробки, доведення техніко-економічного обґрун-я, доведення та уточнення загального обсягу робіт, строків виконання та витрат	Бухгалтер, Програміст	1	6	58,70	352,20
		1	6	60,55	363,30
Теоретичне просування	Програміст	1	11	60,55	666,05
Практична реалізація	Програміст	1	78	60,55	4722,90
Тестування, доробка системи та підготовка документації	Програміст	1	27	60,55	1634,85
Заключна стадія	Програміст	1	16	60,55	726,6
Всього заробітна плата					8770,60

Витрати на науково-дослідницьку роботу по розробку додатку, щодо розробки кваліфікаційної роботи, включають наступні елементи:

- витрати на основну заробітну плату виконавців. Це сума часткових заробітних плат за кожну роботу і вона приведена в таблиці 4.3.

$$ЗП_{\text{основна}} = 8770,60 \text{ грн}$$

- додаткова заробітна плата. Її можна обчислити за формулою:

$$ЗП_{\text{додаткова}} = 0,2 \times ЗП_{\text{основна}} \quad (4.1)$$

$$ЗП_{\text{додаткова}} = 1754,13 \text{ грн}$$

Таким чином загальний фонд заробітної плати, що обчислюється за формулою:

$$\Phi ЗП = ЗП_{\text{основна}} + ЗП_{\text{додаткова}} \quad (4.2)$$

$$\Phi ЗП = 8770,60 + 1754,13 = 10524,73 \text{ грн.}$$

- єдиний соціальний внесок

$$\text{Відр} = 10524,73 \times 22 / 100 = 2315,44 \text{ грн.}$$

- накладні витрати розраховуємо за формулою:

$$\text{НВ} = 0,4 \times ЗП_{\text{основна}} \quad (4.3)$$

$$\text{НВ} = 8220,00 \times 0,4 = 3\,508,24 \text{ грн.}$$

Результати розрахунку інвестиційних витрат приведені в таблиці 1.7.

Таблиця 1.7 - Перелік інвестиційних витрат

Найменування елементу витрат	Сума, грн.
Капітальні затрати	-
Основна заробітна плата виконавців	8770,60
Додаткова заробітна плата виконавців	1754,13
Єдиний соціальний внесок	2315,44
Накладні витрати	3508,24
Витрат всього	16348,41

Загальні інвестиційні витрати (ЗІВ) на розробку проекту складають 16348,41 грн.

Визначимо витрати на амортизацію необхідного обладнання, програмного забезпечення за формулою:

$$Va = \frac{Вобл}{100} \times Na \quad (4.4)$$

Де Вобл – вартість обладнання, програмного забезпечення, грн.;

Na – норма амортизаційних відрахувань.

$$Va = (37828,00/100) \times 25 \times (23/251) = 378,28 \times 25 \times 0,0916 = 866,2612 \text{ грн.}$$

Таблиця 1.8 - Собівартість створення проекту для ДП ТзОВ «Імпульс».

Найменування елементу витрат	Сума, грн
Основна заробітна плата виконавців	8770,60
Додаткова заробітна плата виконавців	1754,13
Єдиний соціальний внесок	2315,44
Накладні витрати	3508,24
Витрати на амортизацію обладнання, програмного забезпечення	866,2612
Витрат всього	17 214,70

Таким чином собівартість проекту складе 17 214,70 грн. і основний елемент витрат – це заробітна плата виконавців.

Експлуатаційними витратами є такі витрати, які забезпечують нормальне функціонування певного технічного рішення в період його експлуатації. Величина експлуатаційних витрат в розрахунку за 1 рік може бути прогнозована за формулою:

$$E = k \cdot Ц \cdot \beta = k \cdot A \cdot СП \cdot \beta \quad (4.5)$$

де Ц — ціна реалізації нової розробки, якщо вона була визначена раніше, грн/од.,

k — коефіцієнт, який ураховує витрати на амортизацію, електроенергію, обслуговування, ремонти тощо. Для обчислювальної техніки $k = 0,5 \dots 0,7$.

A — коефіцієнт, який ураховує прогнозований прибуток та податки, які повинен сплачувати виробник $A \approx 1,7 \dots 2,3$;

СП — собівартість нової розробки;

B — доля часу, який витрачає працівник на обслуговування технічної розробки в загальному часі своєї роботи. ($B \approx 0,07$)

$$E = 0,6 \times 2,0 \times 17214,70 \times 0,07 = 1\,446,03$$

Після реалізації річна економія часу становитиме 260 год.

Таким чином, річний економічний ефект становитиме:

$$260 \times 58,7 - 1\,446,03 = 13\,815,97 \text{ грн.}$$

Визначимо термін окупності статистично-аналітичної системи за формулою:

$$T_{\text{ок. СП}} / E_{\text{ф}} \quad (4.6)$$

де СП - собівартість створення програмного продукту, грн.

$E_{\text{ф}}$ – прибуток від використання комп'ютерної мережі або програми (за мінусом експлуатаційних витрат), грн.

$$T_{\text{ок.}} = 17\,214,70 / 13\,815,97 = 1,24$$

Коефіцієнт економічної ефективності розраховується за формулою:

$$K_{\text{еф.}} = 1 / T_{\text{ок.}} \quad (4.7)$$

$$K_{\text{еф.}} = 1 / 1,24 = 0,8$$

Таблиця 1.9 – Основні економічні показники

Показники	Значення
Собівартість створення статистично-економічної системи, грн.	17 214,70
Експлуатаційні витрати, грн.	1 446,03
Додатковий прибуток, грн.	13 815,97
Строк окупності, років.	1,24
Коефіцієнт економічності ефективності.	0,8

На створення даного проекту необхідно витратити 17 214,70 грн., що забезпечить економію витрат в розмірі 13 815,97 грн., внаслідок чого витрати окупляться за 1,24 року. Отже, розробка додатку для VESCO є економічно доцільним та обґрунтованим проектом.

1.5.1 Економічна доцільність проекту

Визначення доцільності розробки проекту з економічної точки зору є досить важливою стадією, оскільки розробка програмного забезпечення має бути корисним для VESCO, збільшити зручність при використанні додатку, шляхом економії витрат.

Для розробки програмного додатку необхідно витратити 17 214,70 грн., що забезпечить економію витрат в розмірі 13 815,97 грн., внаслідок чого витрати окупляться за 1,24 року. Розробка додатку для фірми VESCO є економічно доцільним та обґрунтованим проєктом.

2 ТЕХНІЧНЕ ЗАВДАННЯ

2.1 Постановка задачі

Метою кваліфікаційної роботи є розробка програмного додатку для підтримки діяльності підприємства з виготовлення металоконструкцій. Для досягнення поставленої мети необхідно розв'язати такі задачі:

- провести аналіз обліку інформації про металоконструкції, з визначенням параметрів, що найбільше впливають на облік інформації про металоконструкції;
- на основі проведеного аналізу обліку інформації про металоконструкції, визначення методів для прогнозування стану обліку інформації про металоконструкції;
- розробити програмний застосунок, що здійснює прогнозування обліку інформації про металоконструкції;
- підтримка можливості збереження результату;
- провести тестування й оцінку продуктивності розробленого застосунку.

Система може застосовуватись як і окремо, так і вбудовуватися в іншу систему.

2.2 Вимоги до системи

У системі повинні бути прості умови використання, простий спосіб перегляду, щоб користувач міг легко зрозуміти функції програми, програма має легко запускатися і містити в собі всю необхідну інформацію, а саме:

Програма повинна бути орієнтована на платформу операційної системи сімейства Windows.

Для правильної роботи розробленого додаток до складу технічних засобів мають входити:

Мінімальні:

Процесор: 2 ГГц.

Оперативна пам'ять: 2 Гб.

Відеокарта: ATI Radeon 9400 128 Мбайт.

Жорсткий диск: 1 Гігабайт вільного простору.

DirectX: 9.0.

Роздільна здатність монітору: 1366 x 768 Пікселів.

Рекомендовані:

Процесор: Intel pentium 4 2,4 ГГц.

Оперативна пам'ять: 2048 Мбайт.

Відеокарта: ATI Radeon 9550 128 Мбайт.

Жорсткий диск: 4 Гбайт вільного простору.

DirectX: 9.

Роздільна здатність монітору: 1920 x 1080 Пікселів.

Також має бути встановлене таке програмне забезпечення:

- 1) операційна система Windows 8 або вище;
- 2) .NET Framework 4.6.1 або вище;

2.3 Склад і зміст робіт зі створення системи

Роботи зі створення системи виконуються в декілька етапів:

- Дослідження об'єкта автоматизації
- Технічне завдання
- Моделювання БД;
- Узгодження структури БД;
- Розгортання структури БД на відповідній СКБД;
- Створення програмних модулів для реалізації функцій системи;
- Тестування та усунення недоліків;
- Створення документації;
- Впровадження і навчання користувачів.

2.4 Порядок контролю і приймання системи

Перед впровадженням система повинна пройти приймальні випробування згідно з ДСТУ 3974-2000.

В ході приймання робіт будуть задіяні такі види випробувань як:

- Попередні;
- Дослідна експлуатація;
- Приймальні.

Випробування проводяться згідно програми яка складається Розробником та затверджується Замовником. В результаті приймається рішення щодо працездатності та подальшого приймання системи в дослідну експлуатацію.

На стадії «Дослідна експлуатація» здійснюються випробування заявленої функціональності згідно технічного завдання, занотовуються виявлені недоліки та встановлюються бажані терміни їх усунення.

На стадії «Приймальних випробувань» приймається рішення щодо готовності ІСПРКП до впровадження в промислову експлуатацію.

2.5 Вимоги до складу та змісту робіт з підготовки об'єкта автоматизації до введення системи в дію

До початку розробки робочої документації Замовником повинно бути здійснена підготовка приміщення для розміщення АТК ІСПРКП, придбання та встановлення необхідних технічних засобів, а також організація необхідної мережевої взаємодії.

Зі сторони Розробника повинно бути проведено навчання (дистанційно або стаціонарно) посадових осіб Замовника, що здійснюють технічну експлуатацію ІСПРКП.

Для функціонування ІСПРКП не передбачається створення додаткових підрозділів і служб.

2.6 Вимоги до документування

Склад і зміст документів, що розроблюються, визначаються у відповідності з вимогами ДСТУ 3008-2015 [1] та узгоджуються Розробником і Замовником в процесі розробки. Всі документи розробляються державною мовою.

Перелік документів, що будуть створені:

- Технічне завдання;
- Інструкція користувача.

Вся документація повинна бути підготовлена і передана як в друкованому, так і в електронному вигляді (у форматі Microsoft Word).

2.7 Вимоги до функцій

Істотні функції, що здійснює продукт наведено в таблиці 2.1.

Таблиця 2.1 Істотні функції ІС.

№	Функція
1	Авторизація користувача
2	Забезпечення доступу до даних та навігація по ІС
3.	Додавання, видалення, зміна даних
4.	Функція пошуку та фільтрування
5.	Формування замовлень та коригування даних щодо поточних проектів
6.	Формування експорту даних

2.8 Джерела розробки

Дане Технічне завдання розроблено на основі наступних документів та інформаційних матеріалів:

- ДСТУ 3008-2015. Документація. Звіти у сфері науки і техніки.

Структура

та правила оформлювання; [5]

- Постанова КМУ від 13.04.2011 № 461 «Питання прийняття в експлуатацію закінчених будівництвом об'єктів»; [6]

- ДСТУ Б А.2.4-4:2009 «Основні вимоги до проектної та робочої документації»; [7]

- ДБН А.2.2-3-2014 Склад проектної документації для будівництва; [8]

- ДБН А.3.1-5-2009 Організація будівельного виробництва. [9]

3 ОПИС КОМПЛЕКСУ ЗАДАЧ АВТОМАТИЗАЦІЇ

3.1 Вибір програмного та апаратного забезпечення

Програмне забезпечення (ПЗ) - сукупність програм, що дозволяє організувати вирішення завдань на комп'ютері. ПЗ і архітектура машини утворюють комплекс взаємопов'язаних і різноманітних функціональних засобів, що визначають здатність рішення того чи іншого класу задач. Найважливішими класами ПЗ є системне і спеціальне (прикладне), представлене пакетом прикладних програм.

Системне програмне забезпечення - це сукупність програм, що забезпечують: створення операційного середовища функціонування інших програм, надійну і ефективну роботу комп'ютера і комп'ютерних мереж, проведення діагностики та профілактики апаратури комп'ютера і мереж.

3.1.1 C#

C# (вимовляється Сі-шарп) — об'єктно-орієнтована мова програмування з безпечною системою типізації для платформи .NET. Розроблена Андерсом Гейлсбергом, Скотом Вілтанутом та Пітером Гольде під егідою Microsoft Research (належить Microsoft).

Синтаксис C# близький до C++ і Java. Мова має строгу статичну типізацію, підтримує поліморфізм, перевантаження операторів, вказівники на функції-члени класів, атрибути, події, властивості, винятки, коментарі у форматі XML. Переїнявши багато від своїх попередників — мов C++, Object Pascal, Модула і Smalltalk — C#, спираючись на практику їхнього використання, виключає деякі моделі, що зарекомендували себе як проблематичні при розробці програмних систем, наприклад, мова C#, на відміну від C++, не передбачає множинне успадкування класів.

3.1.2 WPF

Технологія WPF (Windows Presentation Foundation) є частина екосистеми платформи .NET і являє собою підсистему для побудови графічних інтерфейсів.

Якщо при створенні традиційних додатків на основі WinForms за відображення елементів управління і графіки відповідали такі частини ОС Windows, як User32 і GDI +, то додатки WPF засновані на DirectX . У цьому полягає ключова особливість рендеринга графіки в WPF: використовуючи WPF, значна частина роботи по відображенні графіки, як найпростіших кнопочок, так і складних 3D-моделей, розміщується на відеокарті графічного процесора, що також дозволяє скористатися апаратним прискоренням графіки.

Однією з важливих особливостей є використання мови декларативною розмітки інтерфейсу XAML, заснованого на XML: можна створювати насичений графічний інтерфейс, використовуючи або декларативне оголошення інтерфейсу, або код на керованих мовах C # і VB.NET, або поєднувати і те, і інше.

Переваги WPF:

- Використання традиційних мов .NET-платформи - C # і VB.NET для створення логіки додатка.
- Можливість декларативного визначення графічного інтерфейсу за допомогою спеціальної мови розмітки XAML, заснованому на xml і представляє альтернативу програмному створення графіки та елементів управління, а також можливість комбінувати XAML і C # / VB.NET.
- Незалежність від дозволу екрану : оскільки в WPF всі елементи вимірюються в незалежних від пристрою одиницях, додатки на WPF легко масштабуються під різні екрани з різним дозволом.
- Нові можливості, яких складно було досягти в WinForms, наприклад, створення тривимірних моделей, прив'язка даних, використання таких елементів, як стилі, шаблони, теми і ін.
- Гарне взаємодія з WinForms , завдяки чому, наприклад, в додатках WPF можна використовувати традиційні елементи управління з WinForms.

- Багаті можливості по створенню різних додатків: це і мультимедіа, і двомірна і тривимірна графіка, і багатий набір вбудованих елементів управління, а також можливість самим створювати нові елементи, створення анімацій, прив'язка даних, стилі, шаблони, теми і багато іншого.
- Апаратне прискорення графіки - незалежно від того, чи працюєте ви з 2D або 3D, графікою або текстом, всі компоненти програми транслюються в об'єкти, зрозумілі Direct3D, і потім візуалізуються за допомогою процесора на відеокарті, що підвищує продуктивність, робить графіку більш плавною.
- Створення додатків під множину ОС сімейства Windows - від Windows XP до Windows 10.

3.1.3 .NET Framework

.NET Framework (читається дот-нет) — програмна технологія, запропонована фірмою Microsoft як платформа для створення як звичайних програм, так і веб-застосунків. Багато в чому є продовженням ідей та принципів, покладених в технологію Java. Однією з ідей .NET є сумісність служб, написаних різними мовами. Хоча ця можливість рекламується Microsoft як перевага .NET, платформа Java має таку саму можливість.

Кожна бібліотека (збірка) в .NET має дані про свою версію, що дозволяє усунути можливі конфлікти між різними версіями збірок.

.NET — крос-платформова технологія, в цей час існує реалізація для платформи Microsoft Windows, FreeBSD (від Microsoft) і варіант технології для ОС Linux в проєкті Mono (в рамках угоди між Microsoft з Novell), DotGNU.

3.1.4 .NET Core

.NET Core це безкоштовний крос-платформний фреймворк з керованим кодом підтримуваний на Windows, Linux і Mac OSX. На відміну від .NET Framework вихідний код .NET Core є повністю відкритим і доступним за наступним посиланням <https://github.com/dotnet/core>. Він містить CoreCLR - повністю крос-платформну реалізацію CLR, віртуальну машину, яка керує

виконанням програм в .NET середовищі. CoreCLR поставляється з оптимізованим «just-in-time» компілятором RyuJIT. .NET Core також містить у собі CoreFX, яка становить собою часткове відгалуження FCL (стандартна бібліотека класів .NET фреймворку) Реалізації усіх класів також відкриті та доступні за посиланням <https://github.com/dotnet/corefx>. У той час як .NET Core розділяє підмножину API .NET Framework, він містить також власний API, який не є частиною .NET Framework. Крім того .NET Core містить CoreRT, оптимізований під інтеграцію в AOT(компіляція перед виконанням) бінарні файли. Варіант бібліотеки .NET Core використовується для UWP (універсальна платформа Windows). UWP платформа, створена Microsoft і вперше представлена в Windows 10. Метою даної платформи є допомога у створенні універсальних додатків Windows, що запускаються як на Windows 10, так і на Windows 10 Mobile без зміни в коді. Інтерфейс командного рядка .NET Core пропонує точку входу для операційних систем і надає послуги для розробників, такі як компіляція і пакети управління.

3.2 Проектування власного програмного додатку

При завантаженні одних пакетів автоматично можуть завантажитися інші, необхідні для його роботи пакети.

Для створення бази даних було використано додаток Microsoft Access зображену на рисунку 3.1.

ID	Назва	Категорія	Дата	Телефон	Тег	Логін	Пароль
120	Відділ управління		12.11.2013	+380981180178			admin
88	Відділ управління		12.05.2013	+380981180178			admin
88	Відділ управління		12.05.2013	+380981180178			admin
11	Відділ управління		12.05.2013	+380981180178			admin
11	Відділ управління		12.05.2013	+380981180178			admin
88	Відділ управління		12.05.2013	+380981180178			admin
8	Відділ управління		12.05.2013	+380981180178			admin
11	Відділ управління		12.05.2013	+380981180178			admin
1	Відділ управління		12.11.2013	+380981180178			admin
6	Відділ управління		12.12.2013	+380981180178			admin

Рисунок 3.1 – Access

Були створені таблиці для роботи з даними та згенеровані зв'язки між ними, зображені на рисунках 3.2 та 3.3.

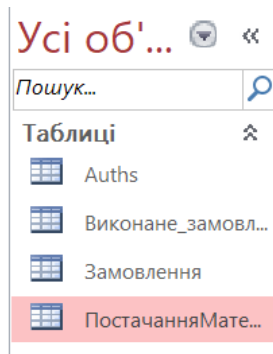


Рисунок 3.2 – Створення таблиць

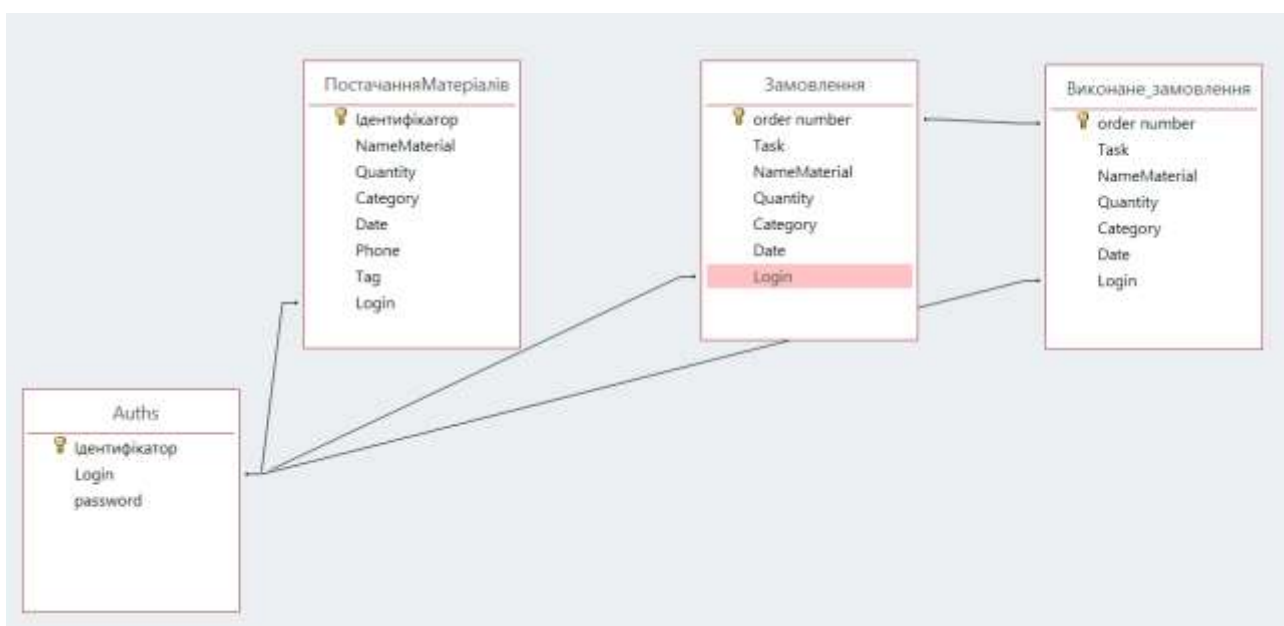


Рисунок 3.3 – Створення зв'язків

Скинь мені, будь-ласка, файл БД в Access

Щоб працювати з Microsoft Access Database у Visual Studio, необхідно спочатку підключити базу даних до проекту. Для цього виконайте такі дії:

У відкритому проекті на верхній панелі інструментів обрати tools> connect to Database...

Далі у вікні яке відкриється обрати свій файл бази даних, цей процес зображено на рисунку 3.4.

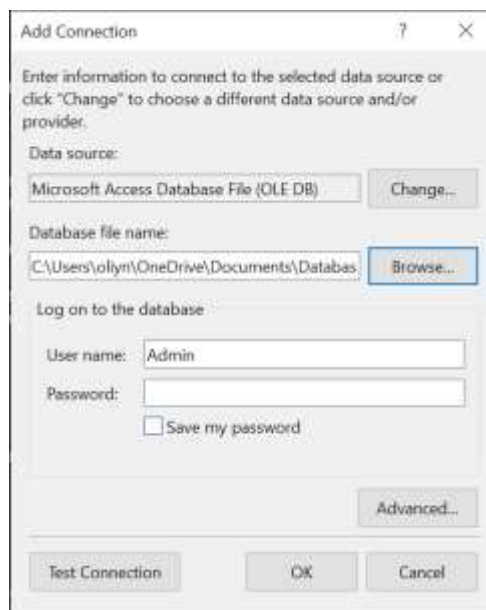


Рисунок 3.4 – Підключення DB у Visual Studio

Тепер у вікні Server Explorer буде відображено підключену базу даних та її таблиці і елементи.

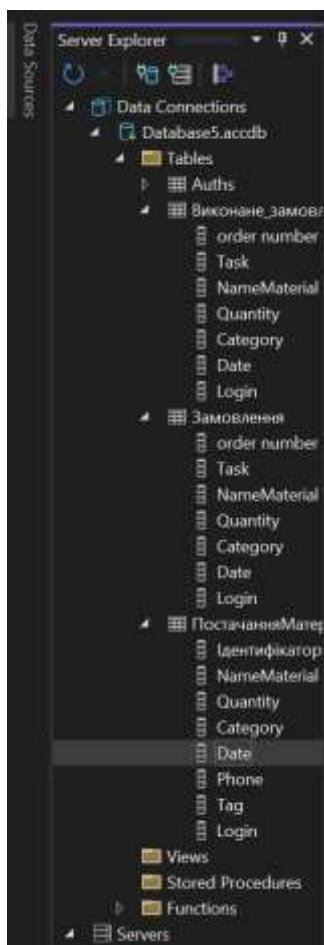


Рисунок 3.5 – DB у Server Explorer

Щоб працювати з базою даних, потрібно підключити бібліотеку System.Data.OleDb;

Для роботи з базою даних потрібно підключитися до неї та створити функції для роботи з нею:

```

try
    {
        string connectionString = "Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=Database5.accdb;";
        using (OleDbConnection connection = new OleDbConnection(connectionString))
        {
            connection.Open();

            string selectAllQuery = "SELECT * FROM ПостачанняМатеріалів";
            using (OleDbCommand selectAllCommand = new
OleDbCommand(selectAllQuery, connection))
            using (OleDbDataReader reader = selectAllCommand.ExecuteReader())
            {
                List<Materials> materialsList = new List<Materials>();

                while (reader.Read())
                {
                    Materials material = new Materials
                    {
                        id = Convert.ToInt32(reader["Ідентифікатор"]),
                        Name = Convert.ToString(reader["NameMaterial"]),
                        Quantity = Convert.ToString(reader["Quantity"]),
                        Category = Convert.ToString(reader["Category"]),
                        Date = Convert.ToString(reader["Date"]),
                        Phone = Convert.ToString(reader["Phone"]),
                        Tag = Convert.ToString(reader["Tag"]),
                    };

                    materialsList.Add(material);
                }

                listofusers.ItemsSource = materialsList;
            }
            string selectOrder = "SELECT * FROM Замовлення";
            using (OleDbCommand selectAllCommand = new
OleDbCommand(selectAllQuery, connection))
            using (OleDbDataReader reader = selectAllCommand.ExecuteReader())
            {
                List<Materials> materialsList = new List<Materials>();

                while (reader.Read())
                {
                    Materials material = new Materials
                    {

```

```

        // Припускається, що ваші стовпці мають відповідні типи даних,
імена стовпців слід замінити на ваші
        id = Convert.ToInt32(reader["order number"]),
        Name = Convert.ToString(reader["NameMaterial"]),
        Task = Convert.ToString(reader["Task"]),
        Quantity = Convert.ToString(reader["Quantity"]),
        Category = Convert.ToString(reader["Category"]),
        Date = Convert.ToString(reader["Date"]),

        Tag = Convert.ToString(reader["Tag"]),
        // і так далі, відповідно до вашої моделі Materials
    };

    materialsList.Add(material);
}

Listofusers2.ItemsSource = materialsList;
}
string selectOrderComplite = "SELECT * FROM Виконане_замовлення";
using (OleDbCommand selectAllCommand = new
OleDbCommand(selectAllQuery, connection))
using (OleDbDataReader reader = selectAllCommand.ExecuteReader())
{
    List<Materials> materialsList = new List<Materials>();

    while (reader.Read())
    {
        Materials material = new Materials
        {
            // Припускається, що ваші стовпці мають відповідні типи даних,
імена стовпців слід замінити на ваші
            id = Convert.ToInt32(reader["order number"]),
            Name = Convert.ToString(reader["NameMaterial"]),
            Task = Convert.ToString(reader["Task"]),
            Quantity = Convert.ToString(reader["Quantity"]),
            Category = Convert.ToString(reader["Category"]),
            Date = Convert.ToString(reader["Date"]),
            Tag = Convert.ToString(reader["Tag"]),
            // і так далі, відповідно до вашої моделі Materials
        };

        materialsList.Add(material);
    }

    Listofusers3.ItemsSource = materialsList;
}

string selectDistinctCategoriesQuery = "SELECT DISTINCT Category FROM
Materials";
using (OleDbCommand selectDistinctCategoriesCommand = new
OleDbCommand(selectDistinctCategoriesQuery, connection))
using (OleDbDataReader categoryReader =
selectDistinctCategoriesCommand.ExecuteReader())

```

```

{
    List<string> distinctCategories = new List<string>();

    while (categoryReader.Read())
    {
        distinctCategories.Add(Convert.ToString(categoryReader["Category"]));
    }
    listofcategories.ItemsSource = distinctCategories;
}
}
}
catch (Exception ex)
{
    MessageBox.Show($"Exception caught: {ex.Message}");
}
}

```

3.2.1 Вікно авторизації користувача

Для реалізації авторизації користувача заздалегідь було створено таблицю Користувач в БД яка містить наступні поля: id, логін, пароль. В ній був створений перший запис, який і буде передбачати вхід в систему для координатора.

На головну форму меню було додано два поля textbox (для введення логіну та паролю), та кнопку button (вхід в систему).

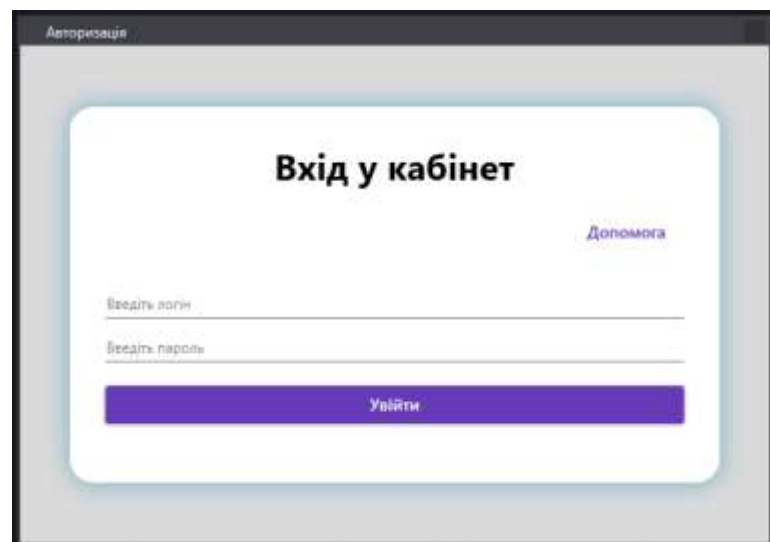


Рисунок 3.6 –Вікно авторизації

При введенні даних для входу необхідно вказати Логін та Пароль. Після чого натиснувши на кнопку «Вхід» спрацьовує наступний код:

```

private void Button_Click(object sender, RoutedEventArgs e)
{
    try
    {
        string login = LoginBox.Text.Trim();
        string pass = PasswordBox.Password.Trim();

        if (login.Length < 4)
        {
            LoginBox.ToolTip = "Короткий логін";
            LoginBox.Background = Brushes.IndianRed;
        }
        else if (pass.Length < 5)
        {
            PasswordBox.ToolTip = "Короткий пароль";
            PasswordBox.Background = Brushes.IndianRed;
        }
        else
        {
            LoginBox.ToolTip = "";
            LoginBox.Background = Brushes.Transparent;
            PasswordBox.ToolTip = "";
            PasswordBox.Background = Brushes.Transparent;

            Auth auth = null;
            using (ApplicationContext db = new ApplicationContext())
            {
                auth = db.Auths.Where(a => a.Login == LoginBox.Text && a.Password ==
                PasswordBox.Password).FirstOrDefault();

            }
            if (auth == null)
            {
                MessageBox.Show("error");
            }
            else
            {
                MainWindow MainWindow = new MainWindow();
                MainWindow.Show();
                this.Hide();
            }
        }
    }
    catch (ArgumentNullException ex)
    {
        MessageBox.Show("{0} First exception caught." + ex);
    }
    catch (Exception ex)
    {
        MessageBox.Show("{0} Second exception caught." + ex);
    }
}

```

3.2.2 Головне вікно програми

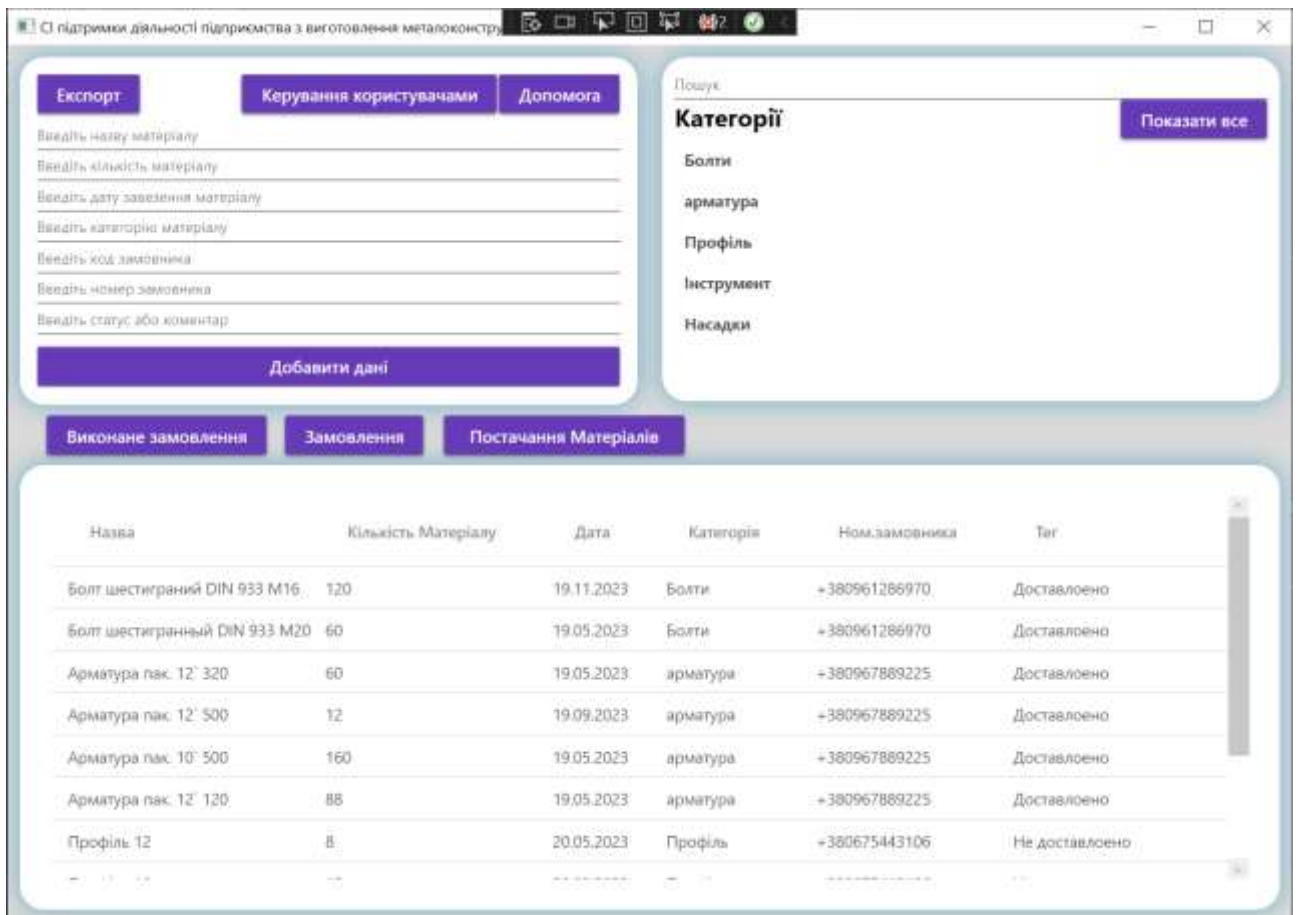


Рисунок 3.7 –Головне вікно програми

Код до кнопки «Додати дані» :

```
private void Button_Click_AddMaterials(object sender, RoutedEventArgs e)
{
    try
    {
        string bk1 = Bk1.Text.Trim();
        string bk2 = Bk2.Text.Trim();
        string bk3 = Bk3.Text.Trim();
        string bk4 = Bk4.Text.Trim();
        string bk5 = Bk5.Text.Trim();
        string bk6 = Bk6.Text.Trim();
        string bk7 = Bk7.Text.Trim();
        if (bk1.Length != 0 && bk2.Length != 0 && bk3.Length != 0)
        {
            if (bk6.IndexOf("+") != 0 && bk6 == "")
            {
                if (bk1.Length < 92 && bk2.Length < 92 && bk3.Length < 92 && bk4.Length < 92
&& bk5.Length < 92 && bk6.Length < 92 && bk7.Length < 92)
                {
```

```

        Materials Materialsadd = new Materials(Bk1.Text, Bk2.Text, Bk3.Text, Bk4.Text,
Bk5.Text, Bk6.Text, Bk7.Text);
        db.Materials.Add(Materialsadd);
        db.SaveChanges();
        UpdateList();
    }
    else
    {
        MessageBox.Show("Введений текст перевищує 92 символа!");
    }
}
else
{
    MessageBox.Show("Введено не проавильно номер телефону!");
}
}
else
{
    MessageBox.Show("Введений текст менше 1 символа!");
}
}
catch (ArgumentNullException ex)
{
    MessageBox.Show("{0} First exception caught." + ex);
}
catch (Exception ex)
{
    MessageBox.Show("{0} Second exception caught." + ex);
}
}

```

Блок-схема до кнопки «Добавити даних», зображена на рисунку 3.8.

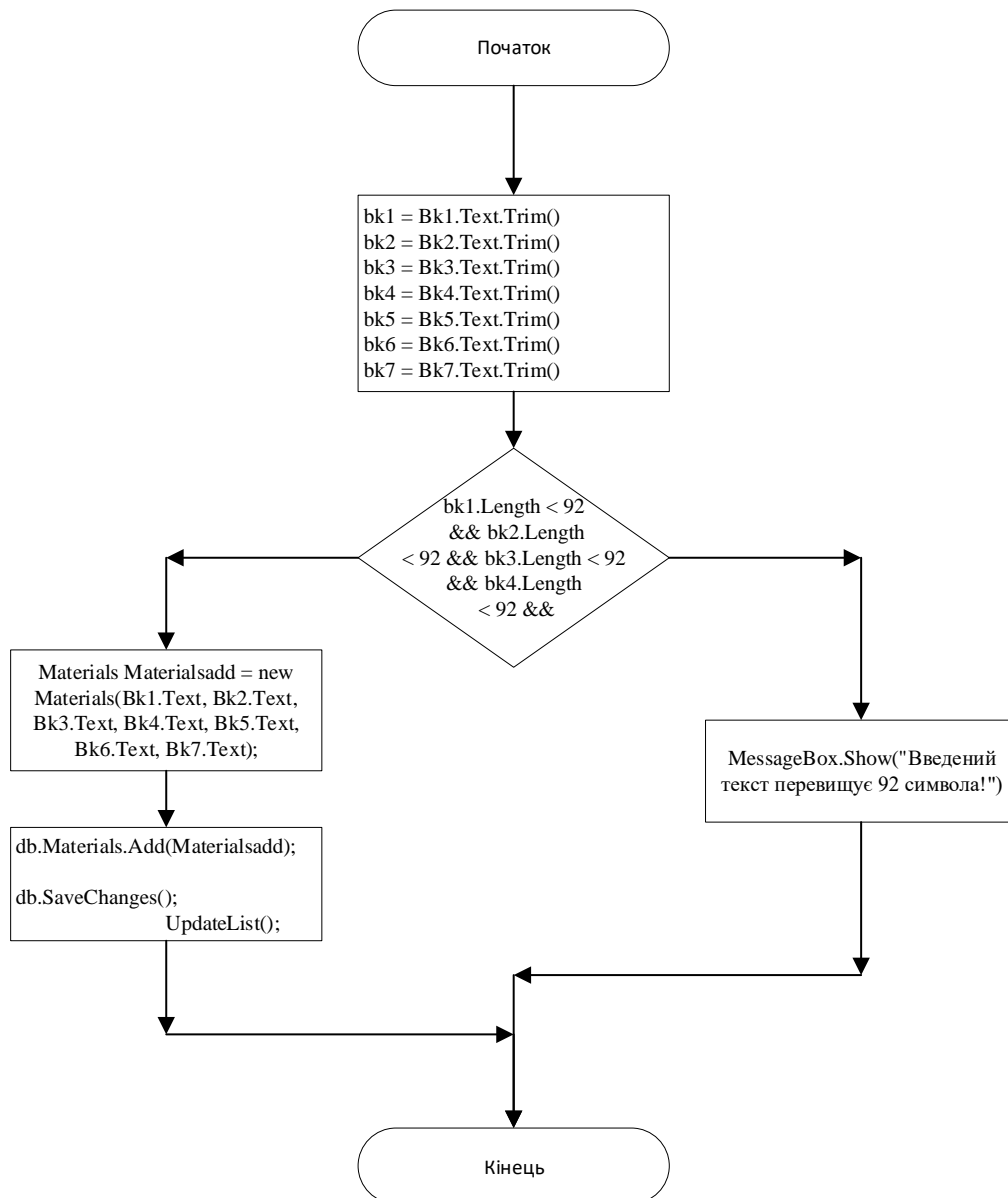


Рисунок 3.8 – Блок-схема до кнопки «Добавити даних»

Код до кнопки змінити дані:

```

private void MenuItem_Click_Edit(object sender, RoutedEventArgs e)
{
    try
    {
        if (listofusers.SelectedItem == null) return;
        Materials Materials = listofusers.SelectedItem as Materials;

        EditItem_Window Bkwin = new EditItem_Window(new Materials
        {
            id = Materials.id,
            Name = Materials.Name,
            Quantity = Materials.Quantity,
            Date = Materials.Date,
            Category = Materials.Category,
            Kod = Materials.Kod,
            Phone = Materials.Phone,

```

```
        Tag = Materials.Tag
    });

    if (Bkwin.ShowDialog() == true)
    {
        Materials = db.Materials.Find(EditItem_Window.Materials.id);
        if (Materials != null)
        {
            Materials.Name = EditItem_Window.Materials.Name;
            Materials.Quantity = EditItem_Window.Materials.Quantity;
            Materials.Date = EditItem_Window.Materials.Date;
            Materials.Category = EditItem_Window.Materials.Category;
            Materials.Kod = EditItem_Window.Materials.Kod;
            Materials.Phone = EditItem_Window.Materials.Phone;
            Materials.Tag = EditItem_Window.Materials.Tag;
            db.SaveChanges();
        }
    }
    UpdateList();
}
catch (ArgumentNullException ex)
{
    MessageBox.Show("{0} First exception caught." + ex);
}
catch (Exception ex)
{
    MessageBox.Show("{0} Second exception caught." + ex);
}
}
```

Блок-схема до кнопки «Змінити», зображена на рисунку 3.9.

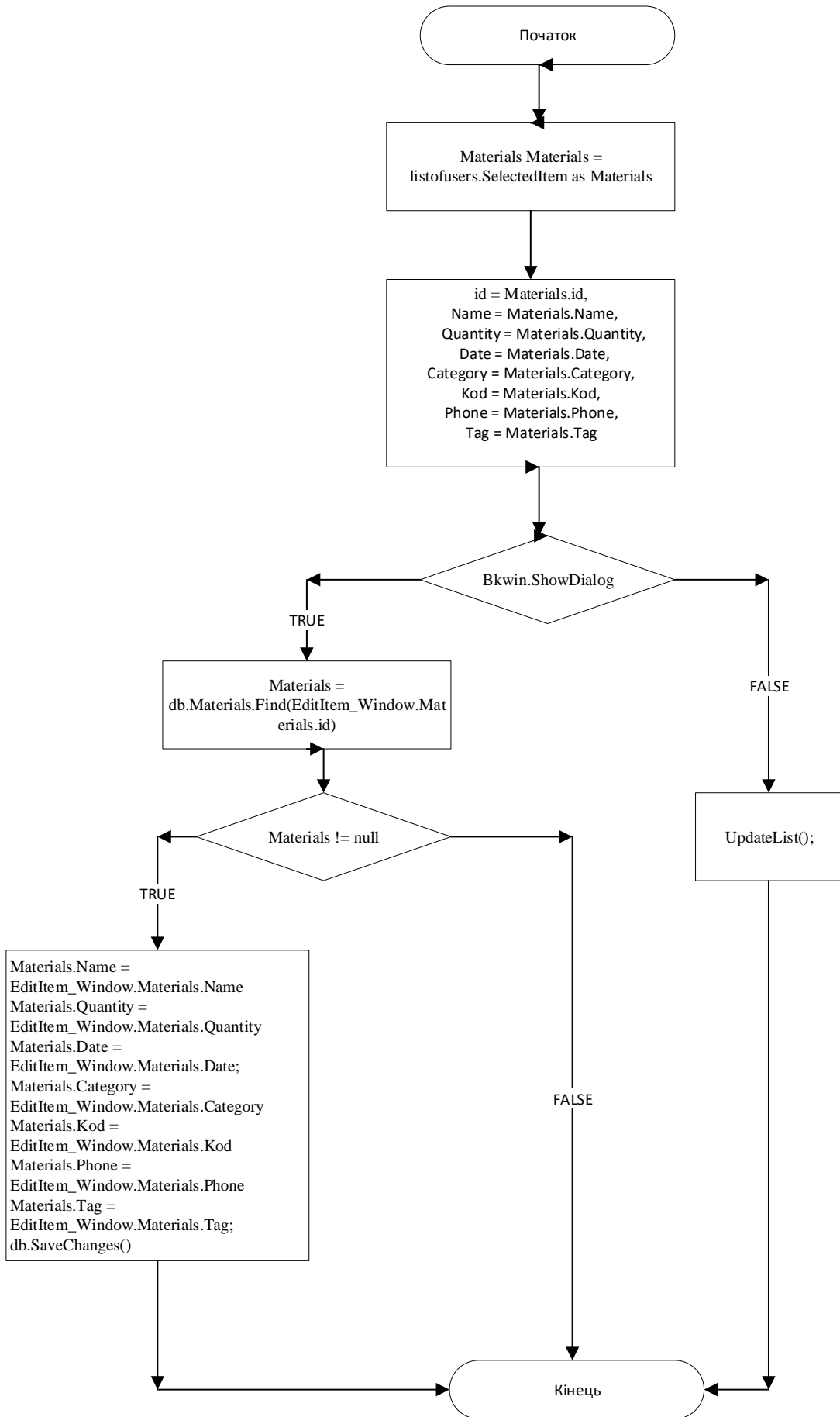


Рисунок 3.9 – Блок-схема до кнопки «Змінити»

Код кнопки «Оновити список» контекстного меню:

```
private void MenuItem_Click_Update(object sender, RoutedEventArgs e)
{
    UpdateList();
}
```

Код кнопки «Видалити» контекстного меню:

```
private void MenuItem_Click_Delete(object sender, RoutedEventArgs e)
{
    try
    {
        MessageBoxResult messageBoxResult = System.Windows.MessageBox.Show(«Are you
sure?», «Delete Confirmation», System.Windows.MessageBoxButton.YesNo);
        if (messageBoxResult == MessageBoxResult.Yes)
        {
            if (listofusers.SelectedItem != null)
            {
                Materials Materialsrem = listofusers.SelectedItem as Materials;
                db.Materials.Remove(Materialsrem);
                db.SaveChanges();
                UpdateList();
            }
        }
    }
    catch (ArgumentNullException ex)
    {
        MessageBox.Show(«{0} First exception caught.» + ex);
    }
    catch (Exception ex)
    {
        MessageBox.Show(«{0} Second exception caught.» + ex);
    }
}
```

Поле пошуку:

```
private void SeachBox_TextChanged(object sender, TextChangedEventArgs e)
{
    try
```

```

{
    if (SeachBox.Text != «»)
    {
        List<Materials> Materials = db.Materials.Where(a => a.Name == SeachBox.Text ||
a.Author == SeachBox.Text || a.Date == SeachBox.Text).ToList();
        listofusers.ItemsSource = Materials;
    }
    else
    {
        UpdateList();
    }
}
catch (ArgumentNullException ex)
{
    MessageBox.Show(«{0} First exception caught.» + ex);
}
catch (Exception ex)
{
    MessageBox.Show(«{0} Second exception caught.» + ex);
}
}

```

Функція вибору категорій:

```

private void Listofcategories_MouseDoubleClick(object sender, MouseButtonEventArgs e)
{
    try
    {
        if (listofcategories.SelectedItem.ToString() != «»)
        {
            string ss = listofcategories.SelectedItem.ToString();
            List<Materials> Materialsf = db.Materials.Where(a => a.Category == ss).ToList();
            listofusers.ItemsSource = Materialsf;
        }
    }
    else
    {
        UpdateList();
    }
}

```

```

    }
}
catch (ArgumentNullException ex)
{
    MessageBox.Show(«{0} First exception caught.» + ex);
}
}

```

Код до кнопки экспорту даних:

```

private void Button_Click_Export(object sender, RoutedEventArgs e)
{
    db = new ApplicationContext();
    List<Materials> Materials = db.Materials.ToList();
    SaveFileDialog saveFileDialog = new SaveFileDialog();
    saveFileDialog.Filter = "PDF Files|*.pdf";
    saveFileDialog.Title = "Save PDF File";
    saveFileDialog.FileName = "output.pdf";
    if (saveFileDialog.ShowDialog() == true)
    {
        using (FileStream fs = new FileStream(saveFileDialog.FileName, FileMode.Create))
        {
            Document document = new Document();
            PdfWriter writer = PdfWriter.GetInstance(document, fs);
            document.Open();
            //Paragraph header = new Paragraph("Export data list");
            //document.Add(header);
            PdfPTable table = new PdfPTable(7);
            PdfPCell cell;
            cell = new PdfPCell(new Phrase("Name"));
            table.AddCell(cell);
            cell = new PdfPCell(new Phrase("Author"));
            table.AddCell(cell);
            cell = new PdfPCell(new Phrase("Date"));
            table.AddCell(cell);
            cell = new PdfPCell(new Phrase("Category"));
            table.AddCell(cell);

```

```

cell = new PdfPCell(new Phrase("Kod"));
table.AddCell(cell);
cell = new PdfPCell(new Phrase("Phone"));
table.AddCell(cell);
cell = new PdfPCell(new Phrase("Status"));
table.AddCell(cell);
for (int i = 0; i < Materials.Count; i++)
{
    Materials currentMaterials = Materials[i];

    cell = new PdfPCell(new Phrase(currentMaterials.Name));
table.AddCell(cell);
    cell = new PdfPCell(new Phrase(currentMaterials.Author));
table.AddCell(cell);
    cell = new PdfPCell(new Phrase(currentMaterials.Date));
table.AddCell(cell);
    cell = new PdfPCell(new Phrase(currentMaterials.Category));
table.AddCell(cell);
    cell = new PdfPCell(new Phrase(currentMaterials.Kod));
table.AddCell(cell);
    cell = new PdfPCell(new Phrase(currentMaterials.Phone));
table.AddCell(cell);
    cell = new PdfPCell(new Phrase(currentMaterials.Status));
table.AddCell(cell);

    //Paragraph paragraph = new Paragraph($"Name: {currentMaterials.Name}, Author:
{currentMaterials.Author},          Date:          {currentMaterials.Date},          Category:
{currentMaterials.Category},Kod: {currentMaterials.Kod},Phone: {currentMaterials.Phone},Status:
{currentMaterials.Status}");

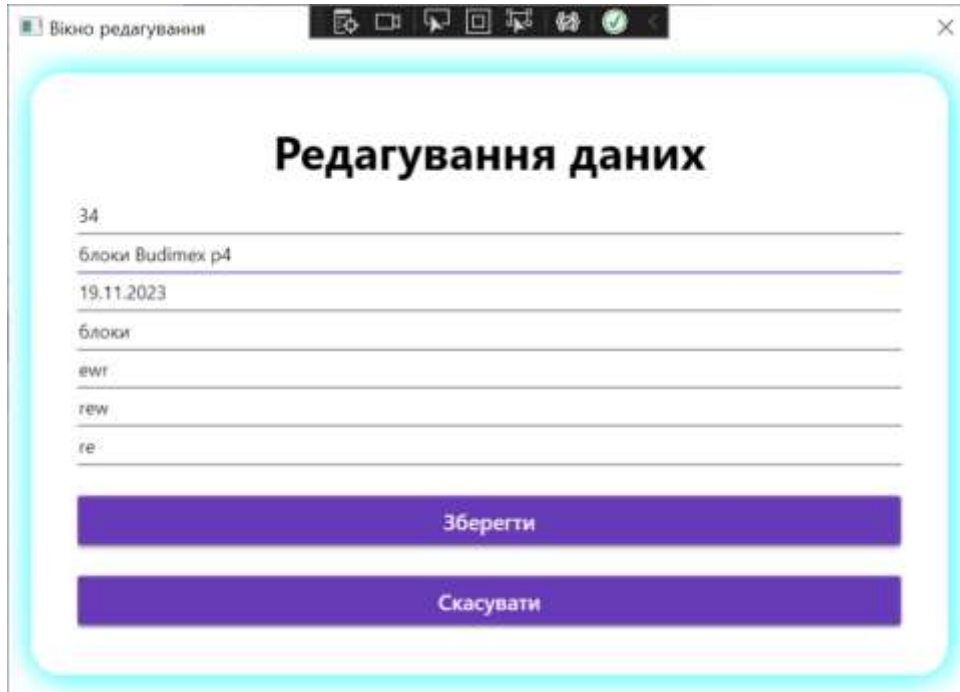
    //document.Add(paragraph);
}
document.Add(table);
document.Close();
}
}

```

}

3.2.3 Вікно редагування даних

Вікно редагування даних зображене на рисунку 3.10.

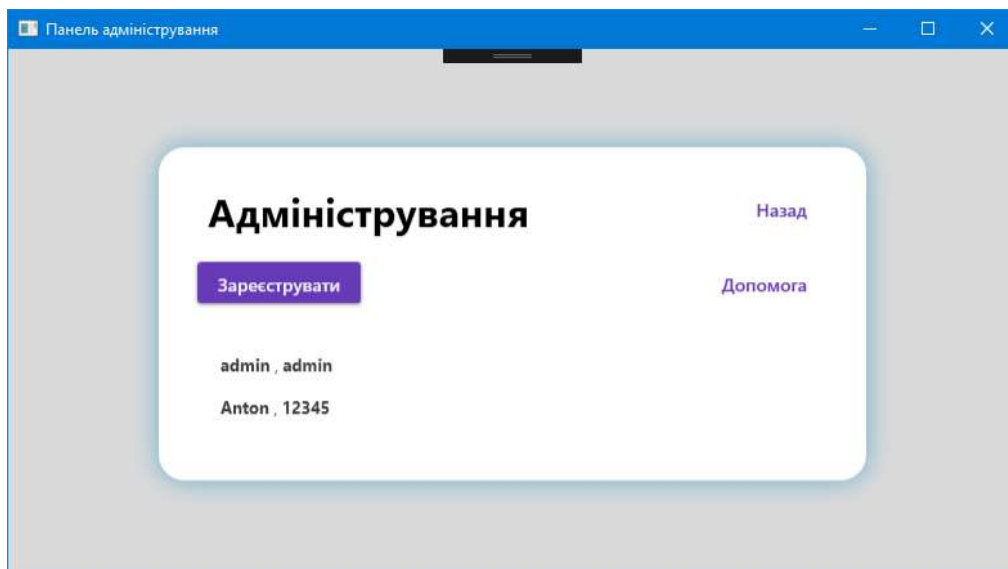


The screenshot shows a window titled "Вікно редагування" (Editing Window). The main content area is titled "Редагування даних" (Data Editing). It contains several input fields with the following text: "34", "блоки Viditex р4", "19.11.2023", "блоки", "ewr", "gew", and "ge". At the bottom of the form, there are two large purple buttons: "Зберегти" (Save) and "Скасувати" (Cancel).

Рисунок 3.10 – Вікно редагування даних

3.2.4 Вікно керування адміністраторами

Вікно реєстрації нових користувачів зображене на рисунку 3.11



The screenshot shows a window titled "Панель адміністрування" (Admin Panel). The main content area is titled "Адміністрування" (Administration). It features a purple button labeled "Зареєструвати" (Register). Below this, there are two lines of text: "admin , admin" and "Anton , 12345". In the top right corner, there are two links: "Назад" (Back) and "Допомога" (Help).

Рисунок 3.11 – Вікно реєстрації нових користувачів

Код до кнопки **реєстрації нового користувача**:

```
private void Button_Click(object sender, RoutedEventArgs e)
{
    try
    {
        string login = LoginBox.Text.Trim();
        string pass = PasswordBox.Password.Trim();
        string pass_2 = PasswordBox_2.Password.Trim();
        if (login.Length < 4)
        {
            LoginBox.ToolTip = "Короткий логін";
            LoginBox.Background = Brushes.IndianRed;
        }
        else
        {
            LoginBox.ToolTip = "";
            LoginBox.Background = Brushes.Transparent;
        }
        if (pass.Length < 5)
        {
            PasswordBox.ToolTip = "Короткий пароль";
            PasswordBox.Background = Brushes.IndianRed;
        }
        else
        {
            PasswordBox.ToolTip = "";
            PasswordBox.Background = Brushes.Transparent;
        }
        if (pass != pass_2)
        {
            PasswordBox_2.ToolTip = "Повторний пароль не вірний!";
            PasswordBox_2.Background = Brushes.IndianRed;
        }
        else
    }
}
```

```
{  
    PasswordBox_2.ToolTip = "";  
    PasswordBox_2.Background = Brushes.Transparent;  
}  
Auth boo = new Auth(login, pass);  
db.Auths.Add(boo);  
db.SaveChanges();  
}  
catch (ArgumentNullException ex)  
{  
    MessageBox.Show("{0} First exception caught." + ex);  
}  
catch (Exception ex)  
{  
    MessageBox.Show("{0} Second exception caught." + ex);  
}  
}
```

Вікно додавання користувачів зображене на рисунку 3.12.

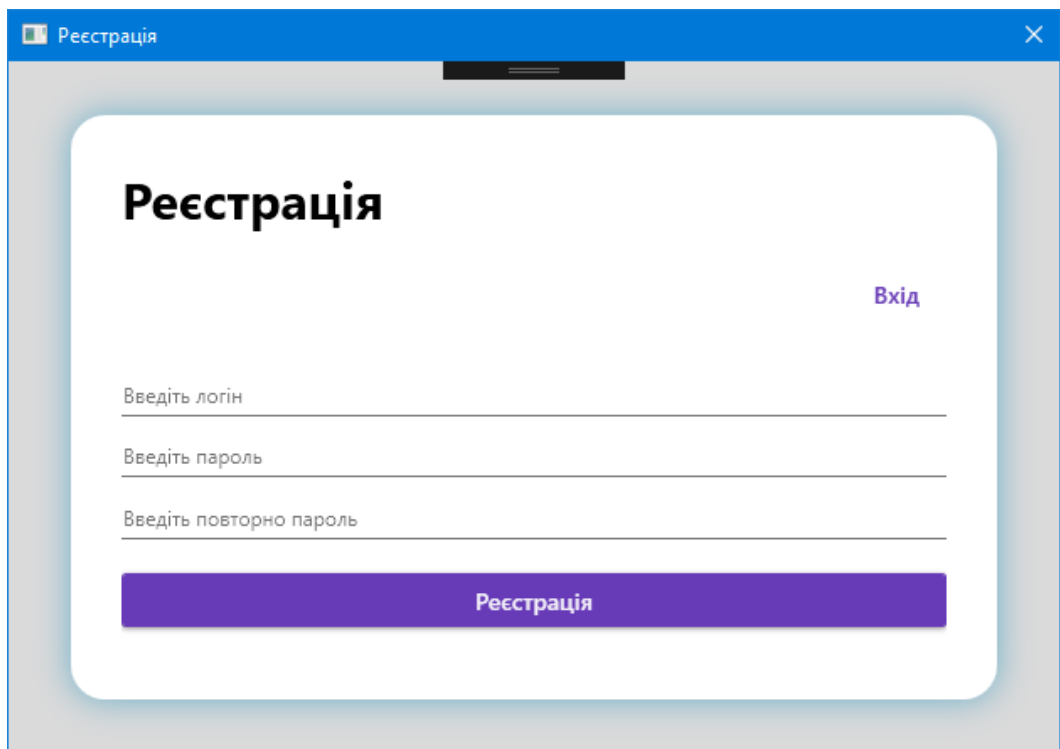


Рисунок 3.12 – Вікно додавання користувачів

На рисунку 3.13 відображено вікно з інструкцією користувача та зворотнім зв'язком, яке представлено в графічній частині.

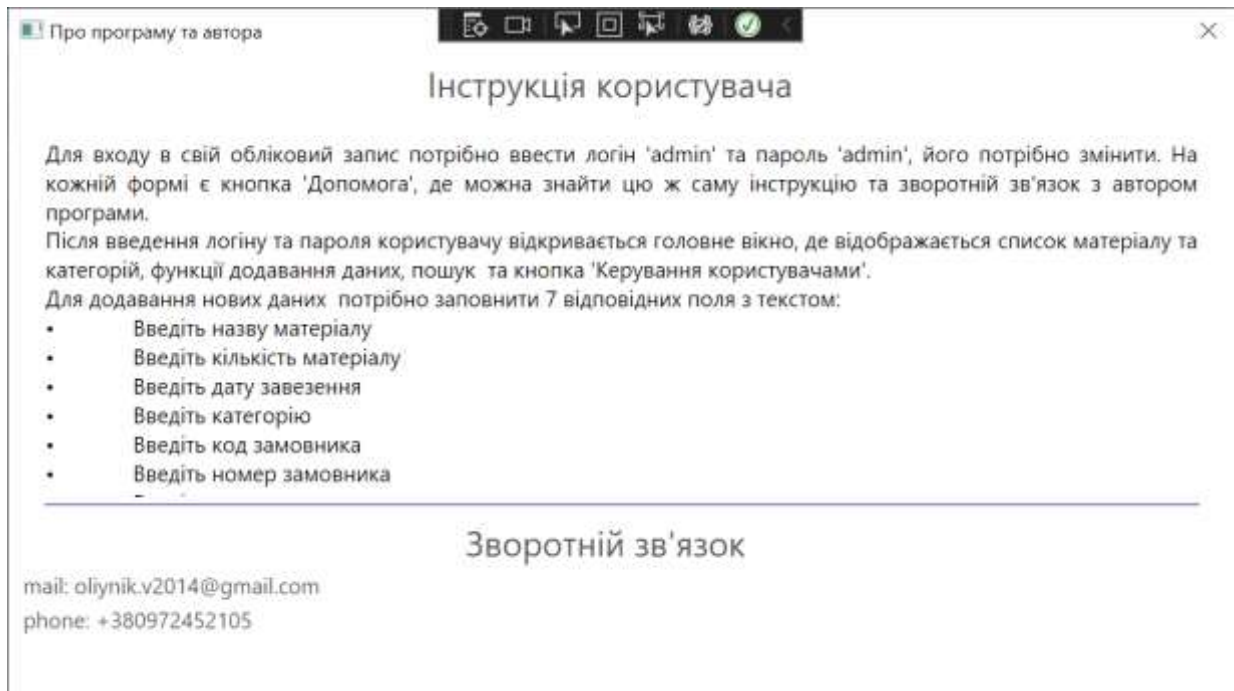


Рисунок 3.13 – Вікно інструкції користувача та зворотного зв'язку

Після створення програмного додатку, для нього потрібно створити установочний файл, з можливістю його подальшої установки на різних комп'ютерах. Для створення такого файлу можна використати один із безплатних засобів Inno Setup.

Для створення файлу потрібно запустити помічника і виконувати всі вказівки, в результаті згенерується скрипт, зображений на рисунку 3.12, який потрібно запустити, і після його завершення в папці з програмою згенерується папка output з файлом установки.

```

Inno Setup Compiler 6.2.0
File Edit View Build Run Tools Help

Script generated by the Inno Setup Script Wizard.
; SEE THE DOCUMENTATION FOR DETAILS ON CREATING INNO SETUP SCRIPT FILES!

#define MyAppName "BibliotekaKobocounting"
#define MyAppVersion "1.0"
#define MyAppPublisher "Нав'ямені Дмитроювічій Колякх мнрчююі проєктуювнн"
#define MyAppURL "https://kp-cfi.km.ua/en/"
#define MyAppExeName "BiblioChitKWF.exe"

[Setup]
; WUID: The value of AppID uniquely identifies this application. Do not use the same AppID value on installers for other applications.
; (To generate a new WUID, click Tools | Generate WUID inside the IDE.)
AppID={1DC7B270-22F2-4E56-A68B-91E065A686A3}
AppName={#MyAppName}
AppVersion={#MyAppVersion}
AppPublisher={#MyAppName} {#MyAppVersion}
AppPublisherURL={#MyAppPublisher}
AppSupportURL={#MyAppURL}
AppUpdatesURL={#MyAppURL}
SetupIconName={appdir}\{#MyAppName}
DisableProgramGroupPage=yes
InstallBeforeFile=C:\Program Files (x86)\Inno Setup 6\license.txt
; Uncomment the following line to run in non-administrative install mode (install for current user only.)
;PrivilegesRequired=lowest
OutputBaseFilename=mysetup
SetupIconsFile=C:\Users\User\OneDrive\Документи\941\2 семестр\Допомога проєктування AAAA\Матко\Проеа\biblio.ico
Compression=lzma
SolidCompression=yes
WizardStyle=modern

[Languages]
Name: "ukrainian"; MessagesFile: "compiler:Languages\Ukrainian.isl"

[Tasks]
Name: "desktopicon"; Description: "{#InstallDesktopIcon}"; GroupDescription: "{#InstallAdditionalIcons}"; Flags: unchecked

[Files]
Source: "C:\Users\User\OneDrive\Документи\941\2 семестр\Допомога проєктування AAAA\Матко\Проеа\{#MyAppExeName}"; DestDir: "{app}"; Flags: ignoreversion
Source: "C:\Users\User\OneDrive\Документи\941\2 семестр\Допомога проєктування AAAA\Матко\Проеа\{#MyAppExeName}.ico"; DestDir: "{app}"; Flags: ignoreversion

```

Рисунок 3.14 – Скрипт для створення установочного файлу

3.3 Інструкція користувача

Для входу в свій обліковий запис потрібно ввести логін 'admin' та пароль 'admin', його потрібно змінити. На кожній формі є кнопка 'Допомога', де можна знайти цю ж саму інструкцію та зворотній зв'язок з автором програми.

Після введення логіну та пароля користувачу відкривається головне вікно, де відображається список матеріалу та категорій, функції додавання даних, пошук та кнопка 'Керування користувачами'.

Для додавання нових даних потрібно заповнити 7 відповідних поля з текстом:

- Введіть назву матеріалу
- Введіть кількість матеріалу
- Введіть дату завезення
- Введіть категорію
- Введіть номер телефону
- Введіть статус

Далі натиснути кнопку 'Добавити дані', вони з'являться в списку.

Для фільтрування списку можна обрати одну з категорій, натиснувши два рази ЛКМ по вибраній категорії, щоб відобразити весь список, потрібно натиснути 'Показати все'.

Для пошуку можна використати поле, під назвою 'Пошук', у якому потрібно ввести повну назву матеріалу, дату або кількість пошук виконується по всім критеріям.

У списку, натиснувши ПКМ з'явиться контекстне меню з такими функціями:

1. Змінити – відкриється вікно де можна змінити любі дані.
2. Оновити список – оновляє список та відображає весь список.
3. Видалити – видаляє вибраний матеріал з бази даних.

Щоб керувати обліковими записами користувачів, потрібно натиснути кнопку 'Керування користувачами', відкриється вікно зі списком користувачів та кнопками 'зареєструвати', 'назад' та 'допомога'.

Щоб видалити користувача потрібно його обрати зі списку та натиснути ПКМ і в контекстному меню натиснути 'Видалити'.

Щоб додати нового користувача потрібно натиснути кнопку 'Зареєструвати', далі потрібно ввести логін та пароль 2 рази для підтвердження.

4. ОХОРОНА ПРАЦІ

Програмний додаток для підтримки діяльності підприємства з виготовлення металоконструкцій VESCO може бути застосований для працівників VESCO і тому проєктом передбачено створення трьох робочих місць в приміщенні цеху і дистанційно в приміщенні бухгалтерії.

Процес обслуговування комп'ютерної та офісної техніки користувачами супроводжується можливою дією шкідливих та небезпечних факторів, а саме: електромагнітні поля, статична електрика, шум, незадовільна освітленість робочого місця, неправильна організація робочого місця та інші.

Дія небезпечних факторів може призвести до нещасних випадків і травмувань. Для користувачів причиною нещасних випадків найчастіше можуть бути порушення в експлуатації устаткування, що живиться від електромережі, перевантаження мережі, порушення правил пожежної безпеки. Дія шкідливих виробничих факторів, які мають тривалий термін, може спричинити захворювання, найпоширенішими з яких є порушення зору, захворювання шкіри, кістково-м'язові, нервово-психічні порушення.

Більшість з цих проблем можуть бути зведені до мінімуму або усунуті взагалі у разі правильної організації робочого місця, створенні оптимальних санітарно-гігієнічних умов праці, дотримання правил техніки безпеки і раціонального розподілу робочого часу.

Площа приміщення в фірми, де розміщені комп'ютери, складає 32 метри квадратних, висота приміщення 3,4м. Приміщення облаштоване вікнами на південну сторону, системою централізованого опалення, природною вентиляцією. В приміщенні організовано три робочих місця з комп'ютерами, сканером і принтером. для кімнати площею 20 м², ширина якої складає 5м, довжина – 6м, висота – 3,4м. Приміщення має вихід назовні через коридор.

З ергономічного-погляду найкращим є плоский екран, який поглинає зовнішні світлові потоки, чим зменшує кількість відблисків. Оптимальна висота розташування екрана відповідає напрямленості зору оператора в секторі 50-35°

відносно до горизонталі. Якщо верхній край екрана вищий за рівень очей, то зчитування інформації з екрана ПК може викликати стан дискомфорту. Для організації робочого місця врахувалися антропометричні дані операторів, а також розміщення елементів обладнання залежно від характеру роботи, яку вони виконують. Робочі столи мають стабільну конструкцію: поверхня стола становить 180x90 см; стіл регулюється за висотою в діапазоні 65-85 см, висота від горизонтальної лінії зору до робочої поверхні стола становить 45-50 см. Висоту сидіння регулюють за зростом у межах 42-55 см.

Покриття поверхні столів без відблисків, з коефіцієнтом відбиття 30%, легко ; кути і передня панель дошки столів заокруглені. Сидіння комфортні, має заокруглені краї, нахилється відносно горизонталі вперед на 20° і назад на 140°, розмір його - 40x40 см. Висота спинки крісел - 48 - 50 см від поверхні сидіння. Підніжки крісел мають по п'ять опор, щоб запобігти його падінню.

Робочі місця з монітором відповідають таким вимогам:

- Відстань між бічними поверхнями ВДТ - 1,2 м;
- Відстань від поверхні одного ВДТ до екрана іншого ВДТ - 2,5 м;
- Прохід між рядами робочих місць - 1 м;
- Висота робочої поверхні робочого стола - 680-800 мм;
- Ширина робочої поверхні робочого стола - 600-1400 мм;
- Глибина робочої поверхні робочого стола - 800-1000 мм;
- Простір для ніг: заввишки - 600мм, завширшки - 500мм, завглибшки - 450 мм.

Ергономічні характеристики робочого місця:

Ступні ніг - на підставці для ніга бо на підлозі;

Стегна - в горизонтальній площині;

Передпліччя- вертикально;

Лікті- під кутом 70-90 градусів до вертикальної площини;

Зап'ястя– зігнуті під кутом не більше 20 градусів відносно горизонтальної площини;

Нахил голови - 15-20 градусів відносно вертикальної площини.

Монітор та клавіатура розташовані на оптимальній відстані від очей користувача, але не ближче 600мм, з урахуванням розміру алфавітно-цифрових знаків і символів, а також розміру екрана по діагоналі.

Для організації праці, що пов'язана з використанням монітору ЕОМ і ПЕОМ, для збереження здоров'я працюючих, запобігання професійним захворюванням і підтримання працездатності передбачені внутрішньо змінні режими при 8-годинному робочому дні для користувачів - 10 хвилин перерви через кожен годину роботи.

Режим праці та відпочинку тих, хто працює з ЕОМ, визначається залежно від виконуваної роботи відповідно до ДСН 3.3.2-007-98.

У всіх випадках тривалість безперервної роботи з ВДТ не перевищує 4 години.

Для проведення робіт було вибрано монітор, який за електромагнітним випромінюванням повністю відповідає стандарту **MPR II**, який має наступні технічні характеристики;

1. Параметри ЕПТ:

- розмір діагоналі 17-дюймів (видимий - 16 дюймів);
- екран з плоскою поверхнею, антистатичним та антибліковим покриттям;

- розмір зерна - 0,20 мм (відповідає MPR II); 4.

2. Частоти розгортки: $f_{гор}$ - 31—71 кГц; $i^*_{ерт}$ - .50- 160 Гц.

3. Максимальна роздільна здатність, пікс./Гц: 1600x 1200/68;

4. Рекомендована роздільна здатність, піксУГц: 1280x1024/75;
1024x768/85;

5. Живлення: 90 ~ 264В (47 ~ 63) Гц;

6. Споживання енергії: робочий режим: менше 68 Вт;

7.Рівні випромінювань відповідають міжнародним стандартам в галузі охорони праці:

Роботи в приміщенні належать до першої категорії важкості (легкі роботи 1а). Згідно вимог діючих стандартів оптимальні метеорологічні умови в таблиці 5.1

Таблиця 5.1 - Оптимальні метеорологічні умови

Пора року	Категорія робіт	Температура повітря, °С не більше	Відносна вологість повітря, %	Швидкість руху повітря, м/с
Холодна	Легка-Ia	22-24	40-60	0,1
Тепла	Легка-Ia	23-25	40-60	0,1

У приміщенні фірми мікроклімат відповідає оптимальним значенням параметрів, встановлених нормативами, що науково обґрунтовані. Для забезпечення дотримання постійності параметрів мікроклімату приміщення лабораторії обладнано системою опалення та кондиціонування повітря відповідно до ДБН В.2.2-3:2018.

Важливе місце в комплексі заходів з охорони праці працюючих займає утворення оптимального світлового середовища, тобто раціональна організація природного і штучного освітлення приміщення і робочих місць.

Приміщення фірми використовує бокове одностороннє природне освітлення з південною орієнтацією світла. Площа світлових прорізів становить 23 % площі підлоги; тому допустимо природне освітлення. Робочі столи розміщуємо подалі від вікон і так, щоб віконні прорізи знаходились зліва від робочого місця, а штучне світло падало з лівого або з правого боку залежно від розміщення обладнання. Проектом передбачено монтаж світлорозсіювальних штор .

У приміщенні, де знаходяться монітори, для запобігання накопиченню статичної електрики проектом передбачено підтримання відносної вологості повітря на рівні 50-60 % за допомогою побутового кондиціонера, або без нього в тому разі, коли природний стан відносної вологості відповідає нормам.

Важливе місце в комплексі заходів з охорони праці працюючих з ПК • займає утворення оптимального світлового середовища, тобто раціональна організація природного і штучного освітлення приміщення і робочих місць.

Штучне освітлення в приміщенні і на робочих місцях створюють добру видимість інформації на екрані ПК, машинописного і рукописного тексту та інших робочих матеріалів.

Для забезпечення нормальних умов роботи велику роль відіграє освітлення. Згідно з ДБН В.2.5-28-2006, проєктоване приміщення має природне та штучне освітлення. Рівень освітленості на робочому столі в зоні розташування документів становить 400-500лк.

Згідно з ДБН В.2.5-28-2006 нормативний коефіцієнт природного освітлення (КПО) для будівель, що розташовані в IV поясі для Кам'янця-Подільського складає 2,3%

У вечірні часи роботи, коли даний рівень освітленості системою загального освітлення недостатній, проєктом передбачено застосування світильників місцевого освітлення.(TIROSS, model NO: TS57).

Щоби не засвічувати екранів моніторів прямими світловими потоками, проєктом передбачено розміщення світильників загального освітлення збоку від робочого місця паралельно до лінії зору оператора.

В приміщенні присутні джерела значного шуму і вібрацій. Джерела виникнення шуму - це: вентилятори на системних блоках, оргтехніці та обладнанні для кондиціонування повітря.

У приміщенні рівні звукового тиску, рівні звуку та еквівалентні рівні звуку на робочих місцях відповідають вимогам, рівні шуму на робочих місцях користувачів визначені ДСН 3.3.6.037-99.

В блоці живлення та в критичних точках материнської плати (процесор, мікросхеми чіпсету) вмонтовані датчики, які програмно регулюють як моменти ввімкнення вентиляторів, так і їх швидкість обертання.

ПК живляться від мережі змінного струму ~ 220 В. Приміщення належить до таких, в яких безпека ураження електричним струмом II класу. Проєктом передбачено впровадження заходів попередження враження електричним струмом згідно з ПУЕ, а саме

- для захисту приладів від перенавантажень та коротких замикань використовується запобіжник;
- для забезпечення електробезпеки в аварійному режимі застосовується захисне заземлення;
- електронні стабілізатори напруги встановлені на щитку для запобігання перевантаження блоків живлення комп'ютерів.

Приміщення обладнане контуром-шиною захисного заземлення, яка з'єднується із заземлювачем. Технічні засоби безпеки праці на електрообладнанні зводяться до захисту від дотику до струмопровідних частин, захисту в разі переходу струму на не струмопровідні частини. Для захисту від дотику передбачена надійна ізоляція і розташування у недоступних місцях струмопровідних частин електрообладнання. Ізоляція мережі перевіряється двічі на рік.

Джерелами займання можуть бути електричні іскри, коротке замикання, перегріті опорні поверхні, несправність обладнання. Для усунення небезпеки загорання встановлений електрощиток з запобіжниками, а також використовуються додаткові електронні стабілізатори напруги, щоб запобігти перевантаженню блоків живлення комп'ютерів.

У випадку виникнення пожежі насамперед треба від'єднати джерело живлення, евакуювати людей і використовувати вуглекислотний вогнегасник. Як первинні заходи пожежогасіння проектом передбачено використання вогнегасників в кількості 2 штук: один встановлюємо безпосередньо приміщені, другий - в коридорі.

ЗАТВЕРДЖЕНО

Голова VESCO

20.06.2021 № 486

ІНСТРУКЦІЯ
з охорони праці № 2
для роботи з комп'ютерами та офісною
технікою

1. Загальні вимоги безпеки для роботи з комп'ютером та офісною технікою

1.1 До самостійної роботи з комп'ютером, принтером, та іншою оргтехнікою допускаються особи, які досягли 18 річного віку, пройшли медичний огляд, ознайомлені з інструкцією з охорони праці для роботи з офісною технікою, не мають протипоказань за станом здоров'я.

1.2. Під час роботи з комп'ютером та оргтехнікою на користувача можуть впливати наступні небезпечні та шкідливі фактори: електрострум і випромінювання; перенапруження зору під час роботи з електронними пристроями, монітором, особливо при нераціональному розташуванні екрана по відношенню до очей.

1.3. Освітлювальні установки повинні забезпечувати рівномірне освітлення і не повинні утворювати засліплюючих відблисків на клавіатурі, а також на екрані монітора за напрямом очей.

1.4. В процесі роботи з комп'ютером, принтером, ксероксом та іншою периферійною технікою не допускається розташування робочого місця в приміщеннях без природного освітлення, без наявності природної або штучної вентиляції.

1.5. Робоче місце з комп'ютером та оргтехнікою повинно розміщуватися на відстані не менше 1м від стіни, від стіни з віконними отворами – на відстані не менше 1,5 м.

1.6. Кут нахилу екрана монітора або ноутбука по відношенню до вертикалі повинен складати 10-15 градусів, а відстань до екрана - 500-600 мм.

1.7. Кут зору екрана повинен бути прямим і становити 90 градусів.

1.8. Для захисту від прямих сонячних променів повинні передбачатися сонцезахисні пристрої (плівка з металізованим покриттям, регульовані жалюзі з вертикальними панелями та ін).

1.9 Освітлення повинно бути змішаним (природним та штучним).

1.10. У приміщенні кабінету і на робочому місці необхідно підтримувати чистоту і порядок, проводити систематичне провітрювання.

1.11. Про всі виявлені під час роботи несправності обладнання необхідно доповісти керівнику, у випадку поломки необхідно припинити роботу до усунення аварійних обставин. При виявленні можливої небезпеки, попередити оточуючих та негайно повідомити керівнику; утримувати в чистоті робоче місце, не захаращувати його сторонніми предметами.

1.12. Про нещасний випадок очевидець, працівник, який його виявив, або сам потерпілий повинні доповісти безпосередньо керівникові установи і вжити заходів з надання медичної допомоги.

1.13. Особи, винні в порушенні вимог даної інструкції з охорони праці для роботи з комп'ютером, принтером, ксероксом та іншою оргтехнікою, притягаються до дисциплінарної відповідальності у відповідності з чиним законодавством.

2. Вимоги безпеки перед початком роботи з комп'ютером та офісною технікою

2.1. Оглянути і переконатися у справності обладнання, електродротів. У разі виявлення несправностей, до роботи не приступати. Повідомити про це керівника і, тільки після усунення несправностей і його дозволу, приступити до роботи.

2.2. Перевірити освітлення робочого місця, за необхідності, вжити заходів до його нормалізації.

2.3. Перевірити наявність та надійність захисного заземлення устаткування.

2.4. Перевірити стан електричного шнура і вилки.

2.5. Перевірити справність вимикачів та інших органів управління персональним комп'ютером та офісної техніки.

2.6. В разі виявлення будь-яких несправностей, комп'ютер та оргтехніку не вмикати і негайно повідомити про це керівника відділу.

2.7. Ретельно провітрити приміщення з персональним комп'ютером та оргтехнікою, переконатися, що мікроклімат у приміщенні знаходиться допустимих межах: температура повітря в холодний період року 22-24°C, в теплий період року - 23-25°C, відносна вологість повітря — 40-60%.

2.8. Ввімкнути монітор і перевірити стабільність і чіткість зображення на екрані, переконатися у відсутності запаху диму від комп'ютера та оргтехніки.

3. Вимоги безпеки під час роботи з комп'ютером, та іншою офісною технікою

3.1. Вмикайте і вимикайте комп'ютер, ноутбук та іншу офісну техніку шляхом вмикання чи відімкнення фільтра, а потім витягніть вилку з розетки чи ввімкніть вилку в розетку.

3.2. Забороняється знімати захисні пристрої з обладнання і працювати без них.

3.3. Не допускати до комп'ютера та оргтехніки сторонніх осіб, які наберуть участі в роботі.

3.4. Забороняється переміщати та переносити системний блок, монітор, принтер, будь-яке обладнання, яке знаходиться під напругою.

3.5. Забороняється під час роботи пити будь-які напої, приймати їжу.

3.6. Забороняється будь-яке фізичне втручання у пристрій комп'ютера, принтера, сканера, ксерокса під час їх роботи.

3.7. Забороняється залишати включене обладнання без нагляду.

3.8. Забороняється класти предмети на комп'ютерне обладнання, монітори, екрани та оргтехніку.

3.9. Суворо виконувати загальні вимоги з електробезпеки та пожежної безпеки.

3.10. Під час усунення застрявання паперу на принтері, задля уникнення ураження електрострумом, необхідно відключити обладнання від електромережі. Необхідно також вимикати обладнання від мережі при тривалому простої.

3.11. Самостійно розбирати та проводити ремонт електронної та електронно-механічної частини комп'ютера, периферійних пристроїв, оргтехніки категорично забороняється. Ці роботи може виконувати тільки спеціаліст або інженер з технічного обслуговування комп'ютерної техніки.

3.12. Сумарний час безпосередньої роботи з персональним комп'ютером та іншою оргтехнікою протягом робочого дня має бути не більше 6 годин.

3.13. Тривалість безперервної роботи з персональним комп'ютером та іншою оргтехнікою без регламентованої перерви не повинна перевищувати 2-х годин. Через кожну годину роботи слід робити перерву тривалістю 15 хв.

3.14. Під час регламентованих перерв, з метою зниження нервово-емоційного напруження, стомлення зорового аналізатора, усунення впливу гіподинамії та гіпокінезії, запобігання розвитку познотонічного стомлення, слід виконувати комплекси вправ для очей або організовувати фізкультурні паузи.

3.15. Комп'ютер, будь-які його периферійні пристрої, оргтехніку необхідно використовувати у відповідності з експлуатаційною документацією до них.

3.16. Під час виконання роботи необхідно бути уважним, не звертати уваги на сторонні речі.

3.17. Про всі виявлені несправності та збої в роботі апаратури необхідно повідомити безпосередньо керівника підрозділу.

4. Вимоги безпеки після закінчення роботи з комп'ютером

4.1. Вимкнути комп'ютер, ноутбук, телевізор, плазмову панель, LCD-екран, принтер, ксерокс, сканер, колонки та іншу оргтехніку від електромережі,

для чого необхідно вимкнути тумблери, а потім акуратно, притримуючи розетку, витягнути штепсельні вилки з розетки.

4.2. Протерти зовнішню поверхню комп'ютера чистою вологою тканиною. При цьому не допускайте використання розчинників, одеколону ,препаратів в аерозольній упаковці.

4.3. Прибрати робоче місце.

4.4. Ретельно провітрити приміщення з персональним комп'ютером та іншою оргтехнікою.

За порушення інструкцій з охорони праці винні особи притягуються до відповідальності згідно законодавства (ст 49 Закону України "Про охорону праці").

З інструкцією ознайомлений _____ Олійник В.В.

(підпис)

(Прізвище ініціали)

Посада відповідального
за стан охорони праці _____

Онятицька О.М.

(підпис)

(Прізвище ініціали)

20 червня 2021р.

ВИСНОВКИ

У Кваліфікаційній роботі розроблено програмний додаток для підтримки діяльності підприємства з виготовлення металоконструкцій для VESCO, яка забезпечує швидкий та зручний доступ до інформації.

Основна перевага створеної системи є логічна та продумана модель бази, а також те, що інформаційна система розроблялась у найновішому та самому поширеному на теперішній час форматі Microsoft access database. Це дає змогу відкривати та працювати з базою даних на різних комп'ютерах та з різними версіями програми.

При розробці програмного додатку було проаналізовано декілька аналогів за призначенням і аналіз довів переваги створеного додатку над існуючими аналогами.

Після проведеного дослідження були виділені основні процеси керування технічним відділом та проведений аналіз взаємодії розглянутого відділу з іншими об'єктами при виконанні замовлення. В результаті аналізу було виявлено завдання, автоматизація яких призведе до збільшення швидкості та зручність роботи з даними, які використовуються при керуванні процесу виконання проекту. Це забезпечить ефективну роботу координатора, і як результат всього підприємства. Саме для цих завдань було вирішено створити систему.

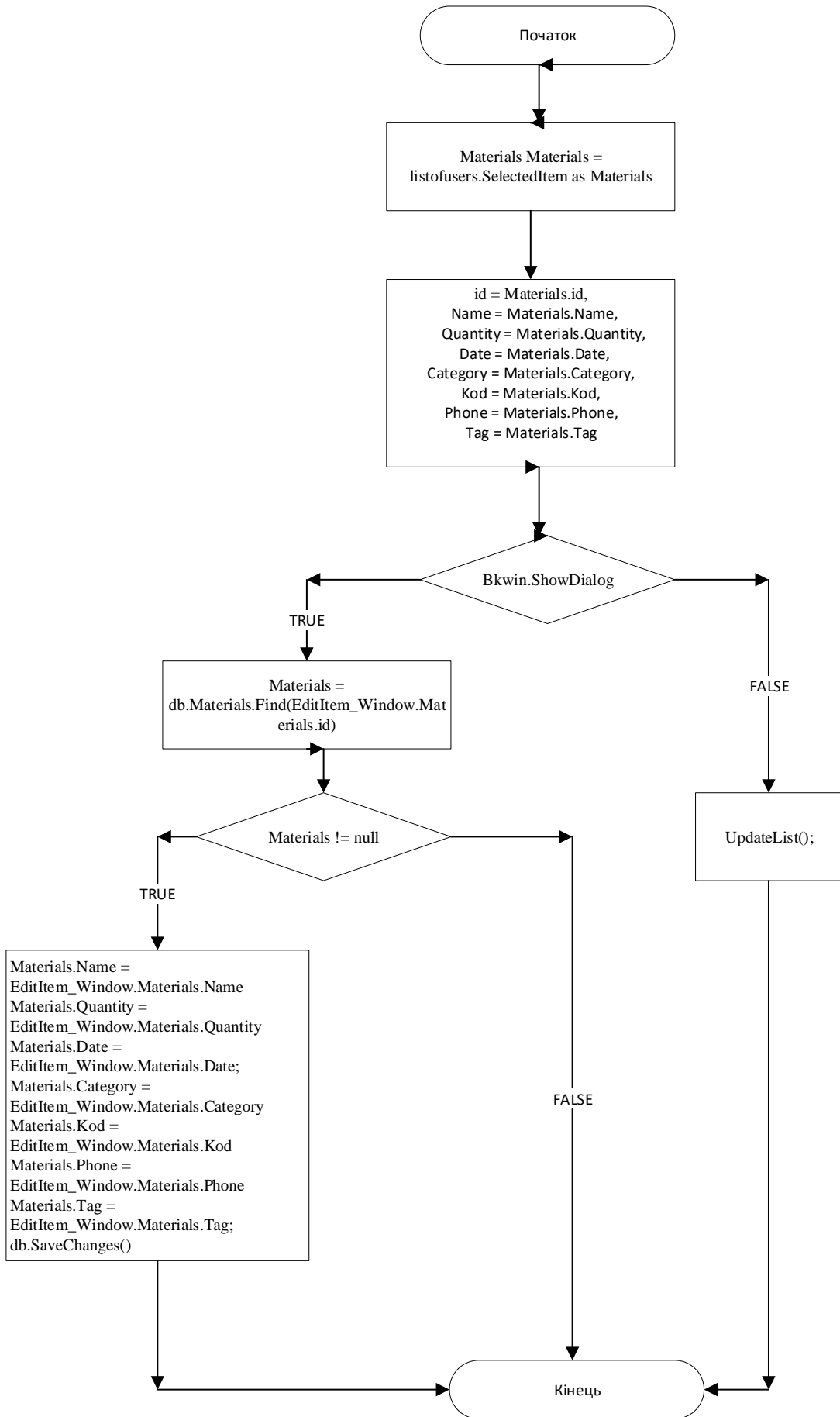
Для обґрунтування доцільності впровадження ІС було проведено розрахунок економічного ефекту. Результати показали, що завдяки прибутку, збільшення якого буде досягнуто завдяки автоматизації робіт з обліку постачання матеріалів, система окупиться за рік та три місяці. Ці результати підкріпили доцільність розробки системи.

Тому, на основі проведеного аналізу та отриманих даних було створено технічне завдання на проектування інформаційної системи, розроблено схему даних, на основі якої ця система працює і відповідно створено саму інформаційну систему для підприємства з виготовлення металоконструкцій.

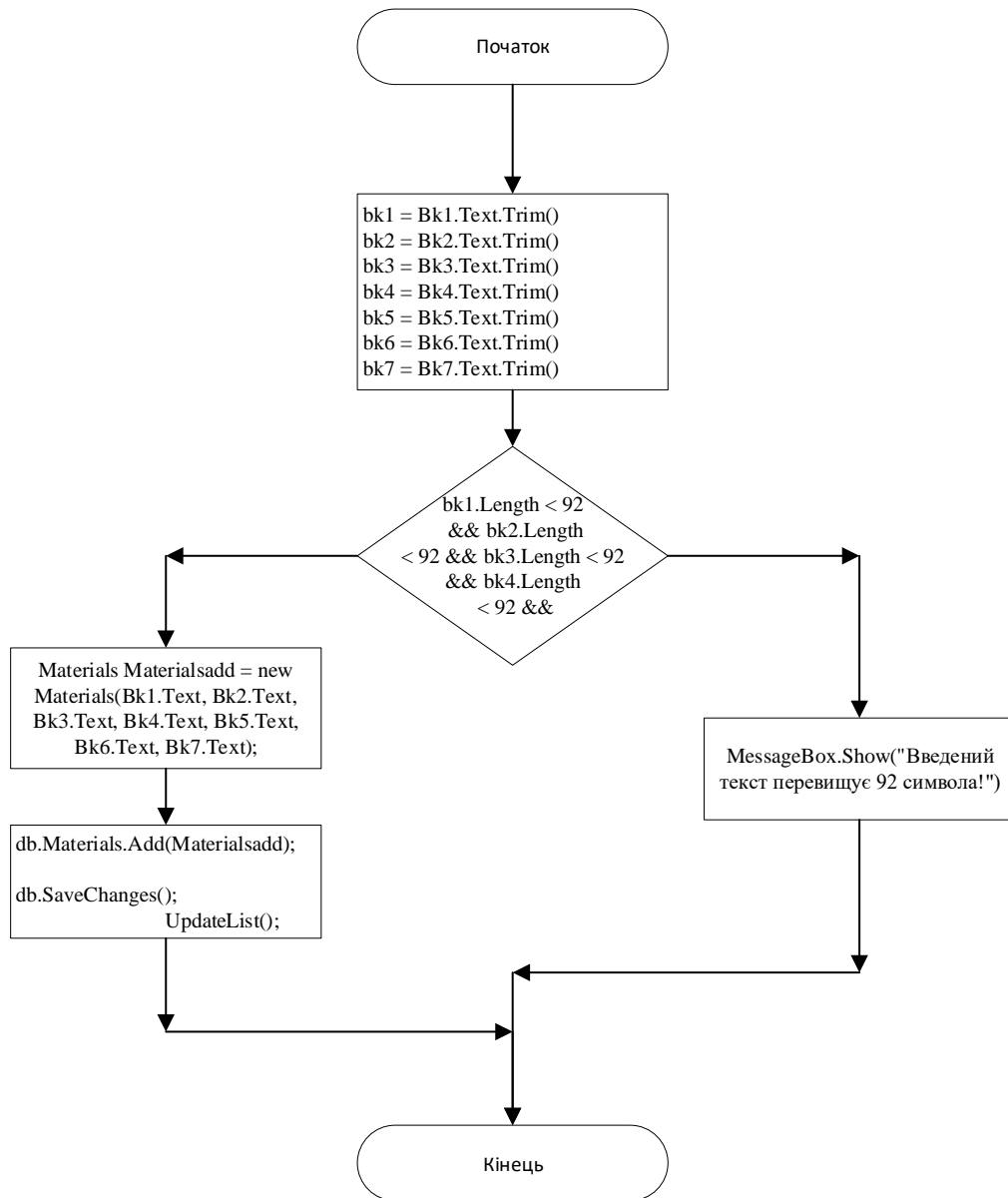
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ДСТУ 3008-2015. Документація. Звіти у сфері науки і техніки. Структура та правила оформлювання;
2. Постанова КМУ від 13.04.2011 № 461 «Питання прийняття в експлуатацію закінчених будівництвом об'єктів».
3. ДСТУ Б А.2.4-4:2009 «Основні вимоги до проектної та робочої документації».
4. <http://microsoft-free.com> - Коротка характеристика пакету офісних програм MicrosoftOffice 2010.
5. <https://metanit.com/sharp/wpf/21.1.php> - Підключення Microsoft access database в WPF.
6. <https://microsoft-access-databasebrowser.org> - DB Browser for Microsoft access database
7. https://uk.wikipedia.org/wiki/Microsoft_access_database -Microsoft access database Вікіпедія
8. https://uk.wikipedia.org/wiki/Системні_вимоги - Вікіпедія
9. webarty.net/databases/sravnenie-sql-baz-dannyh - Порівняння СУБД.
10. yak-prosto.com/yak-rozrahuvati-rentabelnist-proektu -Як розрахувати рентабельність проекту .
11. Єдина система конструкторської документації. Основні написи (ГОСТ 2.104-2006, IDT). ДСТУ ГОСТ 2.104:2006 / Вид. Офіційне. - Київ: Держспоживстандарт України, 2007р-16 с.
12. Текстові документи. Загальні вимоги СОУ 207.01:2004. - 3-тє вид., із змінами / В. Каплун, В. Олександренко, Л. Першина, Л. Безсмертна, О. Снозик. - Хмельницький: ХНУ, 2008. - 40 с.

ДОДАТКИ**Додаток А****Блок-схема до кнопки «Змінити»**



Блок-схема до кнопки «Добавити даних»



Додаток Б

Код програмного додатку

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
  
```

```

using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using System.Data.Entity;

namespace kurs_job_v12_NUXT_3_IT_SQL
{
    public partial class MainWindow : Window
    {
        ApplicationContext db;
        public MainWindow()
        {
            InitializeComponent();
            UpdateList();
        }

        private void UpdateList()
        {
            try
            {
                string connectionString = "Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=Database5.accdb;";
                using (OleDbConnection connection = new OleDbConnection(connectionString))
                {
                    connection.Open();

                    // Отримання всіх матеріалів
                    string selectAllQuery = "SELECT * FROM ПостачанняМатеріалів";
                    using (OleDbCommand selectAllCommand = new OleDbCommand(selectAllQuery,
connection))
                    using (OleDbDataReader reader = selectAllCommand.ExecuteReader())
                    {
                        List<Materials> materialsList = new List<Materials>();

```

```

while (reader.Read())
{
    Materials material = new Materials
    {
        // Припускається, що ваші стовпці мають відповідні типи даних, імена
        // стовпців слід замінити на ваші
        id = Convert.ToInt32(reader["Ідентифікатор"]),
        Name = Convert.ToString(reader["NameMaterial"]),
        Quantity = Convert.ToString(reader["Quantity"]),
        Category = Convert.ToString(reader["Category"]),
        Date = Convert.ToString(reader["Date"]),
        Phone = Convert.ToString(reader["Phone"]),
        Tag = Convert.ToString(reader["Tag"]),
        // і так далі, відповідно до вашої моделі Materials
    };

    materialsList.Add(material);
}

listofusers.ItemsSource = materialsList;
}

// Отримання унікальних категорій
string selectDistinctCategoriesQuery = "SELECT DISTINCT Category FROM
Materials";
using (OleDbCommand selectDistinctCategoriesCommand = new
OleDbCommand(selectDistinctCategoriesQuery, connection))
using (OleDbDataReader categoryReader =
selectDistinctCategoriesCommand.ExecuteReader())
{
    List<string> distinctCategories = new List<string>();

    while (categoryReader.Read())
    {

```

```

        distinctCategories.Add(Convert.ToString(categoryReader["Category"]));
    }

    listofcategories.ItemsSource = distinctCategories;
}
}
}
catch (Exception ex)
{
    MessageBox.Show($"Exception caught: {ex.Message}");
} }

private void Button_Click_Help(object sender, RoutedEventArgs e)
{
    Help_Window Help_Window = new Help_Window();
    Help_Window.Show();
}

private void Button_Click_AddMaterials(object sender, RoutedEventArgs e)
{
try
    {
        string bk1 = Bk1.Text.Trim();
        string bk2 = Bk2.Text.Trim();
        string bk3 = Bk3.Text.Trim();
        string bk4 = Bk4.Text.Trim();
        string bk5 = Bk5.Text.Trim();
        string bk6 = Bk6.Text.Trim();
        string bk7 = Bk7.Text.Trim();
        if (bk1.Length != 0 && bk2.Length != 0 && bk3.Length != 0)
        {
            if (bk6.IndexOf("+") != 0 && bk6 == "")
            {
                if (bk1.Length < 92 && bk2.Length < 92 && bk3.Length < 92 && bk4.Length < 92
&& bk5.Length < 92 && bk6.Length < 92 && bk7.Length < 92)

```

```

        {
            Materials Materialsadd = new Materials(Bk1.Text, Bk2.Text, Bk3.Text, Bk4.Text,
Bk5.Text, Bk6.Text, Bk7.Text);
            db.Materials.Add(Materialsadd);
            db.SaveChanges();
            UpdateList();
        }
        else
        {
            MessageBox.Show("Введеный текст перевищує 92 символа!");
        }
    }
    else {
        MessageBox.Show("Введено не проавильно номер телефону!");
    }
}
else
{
    MessageBox.Show("Введеный текст менше 1 символа!");
}
}
catch (ArgumentNullException ex)
{
    MessageBox.Show("{0} First exception caught." + ex);
}
catch (Exception ex)
{
    MessageBox.Show("{0} Second exception caught." + ex);
} }
private void Button_Click_Admin_panel(object sender, RoutedEventArgs e)
{
    Admin_Panel Admin_Panel = new Admin_Panel();
    Admin_Panel.Show();
    Close();
}

```

```
private void Listofusers_MouseDoubleClick(object sender, MouseButtonEventArgs e)
{
    MessageBox.Show(«Индекс: « + listofusers.SelectedIndex);
}
```

```
private void MenuItem_Click_Edit(object sender, RoutedEventArgs e)
{
    try
    {
        if (listofusers.SelectedItem == null) return;
        Materials Materials = listofusers.SelectedItem as Materials;

        EditItem_Window Bkwin = new EditItem_Window(new Materials
        {
            id = Materials.id,
            Name = Materials.Name,
            Quantity = Materials.Quantity,
            Date = Materials.Date,
            Category = Materials.Category,
            Kod = Materials.Kod,
            Phone = Materials.Phone,
            Tag = Materials.Tag
        });

        if (Bkwin.ShowDialog() == true)
        {
            Materials = db.Materials.Find(EditItem_Window.Materials.id);
            if (Materials != null)
            {
                Materials.Name = EditItem_Window.Materials.Name;
                Materials.Quantity = EditItem_Window.Materials.Quantity;
                Materials.Date = EditItem_Window.Materials.Date;
                Materials.Category = EditItem_Window.Materials.Category;
                Materials.Kod = EditItem_Window.Materials.Kod;
```

```

        Materials.Phone = EditItem_Window.Materials.Phone;
        Materials.Tag = EditItem_Window.Materials.Tag;
db.Entry(Materials).State = EntityState.Modified;
        db.SaveChanges();
    }
}
UpdateList();
}
catch (ArgumentNullException ex)
{
    MessageBox.Show("{0} First exception caught." + ex);
}
catch (Exception ex)
{
    MessageBox.Show("{0} Second exception caught." + ex);
}
}
}
catch (ArgumentNullException ex)
{
    MessageBox.Show(«{0} First exception caught.» + ex);
}
catch (Exception ex)
{
    MessageBox.Show(«{0} Second exception caught.» + ex);
}
}

private void MenuItem_Click_Update(object sender, RoutedEventArgs e)
{
    UpdateList();
}

private void MenuItem_Click_Delete(object sender, RoutedEventArgs e)
{

```

```

try
{
    MessageBoxResult messageBoxResult = System.Windows.MessageBox.Show(«Are you
sure?», «Delete Confirmation», System.Windows.MessageBoxButton.YesNo);
    if (messageBoxResult == MessageBoxResult.Yes)
    {
        if (listofusers.SelectedItem != null)
        {
            Materials Materialsrem = listofusers.SelectedItem as Materials;
            db.Materials.Remove(Materialsrem);
            db.SaveChanges();
            UpdateList();
        }
    }
}
catch (ArgumentNullException ex)
{
    MessageBox.Show(«{0} First exception caught.» + ex);
}
catch (Exception ex)
{
    MessageBox.Show(«{0} Second exception caught.» + ex);
}
}

private void SeachBox_TextChanged(object sender, TextChangedEventArgs e)
{
    try
    {
        if (SeachBox.Text != «»)
        {
            List<Materials> Materials = db.Materials.Where(a => a.Name == SeachBox.Text ||
a.Author == SeachBox.Text || a.Date == SeachBox.Text).ToList();
            listofusers.ItemsSource = Materials;
        }
    }
}

```

```

else
{
    UpdateList();
}
}
catch (ArgumentNullException ex)
{
    MessageBox.Show(«{0} First exception caught.» + ex);
}
catch (Exception ex)
{
    MessageBox.Show(«{0} Second exception caught.» + ex);
}
}

private void Listofcategories_MouseDoubleClick(object sender, MouseButtonEventArgs e)
{
    try
    {
        if (listofcategories.SelectedItem.ToString() != «»)
        {
            //MessageBox.Show(listofcategories.SelectedItem.ToString());
            string ss = listofcategories.SelectedItem.ToString();
            List<Materials> Materials = db.Materials.Where(a => a.Category == ss).ToList();
            listofusers.ItemsSource = Materialsf;
        }
        else
        {
            UpdateList();
        }
    }
    catch (ArgumentNullException ex)
    {
        MessageBox.Show(«{0} First exception caught.» + ex);
    }
}

```

```
catch (Exception ex)
{
    MessageBox.Show(«{0} Second exception caught.» + ex);
}
}

private void Button_Click_Updatedlist(object sender, RoutedEventArgs e)
{
    UpdateList();
}
}
}
```