

ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЕКТ

на тему:

«Проектування програмного забезпечення системи управління технічним обслуговуванням промислового комплексу»

Студентки 2 курсу, 23 групи,
спеціальності 121 «Інженерія
програмного забезпечення»
спеціалізації «Інженерія
програмного забезпечення»

підпис студента

**Балюк
Марії Вікторівни**

Науковий керівник
кандидат технічних наук,
доцент кафедри інженерії
програмного забезпечення та
кібербезпеки

підпис керівника

**Рзаєва Світлана
Леонідівна**

Гарант освітньої програми
доктор економічних наук,
професор кафедри інженерії
програмного забезпечення та
кібербезпеки

підпис керівника

**Токар Володимир
Володимирович**

Київський національний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь магістр

Спеціальність 121 «Інженерія програмного забезпечення»

Затверджую

Зав. кафедри інженерії програмного
забезпечення та кібербезпеки

Криворучко О. В.

«29» грудня 2020 р.

Завдання на випускний кваліфікаційний проект студентіві

Балюн Марія Вікторіна

(прізвище, ім'я, по батькові)

1. Тема випускного кваліфікаційного проекту «Проектування програмного
забезпечення системи управління технічним обслуговуванням промислового
комплексу»

Затверджена наказом ректора від «28» грудня 2020 р. № 3923

2. Строк здачі студентом закінченого проекту (роботи) 25 листопада 2021 р.

3. Цільова установка та вихідні дані до проекту (роботи)

Мета проекту проективання програмного забезпечення системи управління
технічного обслуговування, яка направлена на забезпечення персоналу
промислового комплексу підприємства комп'ютеризованою системою, що
мінімізує час на організацію та управління технічним обслуговуванням та
сприятиме усуненню чинників, які знижують ефективність роботи устаткування

Об'єкт дослідження система управління технічним обслуговуванням

Предмет дослідження процес проведення технічного обслуговування на
промисловому комплексі підприємства

4. Консультанти роботи із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускного кваліфікаційного проекту (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1 ОРГАНІЗАЦІЙНА СТРУКТУРА ПРОМИСЛОВОГО КОМПЛЕКСУ ПІДПРИЄМСТВА

1.1. Аналіз сфери діяльності промислового комплексу на підприємстві

1.2. Особливості комп'ютеризованих систем управління технічним обслуговування промислового комплексу

1.3. Мета та задачі системи технічного обслуговування промислового комплексу

1.4. Висновки до розділу 1

РОЗДІЛ 2 ПРОЕКТУВАННЯ АРХІТЕКТУРИ ТА СТРУКТУРИ ДАНИХ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1. Системні вимоги щодо системи управління технічним обслуговуванням

2.2. Архітектура системи управління технічним обслуговуванням

2.2.1. Модель інформаційної системи в розрізі програмного рівня

2.2.2. Модель класів

2.2.3. Модель інформаційної системи в розрізі користувацького рівня

2.2.4. Модель компонентів управління інформаційної системи

2.3. Проектування структурної моделі бази даних системи управління технічним обслуговуванням

2.4. Висновки до розділу 2

РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1. Функціональні вимоги до системи управління технічним обслуговуванням

3.1.1. Авторизація користувача

3.1.2. Функціональні можливості та компоненти системи

3.2. Висновки до розділу 3

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ТЕХНІЧНЕ ЗАВДАННЯ

ПРОГРАМА ТА МЕТОДИКА ТЕСТУВАННЯ

КЕРІВНИЦТВО КОРИСТУВАЧА

КЕРІВНИЦТВО СИСТЕМНОГО ПРОГРАМІСТА

ДОДАТКИ

6. Календарний план виконання проекту

№ пор.	Назва етапів випускного кваліфікаційного проекту	Строк виконання етапів проекту	
		за планом	фактично
1	2	3	4
1.	<i>Вибір теми випускного кваліфікаційного проекту</i>	21.09.2020	21.09.2020
2.	<i>Розробка та затвердження завдання на проект магістра (стац/заоч)</i>	29.12.2020	29.12.2020
3.	<i>Вступ та перелік літературних джерел</i>	27.02.2021	27.02.2021
4.	<i>Розробка технічного завдання</i>	20.03.2021	20.03.2021
5.	<i>Розділ 1. Організаційна структура промислового комплексу підприємства</i>	16.04.2021	16.04.2021
6.	<i>Розділ 2. Проектування архітектури та структури даних програмного забезпечення</i>	24.05.2021	24.05.2021
7.	<i>Розділ 3. Реалізація програмного забезпечення</i>	21.06.2021	21.06.2021
8.	<i>Розділ 4. (Об'єднання розділів 3 та 4)</i>	20.09.2021	20.09.2021
9.	<i>Розробка програми та методики тестування</i>	18.10.2021	18.10.2021
10.	<i>Написання наукової статті</i>	22.05.2021	22.05.2021
11.	<i>Керівництво користувача</i>	21.10.2021	21.10.2021
12.	<i>Висновки та пропозиції</i>	01.11.2021	01.11.2021
13.	<i>Здача випускного кваліфікаційного проекту на кафедрі (перша перевірка)</i>	03.11.2021	03.11.2021
14.	<i>Підготовка автореферату та презентації доповіді</i>	03.11.2021	03.11.2021
15.	<i>Попередній захист випускного кваліфікаційного проекту</i>	22.11.2021 – 25.11.2021	22.11.2021
16.	<i>Здача зброшурованої випускного кваліфікаційного проекту</i>	25.11.2021	25.11.2021
17.	<i>Зовнішнє рецензування випускного кваліфікаційного проекту</i>	26.11.2021	26.11.2021
18.	<i>Підготовка до публічного захисту випускного кваліфікаційного проекту</i>		

7. Дата видачі завдання «29» грудня 2020 р.

8. Науковий керівник випускного кваліфікаційного проекту _____

Рзаєва С.Л.

(прізвище, ініціали, підпис)

9. Гарант освітньої програми _____

Токар В.В.

(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент _____

Балюн М.В.

(прізвище, ініціали, підпис)

АНОТАЦІЯ

Відповідно до мети дослідження випускний кваліфікаційний проект присвячений проектуванню програмного забезпечення системи управління технічним обслуговуванням промислового комплексу, для гарантування виконання процесів обліку та контролю виконання робіт з технічного обслуговування устаткування згідно норм трудових витрат.

В результаті порівняння та аналізу додатків аналогів визначено, що даний проект є конкурентоспроможним у сфері промислової діяльності комплексу. Спроектовано архітектуру програмного забезпечення на основі моделей: розгортання, класів, варіантів використання та діяльності.

Проект виконаний з використанням клієнт-серверної технології в межах локальної мережі промислового комплексу. Основою проекту є реалізація бази даних, що розташована на серверній частині та розробка програмного забезпечення для Windows, використовуючи середовище розробки Visual Studio Windows Forms та об'єктно-орієнтованої мови C#.

Ключові слова: система управління, технічне обслуговування, промисловий комплекс.

ABSTRACT

According to the purpose the final qualification project is dedicated to the design of computerized maintenance management system of the industrial complex, accounting and control over the performance of maintenance in accordance with the norms of labor.

As a result of comparison and analysis of analogue applications, it is determined that this project is competitive in activities of the industrial complex. The software architecture is designed based on deployment models: classes, use case and activity.

The project is executed using the client-server technology within the local network of the industrial complex. The basis of the project is to implement a database located on the server and software development for Windows using the Visual Studio Windows Forms and object-oriented language C#.

Keywords: management system, maintenance, industrial complex.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

АтаТЗ	– автоматика та технологічний захист;
ЗВТ	– засоби вимірювальної техніки;
НБК	– Новий безпечний конфайнмент;
РАВ	– радіоактивні відходи;
ТЗВТ	– технічні засоби вимірювальної техніки;
ТО	– технічне обслуговування;
SCADA	– система диспетчерського управління та збору даних;
SQL	– Structured Query Language;
MS SQL	– Microsoft SQL Server.

<i>КНТЕУ 121 023-01.MP</i>							
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>			
Зав. каф.		Криворучко О.В.		29.09.20			
Керівник		Рзаєва С.Л.		29.09.20			
Гарант		Токар В.В.		29.09.20			
Розробив		Балюн М.В.		29.09.20			
<i>Перелік умовних скорочень</i>							
<i>Проектування програмного забезпечення системи управління технічним обслуговуванням промислового комплексу</i>					<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
					<i>ПС</i>	<i>2</i>	<i>40</i>
					<i>Факультет інформаційних технологій 2 курс, 23 група</i>		

ЗМІСТ

ВСТУП.....	5
РОЗДІЛ 1 ОРГАНІЗАЦІЙНА СТРУКТУРА ПРОМИСЛОВОГО КОМПЛЕКСУ ПІДПРИЄМСТВА.....	7
1.1. Аналіз сфери діяльності промислового комплексу на підприємстві	7
1.2. Особливості комп'ютеризованих систем управління технічним обслуговуванням промислового комплексу.....	9
1.3. Мета та задачі системи технічного обслуговування промислового комплексу.....	12
1.4. Висновки до розділу 1	13
РОЗДІЛ 2 ПРОЕКТУВАННЯ АРХІТЕКТУРИ ТА СТРУКТУРИ ДАНИХ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	14
2.1. Системні вимоги до системи управління технічним обслуговуванням.....	14
2.2. Архітектура системи управління технічним обслуговуванням	16
2.2.1. Модель інформаційної системи в розрізі програмного рівня	16
2.2.2. Модель класів	17
2.2.3. Модель інформаційної системи в розрізі користувацького рівня	20
2.2.4. Модель компонентів управління інформаційної системи	21
2.3. Проектування структурної моделі бази даних системи управління технічним обслуговуванням	23
2.4. Висновки до розділу 2	24
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	25
3.1. Функціональні вимоги до системи управління технічним обслуговуванням	25
3.1.1. Авторизація користувача	25
3.1.2. Функціональні можливості та компоненти системи	26

Зм.	Аркуші	№ докум.	Підпис	Дата	<i>КНТЕУ 121 02з-01.МР</i>			
Зав. каф.		Криворучко О.В.		29.09.20	Проектування програмного забезпечення системи управління технічним обслуговуванням промислового комплексу	Стадія	Аркуші	Аркуші
Керівник		Рзаєва С.Л.		29.09.20		3	3	40
Гарант		Токар В.В.		29.09.20		Факультет інформаційних технологій 2 курс, 2з група		
Розробив		Баллон М.В.		29.09.20				
					Зміст			

3.2. Висновки до розділу 3	36
ВИСНОВКИ ТА ПРОПОЗИЦІЇ	37
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	39
ТЕХНІЧНЕ ЗАВДАННЯ.....	41
ПРОГРАМА ТА МЕТОДИКА ТЕСТУВАННЯ.....	47
КЕРІВНИЦТВО КОРИСТУВАЧА	49
КЕРІВНИЦТВО СИСТЕМНОГО ПРОГРАМІСТА	51
ДОДАТКИ.....	53

					<i>КНТЕУ 121 023-01.МР</i>	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		4

ВСТУП

Актуальність. В даний час державні підприємства галузі ядерної промисловості України зустрічаються з критичними проблемами в процесі управління технологічними процесами у зв'язку з їх витратністю та простоями устаткування. Основними причинами такої ситуації є обмеження бюджету, геополітичні та економічні фактори, застарілі методології проведення технічного обслуговування устаткування та незацікавленість керівництва щодо підвищення кваліфікаційного рівня інженерно-технічного персоналу.

В свою чергу державні підприємства світового значення, що експлуатують промислові комплекси, обмежені у використанні програмних ресурсів та інструментів в сфері технічного обслуговування, використовуючи й надалі принципи неавтоматизованого управління ресурсами. Системи, що розроблені іноземним ринком програмного забезпечення, є витратними для розгортання, ліцензування та технічної підтримки на промислових підприємствах. Ці обмеження напряму впливають на швидкість перетворення способів організації виробництва, модернізацію устаткування та ефективність роботи промислових комплексів.

З огляду на вищезазначені обмеження, технічне обслуговування відіграє основну роль на всіх етапах життєвого циклу технологічного процесу. Тому, в такому випадку необхідно впроваджувати комп'ютеризовані системи управління технічним обслуговуванням на промислових комплексах державних підприємств як основний програмний ресурс, що забезпечує персонал інформацією щодо стану устаткування, виведення з експлуатації, періодичності проведення технічного обслуговування та оптимізує процес документування та подання інформації у формі звітності [4].

Зм.	Аркуш	№ докум.	Підпис	Дата	<i>КНТЕУ 121 02з-01.МР</i>			
Зав. каф.		Криворучко О.В.		27.02.21	<i>Проектування програмного забезпечення системи управління технічним обслуговуванням промислового комплексу</i>	Стадія	Аркуш	Аркушів
Керівник		Рзасва С.Л.		27.02.21		В	5	40
Гарант		Токар В.В.		27.02.21		<i>Факультет інформаційних технологій 2 курс, 2з група</i>		
Розробив		Баллон М.В.		27.02.21				
<i>Вступ</i>								

Мета проекту: проектування програмного забезпечення системи управління технічного обслуговування, яка направлена на забезпечення персоналу промислового комплексу підприємства комп'ютеризованою системою, що мінімізує час на організацію та управління технічним обслуговуванням та сприятиме усуненню чинників, які знижують ефективність роботи устаткування.

Об'єкт дослідження: система управління технічним обслуговуванням.

Предмет дослідження: процес проведення технічного обслуговування на промисловому комплексі підприємства.

У відповідності з метою проектування програмного забезпечення поставлені наступні завдання:

- провести аналіз організаційної структури промислового комплексу підприємства;
- описати системні та функціональні вимоги до розробки програмного забезпечення;
- розробити моделі архітектури та структури системи управління технічним обслуговуванням;
- розробити програмне забезпечення системи управління технічним обслуговуванням;
- скласти технічне завдання для подальшої співпраці із Замовником та впровадження системи;
- розробити керівництво користувача та системного програміста.

Наукова новизна дослідження полягає в тому, що розробка програмного забезпечення даного типу системи буде впроваджена на промисловому комплексі Новий Безпечний Конфайнмент ДСП «Чорнобильська АЕС», де на даний час відсутня така система у зв'язку з завершення пробної експлуатації комплексу.

Практичне значення дослідження полягає в тому, що розроблена система забезпечує повну прозорість та контроль операцій з управління та організації технічного обслуговування.

					<i>КНТЕУ 121 02з-01.МР</i>	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		6

РОЗДІЛ 1

ОРГАНІЗАЦІЙНА СТРУКТУРА ПРОМИСЛОВОГО КОМПЛЕКСУ ПІДПРИЄМСТВА

1.1. Аналіз сфери діяльності промислового комплексу на підприємстві

На території ДСП «Чорнобильської АЕС» розміщений промисловий комплекс – Новий безпечний конфайнмент, що побудований у 2019 році та створений у рамках основного завдання державного та міжнародного плану здійснення заходів на об'єкті «Укриття», спрямованого на перетворення цього об'єкту в екологічно безпечну систему.

Новий безпечний конфайнмент (НБК) – це унікальний об'єкт та захисна споруда у вигляді арки з сталевих конструкцій, що включає в себе комплекс технологічного обладнання, призначеного для видалення матеріалів, що містять ядерне паливо зі зруйнованого енергоблоку №4 ДСП «Чорнобильської АЕС», для переробки РАВ та інші системи, що призначені для підтримки та забезпечують безпеку персоналу, населення і навколишнього середовища [11].

З точки зору процесів, що відбуваються на промисловому комплексі, до складу НБК входять п'ять основних об'єктів:

а) Арка (АС – ідентифікаційне позначення об'єкту для проектування системи управління) включає в себе:

- 1) кільцевий об'єм – об'єм між зовнішньою і внутрішньою обшивкою арки НБК;
- 2) головний об'єм – простір під внутрішньою обшивкою арки, який разом з кільцевим об'ємом замикає безпечне укриття над об'єктом «Укриття» і забезпечений первинними засобами для виконання завдань по демонтажу об'єкту.

Зм.	Аркуш	№ докум.	Підпис	Дата	<i>КНТЕУ 121 02з-01.МР</i>			
Зав. каф.		Криворучко О.В.		16.04.21	Проектування програмного забезпечення системи управління технічним обслуговуванням промислового комплексу Організаційна структура промислового комплексу підприємства	Стадія	Аркуш	Аркушів
Керівник		Рзасва С.Л.		16.04.21		Р1	7	40
Гарант		Токар В.В.		16.04.21		Факультет інформаційних технологій 2 курс, 2з група		
Розробив		Баллон М.В.		16.04.21				

б) Технологічний будинок (ТС – ідентифікаційне позначення об'єкту для проектування системи управління) включає приміщення, зони та ділянки, які роблять можливим функціонування систем контролю, життєзабезпечення та технічного забезпечення НБК;

в) Будівля електротехнічних пристроїв – це будівля, в якій знаходяться системи життєзабезпечення НБК, включаючи в основному електрощитові, розподільний пристрій та трансформатори;

г) Шлюз доступу пожежних підрозділів – призначений для доступу пожежних автомобілів і персоналу;

Процес управління даними об'єктами відбувається з урахуванням наступних загальних функцій НБК та його основних елементів, якими є:

- обмеження впливу радіації на персонал, населення, навколишнє середовище;
- обмеження поширення радіоактивного забруднення за межі НБК;
- управління процесами НБК (включаючи поведження з РАВ);
- контроль стану конструкцій та елементів НБК протягом терміну експлуатації (100 років);
- фізичний захист і контроль доступу.

Сукупність основних технологічних і інженерних систем НБК включає в себе наступні системи:

- система контролю доступу – для контролю і управління доступом персоналу на об'єкти НБК;
- система електропостачання та електрообладнання призначена для забезпечення електроживленням всіх споживачів НБК при нормальних умовах експлуатації, порушеннях цих умов та аварійних ситуаціях;
- система протипожежного захисту – система, що створює умови, при яких виникнення пожежі не призводить до радіаційної аварії.
- система вентиляції та кондиціонування повітря призначена для

					<i>КНТЕУ 121 023-01.МР</i>	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		8

виконання наступних функцій: обмеження забруднення приміщень НБК, підтримання рівня викидів в атмосферу в межах, визначених відповідними нормами і вимогами забезпечення;

- система промислового телебачення призначена для візуального контролю за ходом протікання технологічного процесу і роботою механізмів;

- система поводження з рідкими радіоактивними відходами призначена для збору радіоактивно забруднених стоків води з робочих площадок/ділянок простору під Аркою і приміщень Технологічної будівлі;

- система основних кранів забезпечує демонтаж нестабільних конструкцій об'єкта «Укриття» і наступні роботи щодо поводження з паливовмісними матеріалами та РАВ всередині НБК.

- система радіаційного контролю (RS – ідентифікаційне позначення системи для проектування системи управління) являє собою сукупність засобів технічного, програмного, інформаційного, метрологічного та організаційного забезпечення для автоматизованого контролю іонізуючого випромінювання.

- інтегрована система управління (IS – ідентифікаційне позначення системи для проектування системи управління) призначена для контролю і управління технологічними процесами НБК [11].

1.2. Особливості комп'ютеризованих систем управління технічним обслуговуванням промислового комплексу

Технічне обслуговування (ТО) – комплекс методів та процедур для підтримки устаткування у справному стані чи працездатності в процесі експлуатації, зберіганні і транспортуванні [15].

Основними завданнями ТО є:

- підтримання працездатності обладнання;
- своєчасне виявлення і усунення дефектів;
- регулювання та налаштування вузлів і деталей;

					<i>КНТЕУ 121 023-01.МР</i>	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		9

– відновлення справності і часткове відновлення ресурсу.

Розрізняють основні типи ТО:

а) поточне технічне обслуговування – нерегламентований процес, що виконується технічним персоналом і включає щогодинну, позмінну перевірку та контроль роботи устаткування (діагностика, огляд, змащення вузлів). Даний тип ТО передбачає усунення виявлених дефектів без зупинки технологічного процесу

б) планове технічне обслуговування – регламентований процес, що виконується в обов'язковому порядку згідно затвердженого регламенту робіт (щорічний план-графік ТО, щомісячний план-графік ТО, журнали ТО та ремонту устаткування, карти огляду). Даний тип ТО передбачає зупинку технологічного процесу чи робочої одиниці устаткування відповідно до план-графіків та регламентованих методів виконання робіт з ТО [17].

CMMS (Computerized maintenance management system) – це програмне рішення, що охоплює інформацію щодо технічного обслуговування устаткування, підтримує процеси та автоматизує деякі завдання для підвищення ефективності виконання робіт. Програмне забезпечення комп'ютеризованої системи управління технічним обслуговуванням оптимізує використання, доступність та термін служби обладнання, машин, інфраструктури об'єктів чи операцій [14].

Основними функціями програмного забезпечення CMMS є:

- створення бази даних обладнання основних фондів;
- обробка даних про необхідні запчастини і ремонтний персонал;
- опрацювання заявок на закупівлю деталей;
- календарне планування технічного обслуговування;
- управління та зберігання інформації щодо витрати та дефектах устаткування на підприємстві;
- створення стандартних і повних звітів щодо ремонтів і ТО.

Важливими перевагами використання CMMS для промислових підприємств є:

					<i>КНТЕУ 121 023-01.МР</i>	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		10

– видимість активів, тобто централізована інформація в базі даних CMMS дозволяє інженерно-технічному персоналу відстежувати коли актив був придбаний, коли проводилося технічне обслуговування, частота дефектів, використані деталі, рейтинги ефективності;

– прозорість робочого процесу, тобто інтерфейс програмного забезпечення може бути налаштований для технічних фахівців для оцінки стану і прогресу практично у режимі реального часу. Фахівці з технічного обслуговування можуть швидко визначити де знаходиться актив, що йому потрібно, хто і коли повинен працювати з ним;

– автоматизація та оптимізація процесів технічного обслуговування активів: автоматизація ручних завдань, таких як замовлення запчастин, поповнення запасів, планування змін, збір інформації для метеорології, допомагає заощадити час, зменшити кількість помилок, підвищити продуктивність і зосередити фахівців на завданнях з технічного обслуговування активів;

– здоров'я, безпека та навколишнє середовище: відповідно до вимог управління відповідністю CMMS пропонують централізовану звітність з питань безпеки, здоров'я і навколишнього середовища. Метою є зниження ризику і підтримку безпечного виробничого середовища. За допомогою CMMS можуть проводитися розслідування для аналізу повторюваних інцидентів або дефектів, відстежуватися інциденти і коригувальні дії, а також керувати змінами процесів.

До початку використання CMMS систем на промислових підприємствах інформація щодо технічного обслуговування устаткування зберігалася у паперових файлах, а також була розкидана по електронних таблицях. На даний час такі типи систем отримують інформацію про стан активів автоматично від датчиків, що приєднані до устаткування, передають стан устаткування бездротовою мережею чи через дротове з'єднання [17].

Також, даний тип систем можливо інтегрувати та вже інтегровані в програмне забезпечення або програмні пакети, які використовуються у

					<i>КНТЕУ 121 023-01.МР</i>	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		11

промислових технологічних процесах, наприклад SCADA.

Розглянемо наступні CMMS системи, що мають широкий вжиток на ринку України та міжнародному ринку та пропонують різноманітний функціонал для управління та контролю за процесом ТО устаткування:

а) «Галактика» – програмний модуль для планування ресурсів підприємства, а також підготовки та створення необхідних документів, звітів, графіків, аналітичних даних, необхідних для оптимізації управління [10];

б) «Champs» – програмний продукт для середніх і великих промислових підприємств, що розгортається як локально так і в хмарному середовищі. Визначає та відстежує інформацію, що пов'язана з усім заводським устаткуванням, що вимагає обслуговування, супутніми видатками, історією виконання робіт [8];

в) «Fiix» – це хмарна комп'ютеризована система управління технічним обслуговуванням на основі штучного інтелекту. Fiix допомагає компаніям управляти всім своїм обслуговуванням, включаючи тисячі активів, замовлень та деталей, в одному місці [9].

1.3. Мета та задачі системи технічного обслуговування промислового комплексу

Згідно вищезазначеного аналізу сфери діяльності промислового комплексу, особливостей системи управління ТО та додатків-аналогів, насамперед необхідно визначити мету та задачі щодо проектованої системи.

Основними користувачами, при введенні в експлуатацію системи, буде інженерно-технічний персонал промислового комплексу НБК по обслуговуванню устаткування автоматизованих систем верхнього та середнього рівнів, а також інші підрозділи, що мають відношення до управління активами, обслуговування та ремонту устаткування систем промислового комплексу.

Мета системи – організація управління ТО устаткуванням верхнього та

					<i>КНТЕУ 121 023-01.МР</i>	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		12

середнього рівня автоматизованих систем промислового комплексу НБК, а також планування, управління активами, розподілення потужностей та звітності.

Основними задачами системи управління ТО устаткування промислового комплексу є:

а) забезпечити облік та контроль щодо виконання робіт з обслуговування устаткування згідно нормативної, експлуатаційної та регламентованої документації підприємства щодо промислового комплексу;

б) забезпечити фахівців та персонал, що залучений до експлуатації, налагодження та ремонтних робіт щодо устаткування промислового комплексу, оперативною та статистичною інформацією при прийнятті рішень у процесі виконання робіт;

в) генерувати інформацію щодо обліку та планування термінів виконання регулярного технічного обслуговування, система дозволяє організувати календарне планування планового ТО, що дозволяє підтримувати на достатньому рівні експлуатаційний стану устаткування;

г) надавати довідкову інформацію щодо основної інформації про устаткування промислового комплексу;

д) надавати інформацію про технічні та експлуатаційні показники устаткування;

е) формувати необхідну документовану та звітну інформацію для виконання ТО.

1.4. Висновки до розділу 1

Отже, враховуючи вищезазначене, можна визначити, що згідно проведеного аналізу предметної області, додатків аналогів та визначення мети та задач даний тип системи необхідно розробити та впроваджувати на промисловому комплексі НБК для мінімізації витрат, запобіганню виводу з експлуатації устаткування та введення аварійного режиму при технологічному процесі.

					<i>КНТЕУ 121 023-01.МР</i>	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		13

РОЗДІЛ 2

ПРОЕКТУВАННЯ АРХІТЕКТУРИ ТА СТРУКТУРИ ДАНИХ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1. Системні вимоги щодо системи управління технічним обслуговуванням

На початку проектування програмного забезпечення системи управління технічним обслуговуванням головним завданням є пред'явлення системних вимог.

Системні вимоги до розробки програмного забезпечення:

- а) інтуїтивно зрозумілий інтерфейс;
- б) операційна система Windows XP/7/8/10;
- в) використання протоколу TCP/IP для передачі даних між сервером та клієнтом;
- г) серверна частина, що використовує SQL Server Management 2012, повинна зберігати базу даних системи. Основними особливостями середовища є:
 - 1) інтерфейс для користувача та група інструментів з розширеним редактором сценаріїв [1];
 - 2) Transact-SQL – мова компоненту Database Engine, до складу якої входять: константи, ідентифікатори, роздільники, логічні функції, типи даних [2, с. 109];
 - 3) продуктивність та масштабованість – властивості, що складаються з непрямих контрольних точок, що видаються в фоновому режимі для задоволення цільового часу заданого користувачем та відновлення для обраної бази даних значної кількості табличних розділів [4];
 - 4) безпека, що забезпечує перевірку бази даних та ролі серверів, що визначаються користувачем [3];

Зм.	Аркуш	№ докум.	Підпис	Дата	<i>КНТЕУ 121 02з-01.МР</i>			
Зав. каф.		Криворучко О.В.		24.05.21	Проектування програмного забезпечення системи управління технічним обслуговуванням промислового комплексу	Стадія	Аркуш	Аркушів
Керівник		Рзасва С.Л.		24.05.21		P2	14	40
Гарант		Токар В.В.		24.05.21		Факультет інформаційних технологій 2 курс, 2з група		
Розробив		Баллон М.В.		24.05.21				

5) повнотекстовий пошук – компонент, що дозволяє здійснювати пошук через дані, що зберігаються в текстових документах (алфавітно-цифрових стовпцях реляційної таблиці) і дозволяє користувачу та додаткам виконувати повнотекстові запити [2, с. 816].

д) клієнтська частина повинна бути розроблена у середовищі Microsoft Visual Studio 2019 на мові програмування C#, що дозволяє розробнику створювати велику кількість типів безпечних та надійних програм, які працюють у екосистемі .NET. [7].

е) використання інструменту Crystal Reports .NET для Visual Studio, за допомогою якого будуть створені форми документів звітності. Інструмент доступний з Crystal Reports .NET дає розробнику швидший та продуктивніший спосіб реалізації звітів, які відповідають потребам користувача [6]. Пакет Crystal Reports дозволяє створювати звіти, здатні здійснювати вибірку та форматування результуючого набору з бази даних або інших джерел даних Крім простого зчитування даних з джерел даних, інструмент дозволяє виконувати певні обчислення [5]. Для цієї мети в ньому є власна мова формул, а також цілий ряд програмних додатків, які можуть бути використані для перетворення необробленого матеріалу у високоякісні представницькі звіти, що містять графіки, діаграми, проміжні суми [4].

У процесі проектування програмного забезпечення визначено моделі архітектури системи управління технічним обслуговуванням устаткування, що представлені у вигляді діаграми розгортання, діаграми класів, діаграма варіантів використання та діаграми діяльності.

Розробка моделі будь-якої системи завжди передуює її створенню, це дозволяє переконатися в архітектурній узгодженості проекту до того, як він буде реалізований в коді. Для полегшення проектування використано уніфіковану мову моделювання (Unified Modeling Language, UML), яка є графічною мовою для проектування, специфікації, візуалізації, та документування [13].

					<i>КНТЕУ 121 023-01.МР</i>	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		15

2.2. Архітектура системи управління технічним обслуговуванням

2.2.1. Модель інформаційної системи в розрізі програмного рівня

Розглянемо розроблену діаграму розгортання, що описує організаційну сторону та підключення фізичних компонентів у проектованій системі. Вона використовується для деталізації моделі реалізації та перевірки – чи всі аспекти функцій системи охоплює планова розробка. Зазначена діаграма показує системні вимоги до розробки [4].

Крім того, діаграма відображає, конфігурацію вузлів, що обробляються (на яких виконується система) і компоненти, що розміщені в цих вузлах. Діаграма розгортання системи управління технічним обслуговуванням зображена на Рис 2.1.

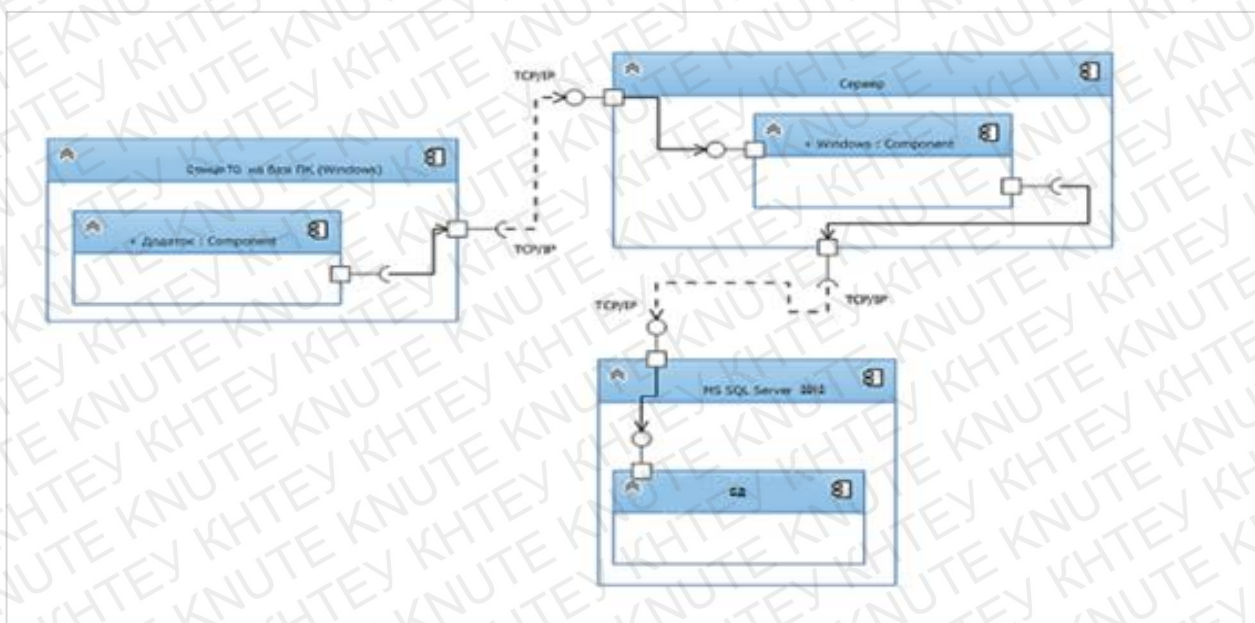


Рис. 2.1. Діаграма розгортання системи управління технічним обслуговуванням

До складу компонентів діаграми входять:

1. Станція технічного обслуговування на базі операційної системи Windows 7 з розробленим додатком (клієнтом) системи управління технічним обслуговуванням устаткування;

					КНТЕУ 121 023-01.МР	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		16

2. Сервер з операційною системою Windows Server 2012 або вище версії;
3. MS SQL Server 2012 є основною системою управління базою даних, що встановленою на сервер, для управління та організації спроектованої бази даних;
4. CMMS – база даних системи управління технічним обслуговуванням устаткування;
5. Зв'язок між компонентами архітектури надається по протоколу TCP/IP.

2.2.2. Модель класів

Діаграма класів програмного забезпечення системи технічного обслуговування зображено на Рис. 2.2.

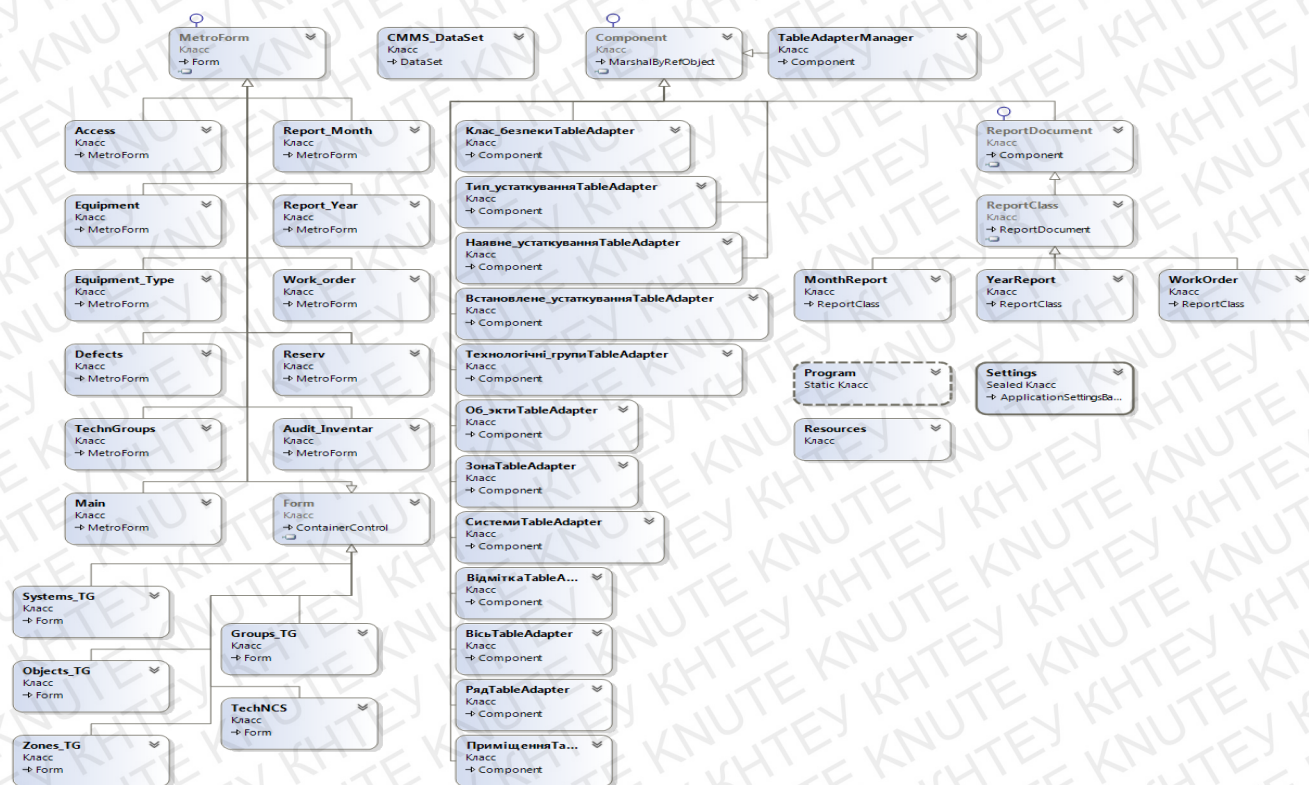


Рис. 2.2. Діаграма класів системи управління технічним обслуговуванням

До структури діаграми класів входять:

1. Клас таблиці – екземпляри таблиць в яких знаходяться дані через які реалізується взаємодія (обмін) даними між сервером та програмним

забезпеченням. Даний тип таблиць не реалізований за допомогою запитів, тому необхідно додати дані у TableAdapter і викликати метод UpdateAll у класі TableAdapterManager:

- CMMS_DataSet;
- TableAdapterManager;
- ЗонаTableAdapter;
- СистемиTableAdapter;
- Об'єктиTableAdapter;
- Технологічні_групиTableAdapter;
- Тип_устаткуванняTableAdapter;
- Наявне_устаткуванняTableAdapter;
- Встановлене_устаткуванняTableAdapter;
- Клас_безпеки_системиTableAdapter;
- ПриміщенняTableAdapter;
- ВісьTableAdapter, РядTableAdapter, ВідміткаTableAdapter;

2. Службові класи – класи, що керують додатком, реалізуються за замовченням. До складу класів входять налаштування (Settings), ресурси (Resources) та програма (Program). Вони розмішують в собі точку входу та реалізують запуск додатку з створення та відображення інтерфейсу першої форми (Access.cs).

3. Класи форми, що складаються з 15 форм з обробниками подій:

- MetroForm – є базовим класом, що встановлений за допомогою диспетчера пакетів для реалізації дизайну інтерфейсу додатку та елементів управління Metro Framework;
- Access – клас, що розроблений для аутентифікації щодо відповідності пароля введеного користувачем на предмет збігів паролю та логіну, що зберігається в базі даних, а також визначення ролі основної користувачів;

					<i>КНТЕУ 121 023-01.МР</i>	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		18

- Main – клас для управління функціоналом системи (даними та інформацією);
- TechnGroups – клас для формування і відображення форми технологічних груп системи, та відповідальний за додавання, редагування та видалення інформації при виконання дій у технологічних групах системи;
- Form – базовий клас до якого входять групи класів – TechNCS, Zones, Systems, Groups, структуровані за допомогою об'єкту treeNode і реалізується у TechnGroups;
- TechNCS, Zones, Systems, Groups – організовані для виконання видалення, редагування та додавання даних при виконанні дії з об'єктами (зоною, група, система, об'єкт). У TechNC, Zones, Systems, Groups наявні запити до бази даних (Insert, Update, Select, Delete), що відносяться до конкретних об'єктів;
- Equipment_Type – відповідає за віднесення устаткування до певного типу. У Equipment_Type реалізований запит до бази даних «Insert» та взаємодія з формами Equipmen, Installed_Equipment;
- Equipmen, Installed_Equipment – організовані для управління та організації форм наявного та встановленого устаткування;
- Reserv – відповідальний за формування та додавання устаткування до резервного. У Reserv наявний запит «Insert» та «Select» для отримання інформації щодо конкретного об'єкту;
- Defects – відповідальний за обробку події об'єкту – занесення дефектного устаткування до дефектів. У Defects реалізовані запити до бази даних (Insert, Select);
- Audit_Inventar – відповідальний за обробку події – занесення резервного устаткування до встановленого. У Audit_Inventar розроблені запити до бази даних (Insert, Select).

					<i>КНТЕУ 121 023-01.МР</i>	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		19

4. Спеціальні класи – класи, що відповідають за реалізацію та обробку даних для процедури щодо звітності:

- WorkOrder – забезпечує створення наряду для виконання робіт з технічного обслуговування. У класі Work_Order реалізований запит до бази даних для наповнення інформацією для створення шаблону наряду;
- ReportMonth – забезпечує створення звіту на необхідний місяць за запитом користувача системи. У класі ReportMonth наявний запит до бази даних для планування місячного звіту;
- ReportYear – забезпечує створення звіту за рік за запитом користувача системи. У класі ReportYear наявний запит до бази даних для планування звіту.

2.2.3. Модель інформаційної системи в розрізі користувацького рівня

Діаграма варіантів використання є цінною для візуалізації функціональних вимог системи. Дана діаграма є графом особливого виду, який вважається графічною нотацією для зображення визначених варіантів використання, акторів, деяких інтерфейсів і відносин між цими складовими [16]. Окремі компоненти діаграми укладені в прямокутник, який позначає проєктовану систему. Необхідно підкреслити, що між акторами і варіантами використання можуть бути тільки деякі фіксовані типи взаємозв'язків, які в сукупності описують функціональні вимоги до модельованої системи. Діаграма допомагає визначити будь-які внутрішні і зовнішні фактори, що мають вплинути на систему та повинні бути обов'язково враховані [4]. Діаграма варіантів використання системи управління технічним обслуговуванням зображена на Рис.2.3.

Одразу кілька користувачів можуть працювати з базою даних у додатку. Роль користувача визначається після процесу авторизації, у разі якщо «Керівництво», він має право переглядати інформацію щодо наявного та

					<i>КНТЕУ 121 023-01.МР</i>	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		20

встановленого устаткування, перегляду функціональних груп системи, типів устаткування, додавання устаткування до резерву, організовує роботу з звітом за обраний місяць. Можливості адміністратора з управління системою: додавання, редагування, видалення та роботи із документами виділяється ролі «Провідний інженер» [4].

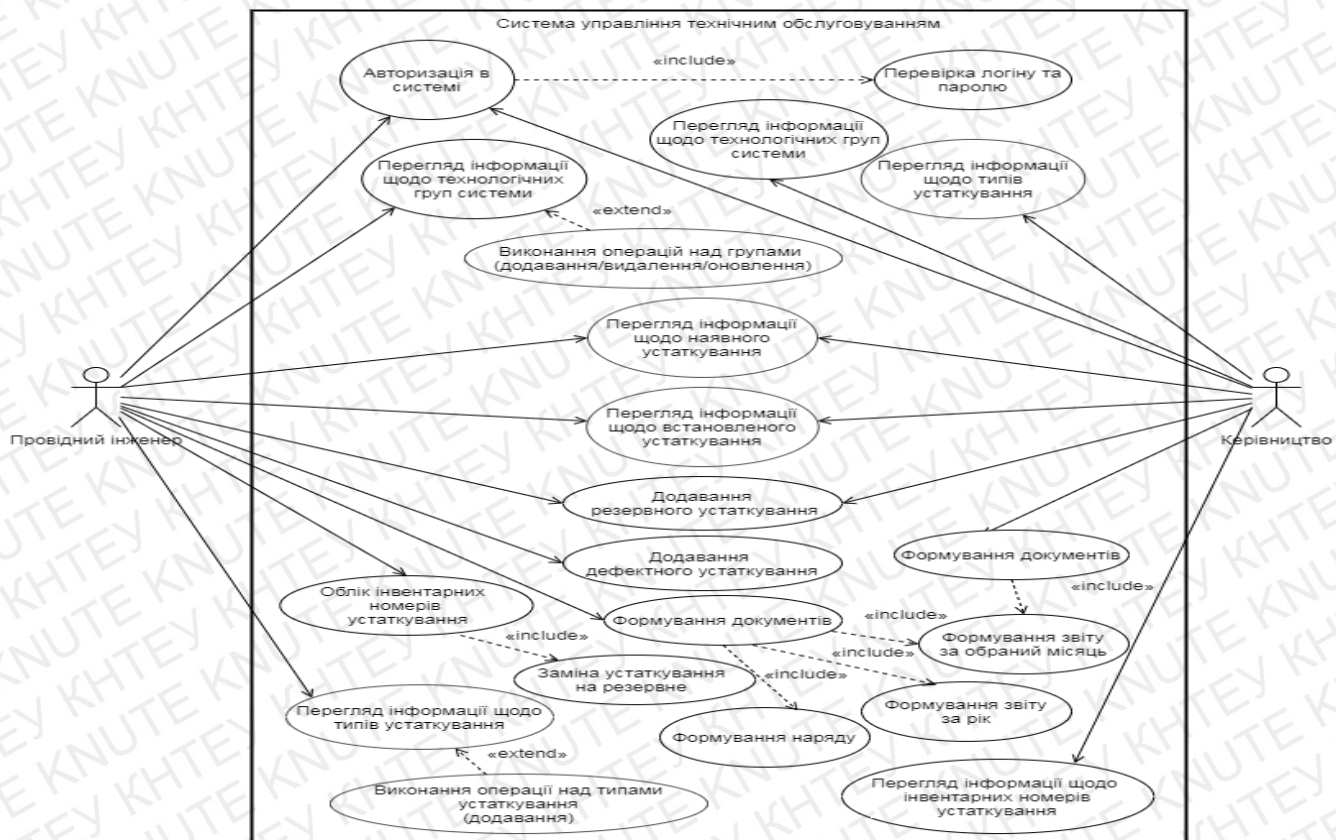


Рис. 2.3. Діаграма варіантів використання системи управління технічним обслуговуванням

2.2.4. Модель компонентів управління інформаційної системи

Завершальним етапом для проектування даної системи є діаграма діяльності, що має на меті ознайомити з прикладом організації робочого процесу системи. Одним з корисних застосувань діаграм діяльності – визначення алгоритму. Алгоритм, по суті, являє собою декомпозицію функцій, що включені в даний процес. На діаграмі алгоритму представлені переходи потоку управління

від однієї діяльності до іншої [4]. Діаграма діяльності системи технічного обслуговування промислового комплексу зображена на Рис. 2.4.

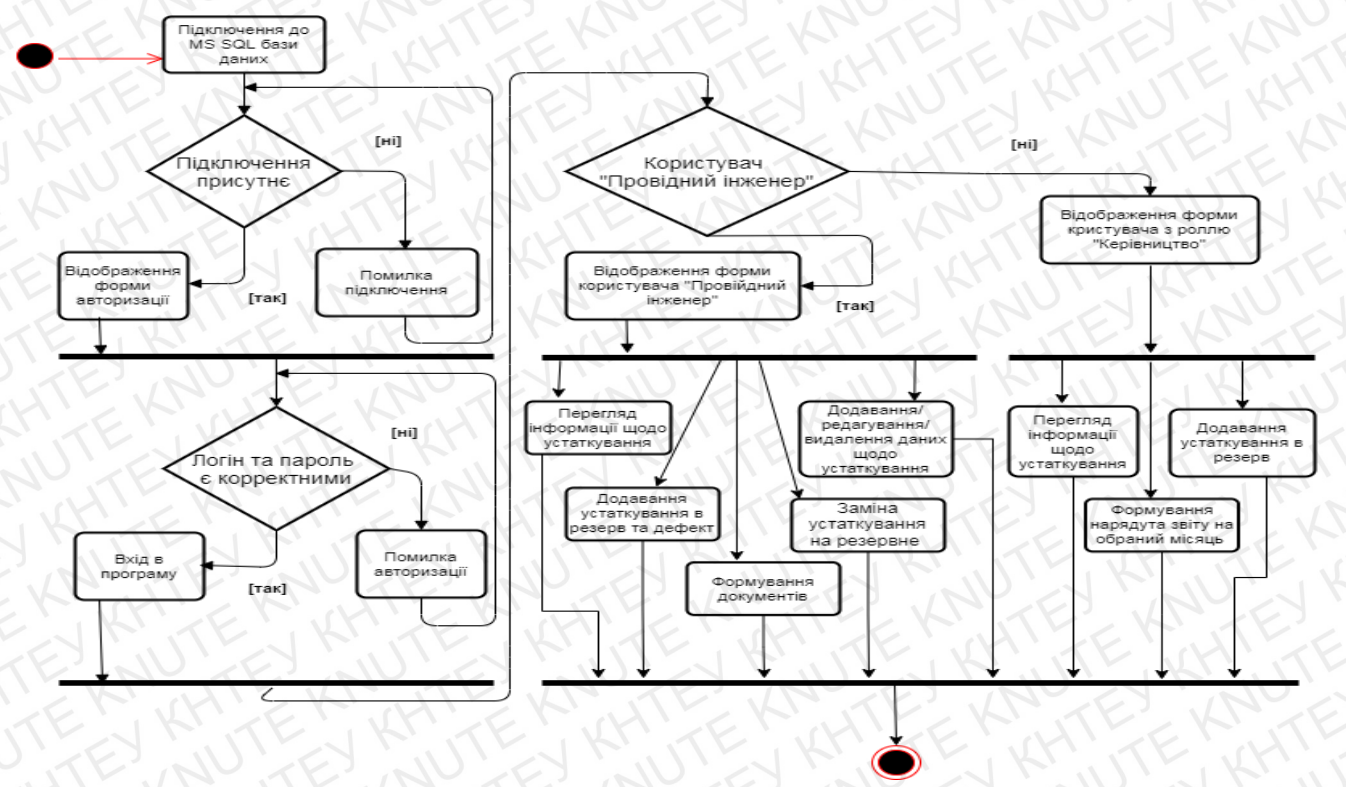


Рис. 2.4. Діаграма діяльності системи управління технічним обслуговуванням

Після операції запуску програми, додаток проводить процес підключення до бази даних у локальній мережі. У разі відсутності зв'язку з сервером видається повідомлення щодо неможливості підключення до бази даних і відбувається автоматичний вихід з програми. У випадку вдалого підключення, програма виводить першу форму – форма авторизації [4].

При виході користувача з системи програма завершить свою роботу, в протилежному випадку користувач може продовжити працювати з нею. В кожному такому випадку створюється і відображається нова форма.

Після дійсного входу користувача у програму відбувається визначення його ролі, в залежності від цього буде відображена відповідна форма. Користувач продовжує виконувати свою роботу з системою управління технічного обслуговування до того часу, поки не завершить програму [4].

2.3. Проектування структурної моделі бази даних системи управління технічним обслуговуванням

Структурна модель бази даних системи технічного обслуговування промислового комплексу НБК включає в себе 22 таблиці та охоплює процес роботи з устаткуванням та ресурсами. Схема бази даних mdf.CMMS системи представлена на Рис. 2.5.

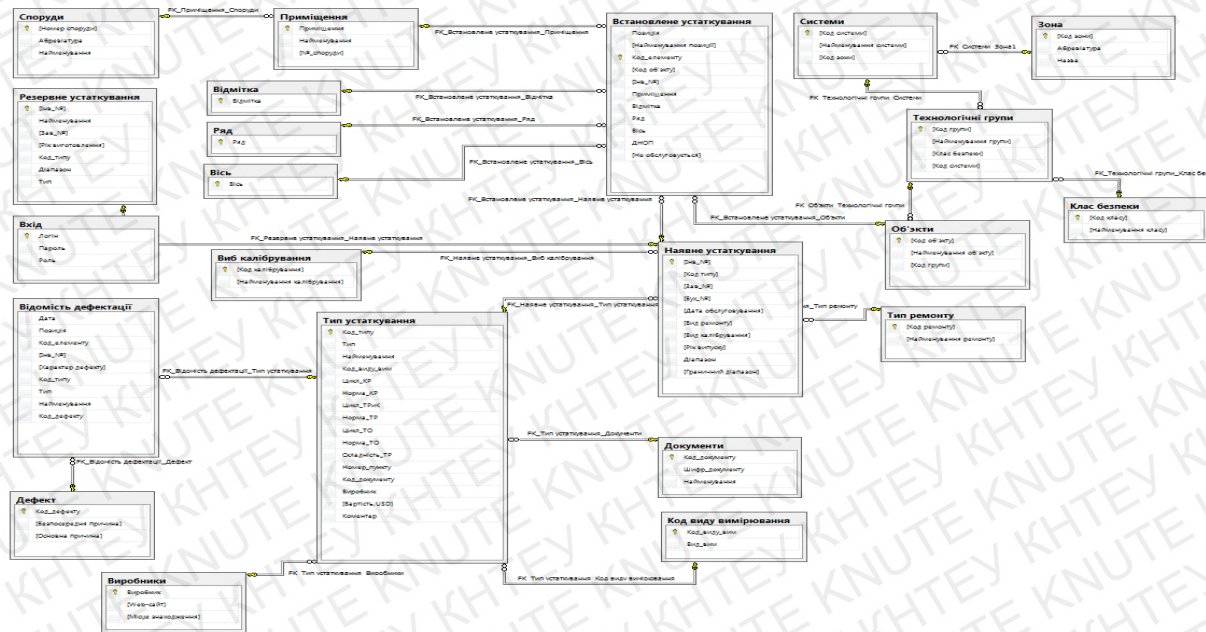


Рис.2.5. Структурна схема бази даних системи управління технічного обслуговування

Таблиця «Вхід» – довідник, що передбачає наявність інформації про користувачів системи та їх ролей. Для роботи та відображення інформації щодо технологічних груп промислового комплексу, в яких розташовано устаткування для проведення ТО, спроектовані таблиці «Зони», «Системи», «Об’єкти», «Технологічні групи» та «Клас безпеки», з сутність зв’язками – «один до багатьох».

У ході реалізації програмного забезпечення буде використано операцію пошуку за запитами та за допомогою фільтра інформації, що дозволить скоротити

час для отримання необхідного устаткування. Для виконання операції резервування та вилучення устаткування спроектована таблиця «Резервне устаткування», що наповнена інформацією щодо устаткування, рік його виготовлення та найменування з сутність зв'язком – «один до багатьох». У процесі виводу устаткування в ремонт та з експлуатації повинно вилучатися з наявного устаткування. Таким чином для виконання цього запиту спроектовано таблиці «Відомість дефектації» та «Дефект», дефектному устаткуванню різного типу присвоюється код – сутність зв'язок «один до багатьох».

Для налагодження процесу роботи з документами інженерно-технічного персоналу промислового комплексу НБК розроблено запити представлення до бази даних: «Журнал_ТО_Місяць», «Річний_графік_ТО», «Наряд_на_виконання_робіт», для того щоб відображати у формі місячного, річного звітів та наряду для виконання робіт з технічного обслуговування устаткування. Запити реалізації даних типів документів наведено у Додатку М.

2.4. Висновки до розділу 2

Отже, враховуючи вищезазначене, для початку стадії реалізації проекту було виконано:

- аналіз та виконання системних вимог згідно технічного завдання;
- діаграму класів, в якій наведена структура основних класів системи;
- структурну модель даних системи, на основі якої база даних виконує процес обробки даних;
- діаграму розгортання, за допомогою якої вибудовані компоненти, що входять до складу системи;
- діаграму варіантів використання, за допомогою якої спроектовані операції користувачів системи та розмежовані ролі;
- діаграму діяльності, в якій наведено алгоритм виконання дій користувачів та системи в цілому від початку запуску до завершення програми.

					<i>КНТЕУ 121 023-01.МР</i>	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		24

РОЗДІЛ 3

РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1. Функціональні вимоги до системи управління технічним обслуговуванням

На етапі реалізації програмного забезпечення системи управління технічним обслуговуванням необхідно пред'явити функціональні вимоги до розробки.

Функціональні вимоги до розробки програмного забезпечення:

- а) надання доступу авторизованим користувачам;
- б) розробка ролей користувачів: «Провідний інженер», «Керівництво»;
- в) обов'язкова наявність, присутність і можливість пошуку устаткування по найменуванню, інвентарному номеру та періодів технічного обслуговування;
- г) наявна можливість редагування, видалення та додавання події, що пов'язана з технічним обслуговуванням устаткуванням.
- д) реалізація документів – звіт за місяць, звіт за рік, наряд на виконання робіт з технічного обслуговування.

3.1.1. Авторизація користувача

Для процесу авторизація запрограмована форма Access.cs (Додаток А) для вікна програми «Вхід в систему», що зображено на Рис.3.1., містить в собі поля логіну та паролю користувача. Основними користувачами системи є «Провідний інженер» з логіном «Administrator», а також «Керівництво» з логіном «User». У графічному інтерфейсі запрограмовані кнопки для входу в систему і виходу з неї. Вони реалізовані для підключення до бази даних та порівняння між введеними значеннями у формі програми CMMS і наявними даними в таблиці dbo. Вхід бази даних mdf.CMMS.

Зм.	Аркуш	№ докум.	Підпис	Дата	<i>КНТЕУ 121 02з-01.МР</i>			
Зав. каф.		Криворучко О.В.		21.06.21	Проектування програмного забезпечення системи управління технічним обслуговуванням промислового комплексу	Стадія	Аркуш	Аркушів
Керівник		Рзасва С.Л.		21.06.21		РЗ	25	40
Гарант		Токар В.В.		21.06.21		Факультет інформаційних технологій 2 курс, 2з група		
Розробив		Баллон М.В.		21.06.21				
					Реалізація програмного забезпечення			

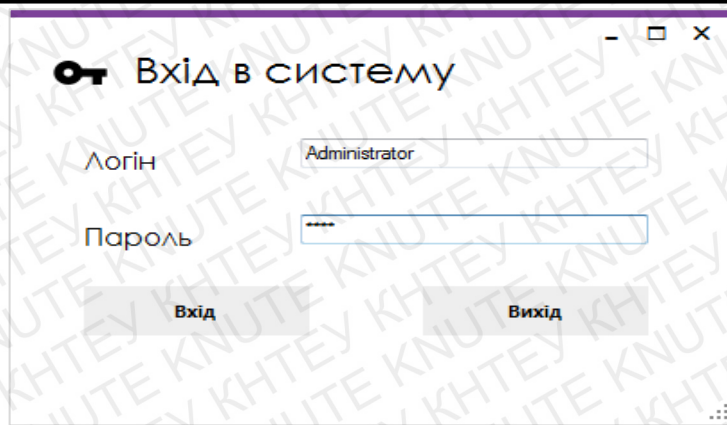


Рис. 3.1. Вікно програми авторизації користувача

При наявності помилки у процесі авторизації передбачений виклик діалогових вікон. Діалогове вікно помилки «Не вірний логін чи пароль» зображене на Рис. 3.2. У разі відсутності з'єднання з базою даних, неодмінно виникне помилка «Неможливо підключитися до бази даних. Будь-ласка, зверніться до системного програміста!».

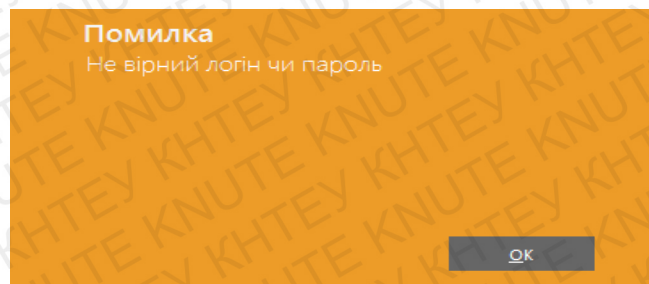


Рис. 3.2. Діалогове вікно програми при невірній авторизації користувача

3.1.2. Функціональні можливості та компоненти системи

Вікно програми «СММС-НБК», що зображене на Рис. 3.3, створене за допомогою головної форми Main.cs (Додаток Б) та необхідне для отримання інформації щодо функціональних можливостей та компонентів системи управління технічним обслуговуванням. В правому кутку вікна програми наявна інформація щодо входу авторизованого користувача з роллю – «Провідний інженер», а у випадку входу іншого користувача – «Керівництво».

					<i>КНТЕУ 121 023-01.МР</i>	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		26

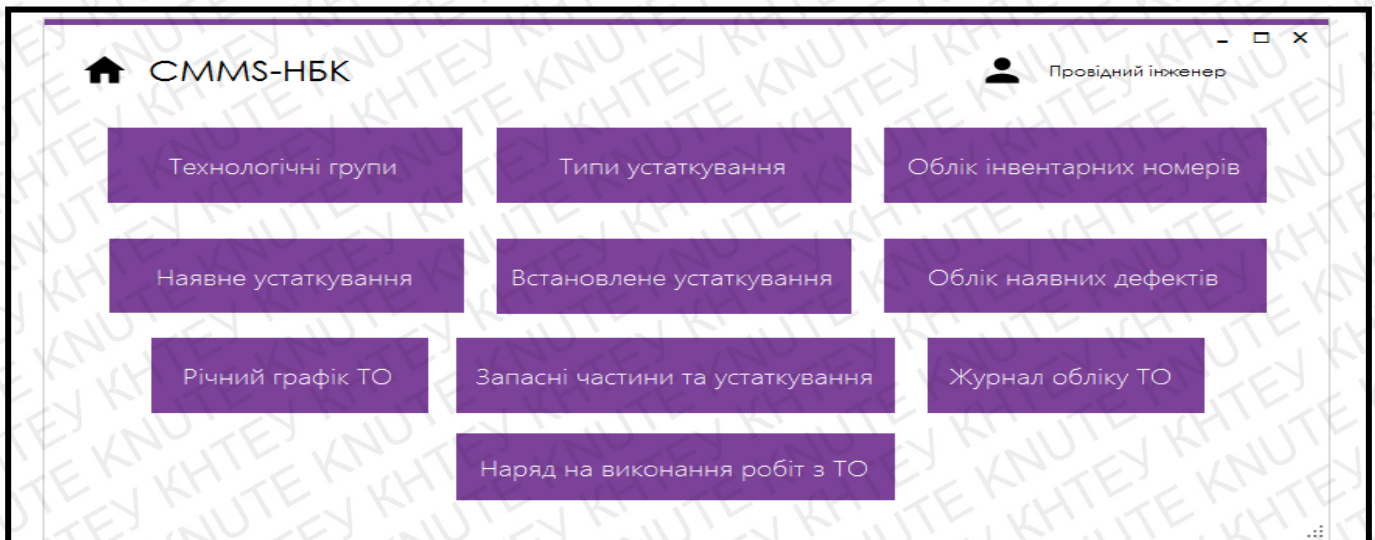


Рис. 3.3. Вікно програми «CMMS-НБК»

Вікно «Технологічні групи», що зображено на Рис. 3.4, розроблене за допомогою форми TechNCS.cs (Додаток В), на якому є наявний перегляд інформації щодо структури промислового комплексу НБК, в якому розміщується технологічне устаткування – зони, системи, технологічні групи та об’єкти. До функціоналу додана кнопка оновлення інформації у випадку додавання, редагування чи видалення. Для користувача з роллю «Керівництво» функція додавання, редагування та видалення інформації не підтримується.

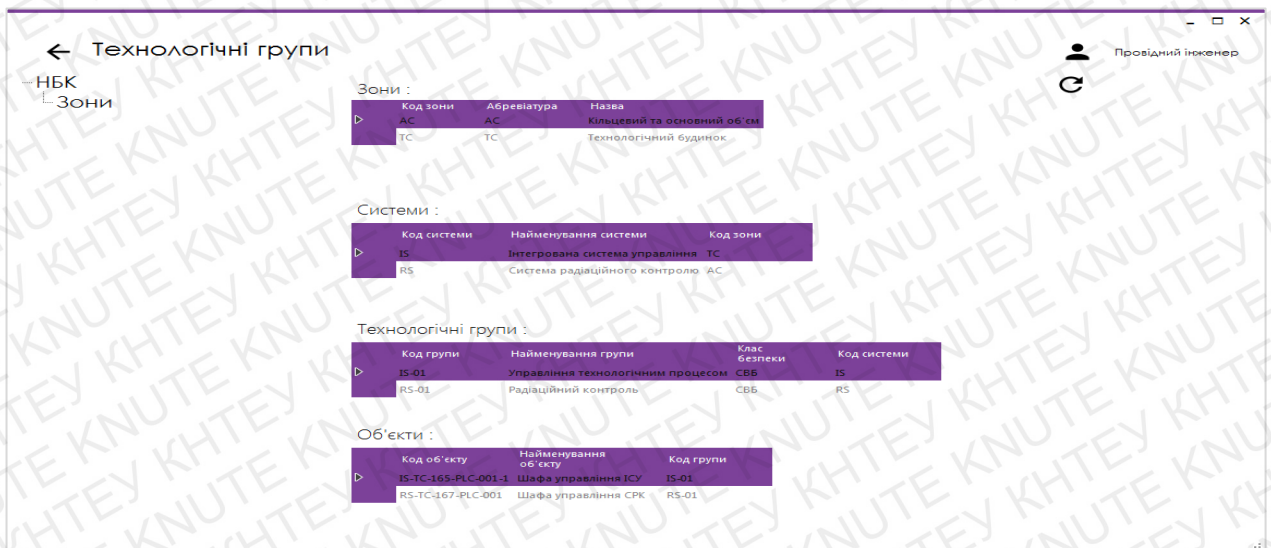


Рис. 3.4. Вікно програми «Технологічні групи»

У формі TechnGroups. Cs (Додаток Г) вікна програми «Технологічні групи» за допомогою методу treeView_AfterSelect організовано дерево «НБК», в якому розміщені форми Zones_TG. cs, Systems_TG. cs, Groups_TG. cs, Objects_TG. cs. В даних формах здійснено операції додавання, редагування та видалення зон, систем, технологічних груп та об'єктів. До кожної операції з'являється діалогове вікно, що забезпечує отримання інформації користувачу щодо його дій над операціями. Відповідні операції зображені на Рис.3.5, Рис.3.6., Рис. 3.7, Рис.3.8.

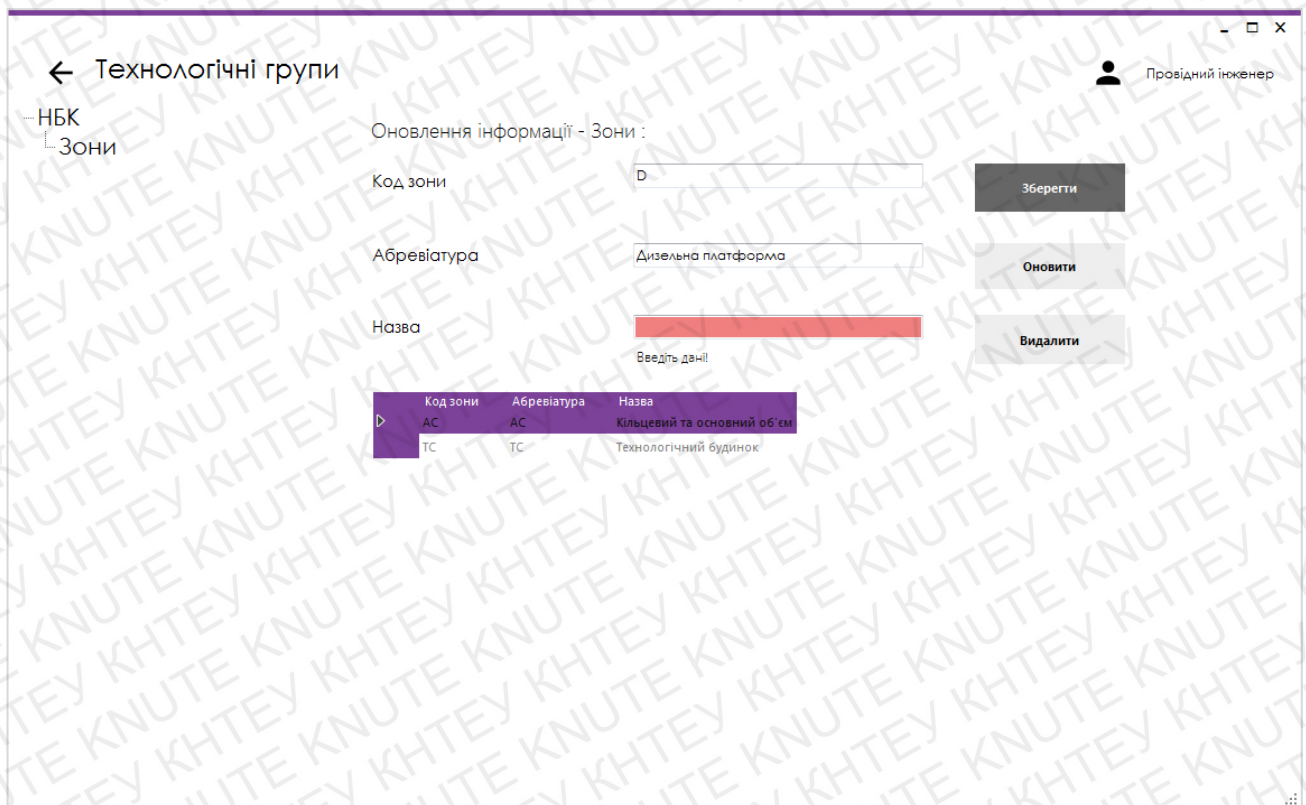


Рис. 3.5. Вікно програми «Технологічні групи – Оновлення інформації – Зони»

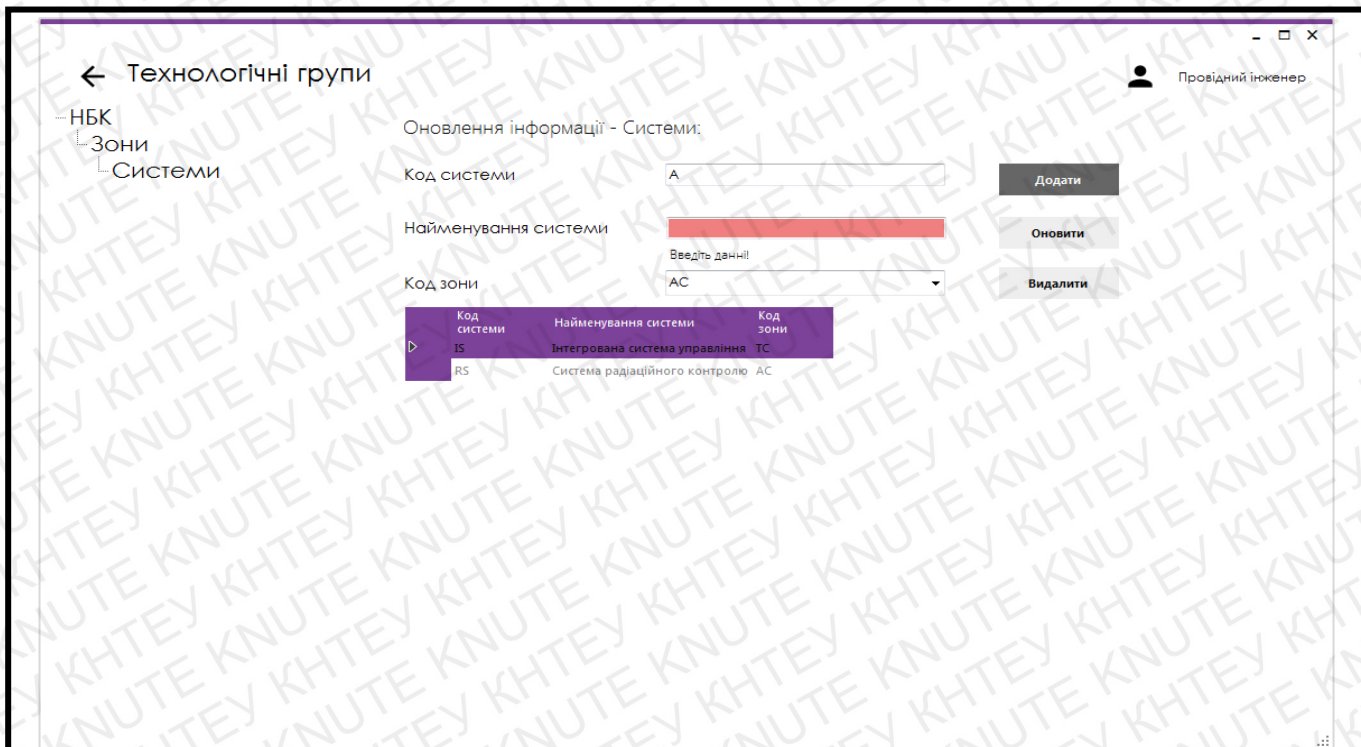


Рис. 3.6. Вікно програми «Технологічні групи – Оновлення інформації – Системи»

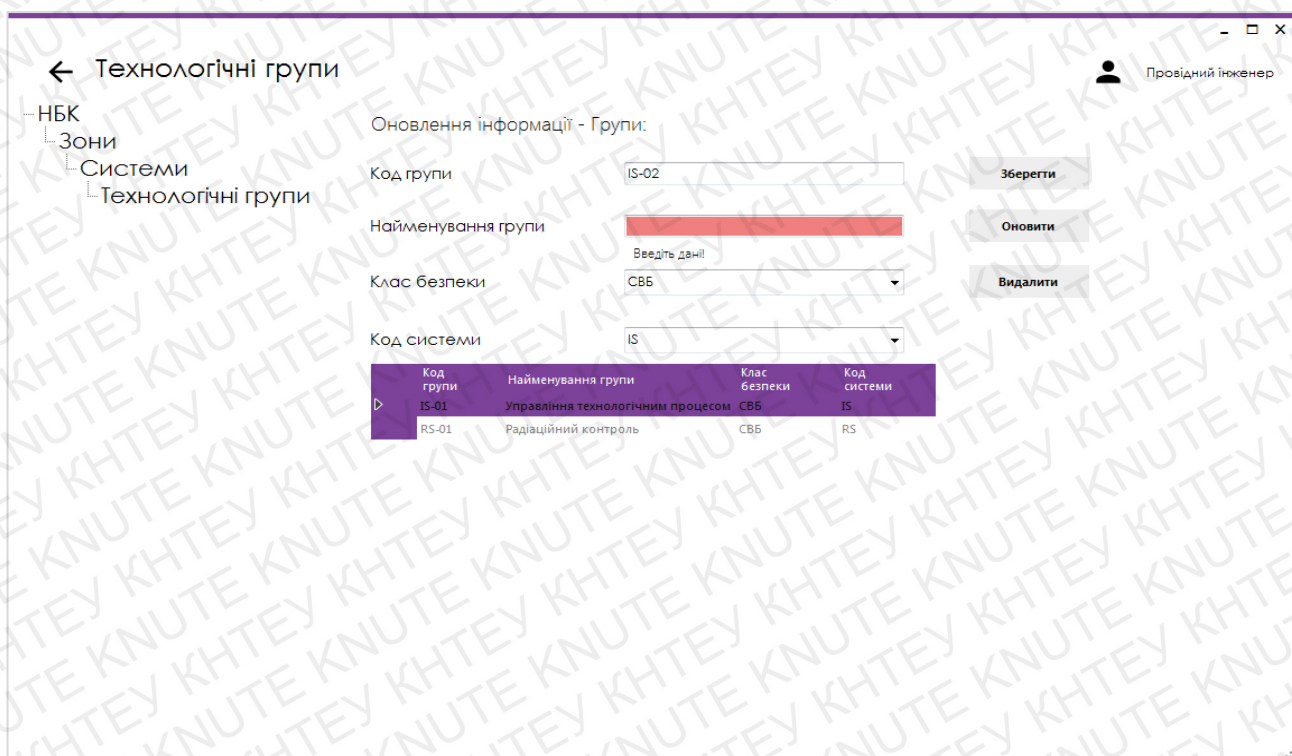


Рис. 3.7. Вікно програми «Технологічні групи – Оновлення інформації – Групи»

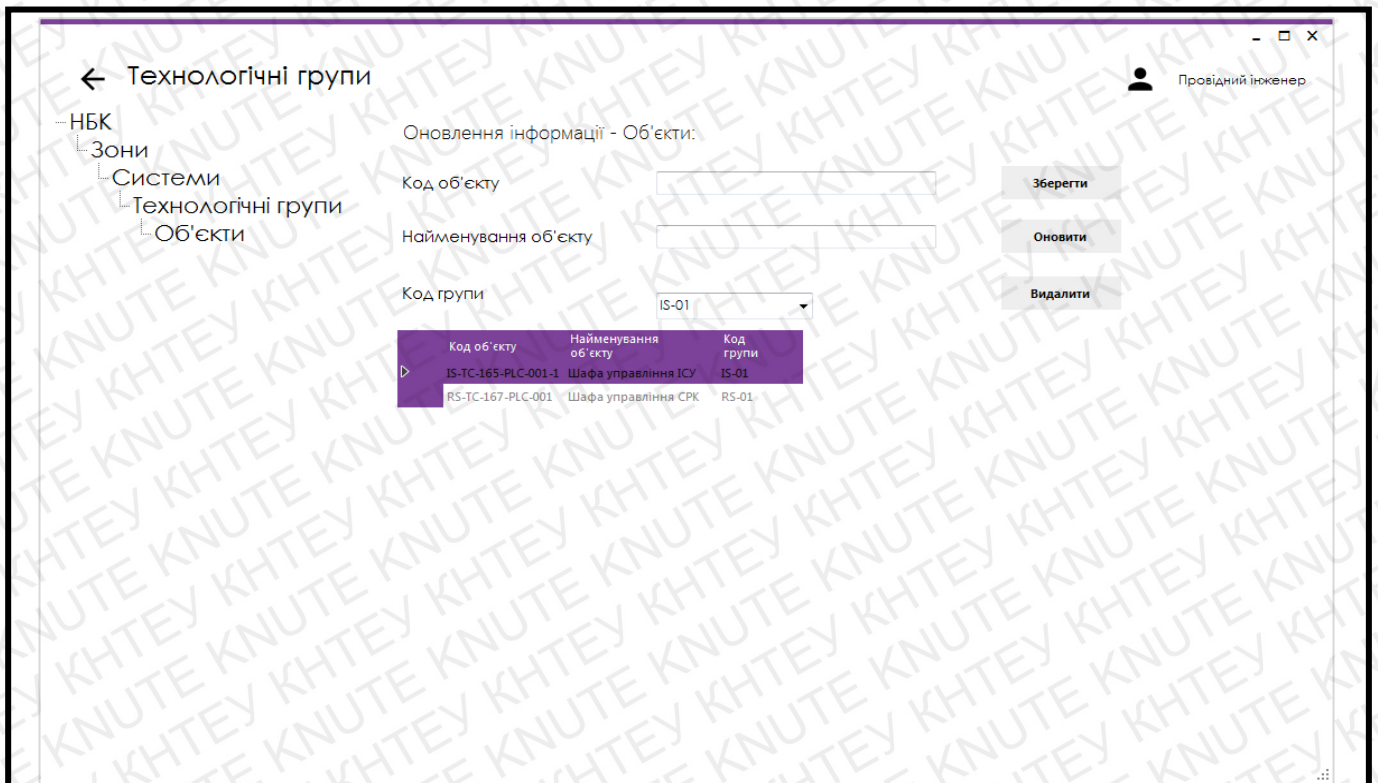


Рис. 3.8. Вікно програми «Технологічні групи – Оновлення інформації – Об’єкти»

Діалогові вікна для виконання операцій додавання, редагування та видалення зображені на Рис.3.9.

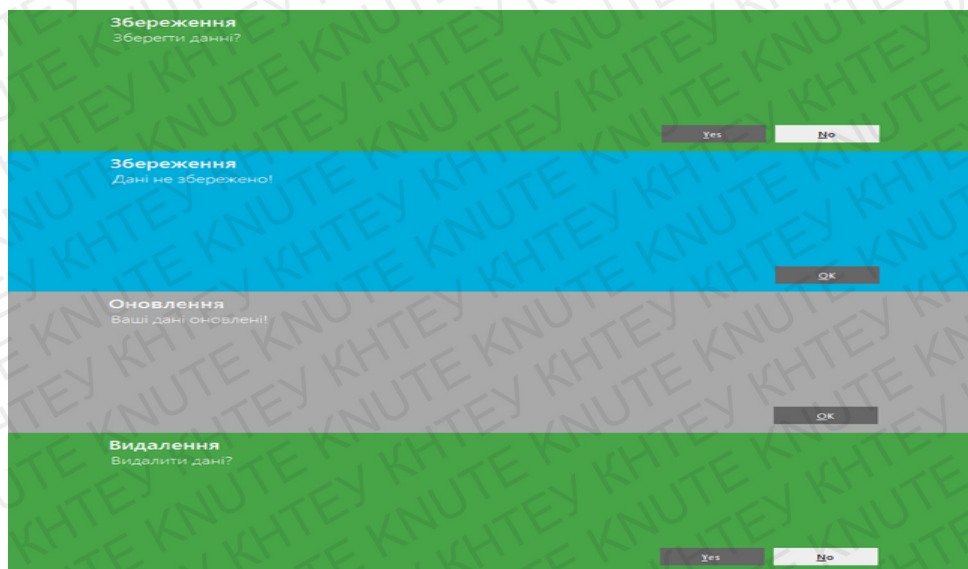


Рис.3.9. Діалогові вікна до операцій додавання редагування та видалення інформації

						Аркуш
						30
Зм.	Аркуш	№ докум.	Підпис	Дата	<i>КНТЕУ 121 023-01.МР</i>	

Виконання форми Equipment_Type.cs (Додаток Д) вікна програми «Тип устаткування», що зображене на Рис. 3.10, дає можливість додавати нове устаткування до таблиць бази даних dbo.Встановлене устаткування та dbo.Наявне устаткування, за типами, що організовано за допомогою елементів інтерфейсу – колекції вкладок: ЗВТ, АтаТЗ, ТЗВТ. Кнопки «Наявне устаткування» та «Встановлене устаткування» дають можливість відкрити дані форми для перегляду інформації. Для користувача з роллю «Керівництво» функція додавання нового устаткування не підтримується.

Код типу	Тип	Найменування	Код_виду_вим	Цикл_КР	Норма_КР	Цикл_ТРИК	Норма_ТР	Цикл_ТО	Норма_ТО	Склад
SAA	SK 3118.000	Регулятор внутрішньої температури шафи для Rittal TS8	EM	0	13	0	4,7	6	0,95	1
SAB	SK 3149.007	Монітор швидкості обертання	EM	0	13	0	4,7	6	1,6	1

Додавання до групи наявного устаткування:

Інв №

Код типу

Зав №

Бухгалтерський номер

Дата обслуговування

Вид калібрування

Рік випуску

Діапазон

Граничний діапазон

Рис. 3.10. Вікно програми «Типи устаткування»

За допомогою форми Audit_Inventar.cs (Додаток Е) створено вікно програми «Облік інвентарних номерів», що зображене на Рис. 3.11. На даному графічному інтерфейсі наявна інформація щодо вільних інвентарних номерів устаткування, що занесене до таблиці бази даних dbo.Резервне устаткування та елементу інтерфейсу – кнопки для додавання устаткування до таблиці

dbo.Встановлене устаткування. Такі елементи як текстові поля є неактивними для введення чи зміни даних. Для користувача з роллю «Керівництво» функція встановлення устаткування не підтримується.

Інв_№	Найменування	Зав_№	Рік виготовлення	Код типу	Діапазон	Тип
3022	Модуль аналогових вхідних сигналів SM 331 (AIB / 16bit)	2365236	2019	TAE	5-6	SIE.6ES7195331-7NF10-0AB0

Інвентарний №	<input type="text" value="3022"/>	Код типу	<input type="text" value="TAE"/>
Найменування	<input type="text" value="Модуль аналогових вхідних сигналів"/>	Діапазон	<input type="text" value="5-6"/>
Заводський №	<input type="text" value="2365236"/>	Тип	<input type="text" value="SIE.6ES7195331-7NF10-0AB0"/>
Рік виробництва	<input type="text" value="2019"/>	<input type="button" value="Встановити"/>	

Рис. 3.11. Вікно програми «Облік інвентарних номерів»

Вікно «Встановлене устаткування», що зроблено за допомогою форми Installed_Equipment.cs (Додаток Ж) та «Наявне устаткування» формою Equipment.cs (Додаток К), що зображені на Рис. 3.12, 3.13, надають інформацію згідно запиту у текстовому полі пошуку, що виконується згідно обраного параметра у елементі інтерфейсу випадаючого списку. Інформація надається з таблиць бази даних dbo.Встановлене устаткування та dbo.Наявне устаткування.

Позиція	Найменування позиції	Код_елементу	Код об'єкту	Інв_№	Приміщення	Відмітка	Ряд	Вісь	ДНОП	Не обслуговується
25A2	Інтерфейсний модуль	30160	IS-TC-165-PLC-001-1	3010	165	+0,8	A	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
25A2	Інтерфейсний модуль	30170	RS-TC-167-PLC-001	3011	167	-1,4	A	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
25A3	Модуль дискретних вхідних сигналів	30180	IS-TC-165-PLC-001-1	3012	165	+0,8	A	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
25A3	Модуль дискретних вхідних сигналів	30190	RS-TC-167-PLC-001	3013	167	-1,4	A	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
25A5.1	Модуль дискретних вхідних сигналів	30200	IS-TC-165-PLC-001-1	3014	165	+0,8	A	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
25A5.1	Модуль дискретних вхідних сигналів	30210	RS-TC-167-PLC-001	3015	167	-1,4	A	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Рис. 3.12. Вікно програми «Встановлене устаткування»

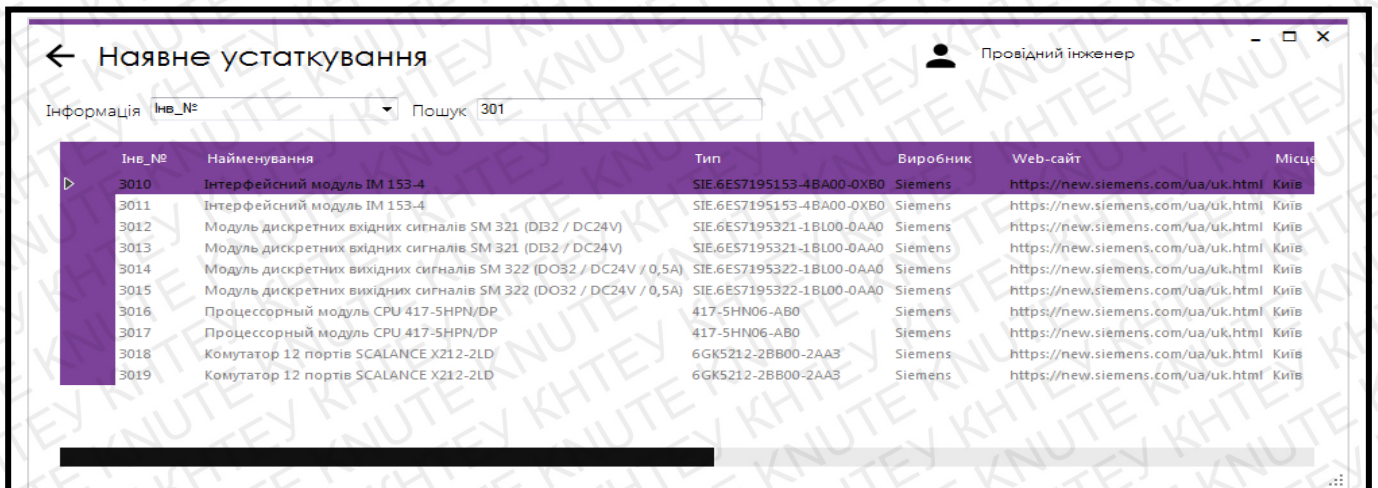


Рис. 3.13. Вікно програми «Нааявне устаткування»

Виконання форми Defects.cs (Додаток И) відображає вікно програми «Облік наявних дефектів», що зображене на Рис. 3.14. Пошук інформації проводиться за допомогою елементу інтерфейсу кнопки пошуку за інвентарним номером на основі запиту до таблиці бази даних dbo.Нааявне устаткування. Обов'язковим є заповнення основних текстових полів для виконання запиту щодо видалення інформації – з таблиці dbo.Нааявне устаткування і додати інформацію у таблицю dbo.Дефект.

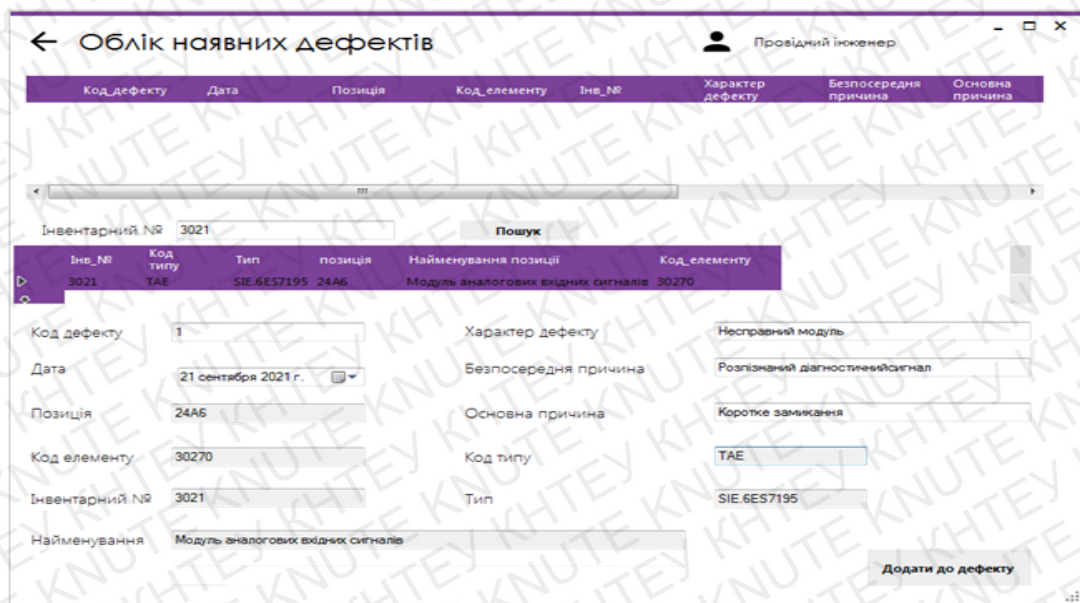


Рис. 3.14. Вікно програми «Облік наявних дефектів»

Виконання форми Reserv.cs (Додаток К) відображає вікно програми «Запасні частини та устаткування», що зображене на Рис. 3.15. Пошук виконується за допомогою елементу інтерфейсу кнопки по інвентарному номеру з таблиці бази даних dbo.Встановлене устаткування. Процес додавання до резерву відбувається за запитом у методі елементу інтерфейсу кнопки – вилучення устаткування з таблиці dbo.Встановлене устаткування і додавання до таблиці dbo.Резервне устаткування.

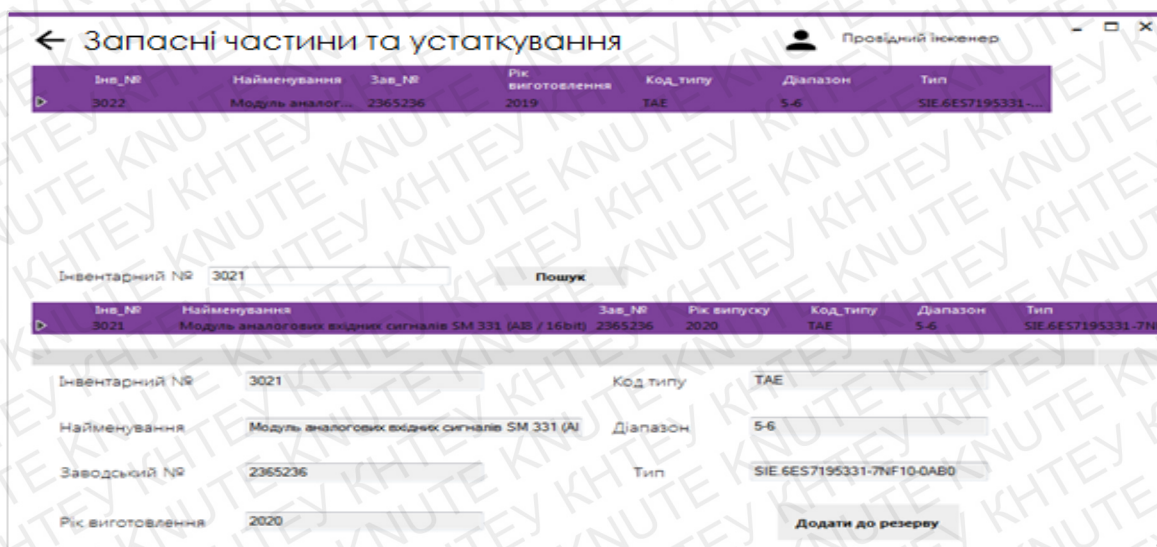


Рис. 3.15. Вікно програми «Устаткування в резерві»

При виконанні форм Report_Year.cs, Report_Month.cs, Work_order.cs (Додаток Л) відображаються вікна програми «Річний графік ТО», «Журнал обліку ТО», «Наряд на виконання робіт з ТО», що зображені на Рис. 3.16, Рис. 3.17, Рис. 3.18.

Для відображення документів використовується:

- а) документ YearReport.rpt та таблиця представлення бази даних dbo.Річний_графік_ТО з запитом для відображення періодичності проведення ТО;
- б) документ MonthReport.rpt та таблиця представлення бази даних dbo. Журнал_ТО_Місяць з розробленим запитом, що виконується при обранні періоду у елементах інтерфейсу дат та кнопки генерування звіту;

в) документ WorkOrder.rpt та таблиця представлення бази даних dbo. Наряд_на_виконання_робіт з розробленим запитом, що виконується при ініціалізації елементу інтерфейсу кнопки генерування наряду згідно введеної дати ТО.

№ п/п	Найменування	Тип	Позиція	Период. рем. мес. ТО-ТР-КР	Технічне обслуговування												Трудомістк. год-год ТО	Виконавець	Примітка
					1	2	3	4	5	6	7	8	9	10	11	12			
1	Автоматичний вимикач 400V 6KA, 1P+N, C, 25 A	5SY6025-7	IS-TC-165-PLC-001-1-4Q1	6 - ртс - ртс															
2	Автоматичний вимикач 400V 6KA, 1P+N, C, 25 A	5SY6025-7	IS-TC-165-PLC-001-1-4Q2	6 - ртс - ртс															
3	Одиночне реле REL-MR- 24DC / 21	2361105	IS-TC-165-PLC-001-1-1BK2	6 - ртс - ртс															
4	Автоматичний вимикач 400V 6KA, 1P+N, C, 25 A	5SY6025-7	RS-TC-167-PLC-001-4Q1	6 - ртс - ртс															
5	Одиночне реле REL-MR- 24DC / 21	2361105	IS-TC-165-PLC-001-1-1BK3	6 - ртс - ртс															
6	Автоматичний вимикач 400V 6KA, 1P+N, C, 25 A	5SY6025-7	RS-TC-167-PLC-001-1BK2	6 - ртс - ртс															

Рис. 3.16. Вікно програми «Річний графік ТО»

п/п №	Тип устаткування	Норма ТО	Позиція у графіку	Приміщення	Найменування	Дата обслуговування	Вид ТО	Підпис	Виконав/Перевірив	Дата друку:	29-10-2021
1	5SY6025-7	0,22	16Q2	167	Автоматичний вимикач 400V 6KA, 1P+N	01.12.2021				Екрановий щит	технічного стану
2	5IE 6ES7 196	0,87	28A2	163	Модуль дискретних вхідних сигналів 8M	01.12.2021					
3	5IE 6ES7 196	0,87	28A3	167	Модуль дискретних вхідних сигналів 8M	08.12.2021					
4	5IE 6ES7 196	0,87	28A5.1	165	Модуль дискретних вхідних сигналів 8M	28.12.2021					

Рис. 3.17. Вікно програми «Журнал обліку ТО»

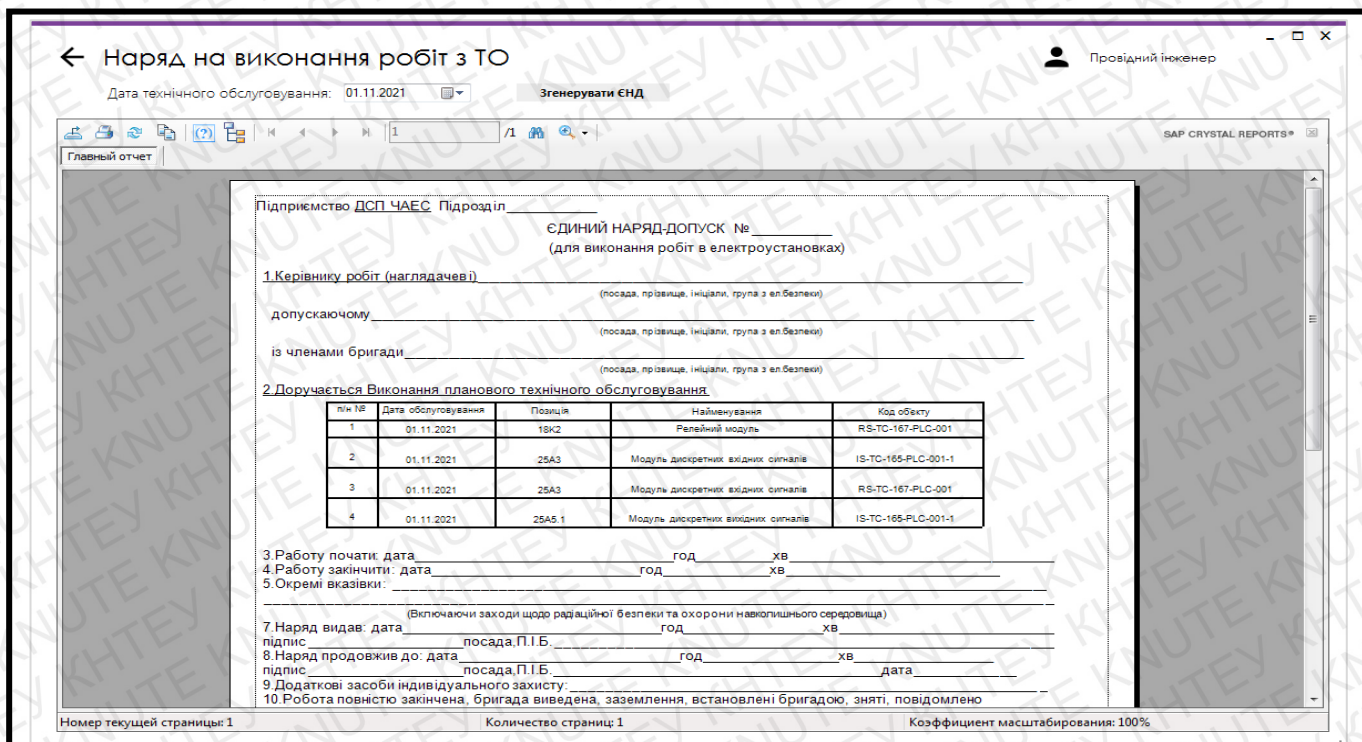


Рис. 3.18. Вікно програми «Наряд на виконання робіт з ТО»

3.2. Висновки до розділу 3

Отже, враховуючи вищезазначене, на стадії реалізації проекту в повному обсязі виконані функціональні вимоги до системи:

- надано доступ авторизованим користувачам та розроблено ролі користувачів з розмежування прав доступу: «Провідний інженер», «Керівництво»;
- наявний пошук устаткування по найменуванню, інвентарному номеру та періодів технічного обслуговування;
- реалізована можливість редагування, видалення та додавання події, що пов'язана з технічним обслуговуванням устаткуванням.
- реалізовано документи – звіт за місяць, звіт за рік, наряд на виконання робіт з технічного обслуговування;
- проведено детальний опис кожного вікна програми.

Інтерфейс системи є інтуїтивно зрозумілим для користувача та запрограмованим в повному обсязі згідно технічного завдання.

						Аркуш
						36
Зм.	Аркуш	№ докум.	Підпис	Дата	КНТЕУ 121 023-01.МР	

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

З проведених досліджень можна зробити наступні висновки:

1. Проведено аналіз предметної області, що пов'язана з застосуванням програмного забезпечення та аналіз роботи додатків аналогів;
2. Сформовано системні та функціональні вимоги до програмного забезпечення системи управління технічного обслуговування;
3. Розроблено моделі архітектури системи управління технічним обслуговуванням, що представлені у вигляді діаграми розгортання, діаграми класів, діаграми варіантів використання та діаграми діяльності;
4. Розроблено структурну модель бази даних системи управління технічним обслуговуванням;
5. В якості операційних систем було обрано Windows 7 або вищої версії та Windows Server 2012. В якості системи управління базою даних було обрано Microsoft SQL Server Management 2012;
6. Здійснено розмежування прав доступу до системи з визначеними ролями користувачів та реалізовано функціональні компоненти системи;
7. Проведено аналіз функціоналу розробленої системи, що дозволяє користувачам здійснювати швидке отримання необхідної інформації щодо технологічних груп до яких належить устаткування;
8. В процесі виконання проекту розроблено алгоритми пошуку устаткування, управління зв'язаних об'єктів програми та організація звітності;
9. В процесі виконання проекту розроблено технічне завдання та керівництва користувача та системного програміста;
10. Виконано тестування програмного забезпечення з позитивним результатом пройдених тестів;

<i>КНТЕУ 121 02з-01.МР</i>				
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>
Зав. каф.		Криворучко О.В.		01.11.21
Керівник		Рзасва С.Л.		01.11.21
Гарант		Токар В.В.		01.11.21
Розробив		Баллон М.В.		01.11.21
<i>Проектування програмного забезпечення системи управління технічним обслуговуванням промислового комплексу</i>				
<i>Висновки та пропозиції</i>				
<i>Стадія</i>				
<i>Аркуш</i>				
<i>Аркушів</i>				
ВП				
37				
40				
Факультет інформаційних технологій				
2 курс, 2з група				

Розроблена система повністю задовольняє потреби інженерно-технічного персоналу промислового комплексу НБК щодо комп'ютеризації роботи при проведенні технічного обслуговування устаткування.

Перспективами та пропозиціями подальшої розробки функціоналу даної системи управління технічним обслуговуванням є:

- вивід графічної інформації для отримання картини регулярності процесу проведення технічного обслуговування – реалізація діаграми Ганта;
- вивід графічної інформації щодо параметрів потоку відмов устаткування;
- створення план-графіку калібрування устаткування;
- створення план-графіку вимірювальних каналів;
- розробити можливість системи до автоматизованого збору інформації щодо стану наявного устаткування в локальній мережі промислового комплексу.

Слід зазначити, що промисловий комплекс НБК є новим і перспективним об'єктом зони відчуження, який потребує впровадження даного типу систем, у зв'язку з наявністю великого об'єму устаткування для подальшого зняття з експлуатації об'єкту «Укриття».

Мета випускного кваліфікаційного проекту досягнена, поставлені задачі виконано.

					<i>КНТЕУ 121 023-01.МР</i>	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		38

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Gennick J. SQL Pocket Guide / J. Gennick. – Sebastopol, O'Reilly Media, 2010. – 200 p.
2. Nagel C. Professional C# 5.0 and .NET 4.5.1 / C. Nagel. – Wiley, Wrox, 2014. – 1560 p.
3. Petkovic D. Microsoft SQL Server 2012 A Beginners Guide 5/E / D. Petkovic. – Glencoe, McGraw-Hill, 2012. – 832 p.
4. Балюн М.В. Проектування системи управління технічним обслуговуванням устаткування для промислових підприємств / М.В. Балюн // Software engineering комп'ютерних систем : зб. наук. ст. студ. /відп. ред. В. Я. Рассамакін. – Київ : Київ. нац. торг. – екон. ун-т, 2021. – 236 с.
5. Загальні відомості про Visual Studio [Електронний ресурс]. – Режим доступу: <https://docs.microsoft.com/en-us/visualstudio/get-started?view=vs-2022>
6. Керівництво користувача Crystal Reports 2013 [Електронний ресурс]. – Режим доступу: <https://help.sap.com/doc/businessoguides2013cr13pdf/2013.pdf>
7. Короткий огляд мови C# [Електронний ресурс]. – Режим доступу: <https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp>
8. Короткий огляд програмного забезпечення Champs [Електронний ресурс]. – Режим доступу: <https://www.champsinc.com/products/champs-cmms-eam>
9. Короткий огляд програмного забезпечення Fiix [Електронний ресурс]. – Режим доступу: <https://www.fiixsoftware.com/cmms/cmms-software>
10. Короткий огляд програмного забезпечення Галактика [Електронний ресурс]. – Режим доступу: <http://galaktika.ua/blog/erp.html>
11. Новий безпечний конфайнмент [Електронний ресурс]. – Режим доступу: <https://www.chnpp.gov.ua/nbk/index.html>

<i>КНТЕУ 121 02з-01.МР</i>							
Зм.	Аркуш	№ докум.	Підпис	Дата			
Зав. каф.		Криворучко О.В.		27.02.21			
Керівник		Рзасва С.Л.		27.02.21			
Гарант		Токар В.В.		27.02.21			
Розробив		Балюн М.В.		27.02.21			
<i>Список використаних джерел</i>							
<i>Проектування програмного забезпечення системи управління технічним обслуговуванням промислового комплексу</i>					Стадія	Аркуш	Аркушів
					СВД	39	40
					<i>Факультет інформаційних технологій 2 курс, 2з група</i>		

12. Основи тестування у Visual Studio [Електронний ресурс]. – Режим доступу: <https://docs.microsoft.com/uk-ua/visualstudio/test/create-unit-testsme>
13. Основні поняття уніфікованої мови моделювання UML [Електронний ресурс]. – Режим доступу: <https://sites.harding.edu/dsteil/301/notes/uml/uml.htm>
14. Особливості системи CMMS [Електронний ресурс]. – Режим доступу: https://www.ibm.com/topics/what-is-a-cmms?mhsrc=ibmsearch_a&mhq=CMMS
15. Особливості технічного обслуговування устаткування [Електронний ресурс]. – Режим доступу: <https://www.getmaintainx.com/learning-center/equipment-maintenance/>
16. Проектування інформаційних системи на основі UML [Електронний ресурс]. – Режим доступу: <https://eprinetes.cdu.edu.ua/analizvimogdopz/lekciie/uml>
17. Стратегії технічного обслуговування [Електронний ресурс]. – Режим доступу : <https://support.industry.siemens.com/cs/sc/5503/preventive-maintenance?lc=en-ww>

					<i>КНТЕУ 121 023-01.МР</i>	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		40

ТЕХНІЧНЕ ЗАВДАННЯ

Основним завданням щодо відповідності вимогам та якісно розробленого програмного забезпечення є вдало спроектоване до початку розробки технічне завдання. Тому, розкриємо основні положення технічного завдання до розробки програмного забезпечення системи управління технічним обслуговуванням промислового комплексу.

1. Загальні відомості

1.1. Найменування системи

Система управління технічним обслуговуванням.

1.1.1. Повне найменування системи

Система управління технічним обслуговуванням промислового комплексу Нового Безпечного Конфайнменту.

1.1.2. Скорочене найменування системи

СММС–НБК

1.2. Планові терміни початку та закінчення робіт

Плановий термін початку робіт – перший квартал 2022 року. Закінчення робіт – четвертий квартал 2022 року. Остаточну дату завершення робіт буде визначено за згодою сторін в межах графіку проекту.

1.3. Порядок оформлення і пред'явлення результатів робіт

Розробник, що обраний за результатами процесу закупівлі, після укладання договору має встановити програмне забезпечення та надати Замовнику послуги з впровадження та налаштування СММС-НБК. На базі наявного програмного забезпечення у ДСП «Чорнобильська АЕС» у відповідності з цими технічними вимогами та згідно із методикою впровадження даного програмного забезпечення.

Зм.	Аркуш	№ докум.	Підпис	Дата	<i>КНТЕУ 121 02з-01.МР</i>			
Зав. каф.		Криворучко О.В.		20.03.21	Проектування програмного забезпечення системи управління технічним обслуговуванням промислового комплексу <i>Технічне завдання</i>	Стадія	Аркуш	Аркушів
Керівник		Рзасва С.Л.		20.03.21		ТЗ	41	40
Гарант		Токар В.В.		20.03.21		Факультет інформаційних технологій 2 курс, 2з група		
Розробив		Баллон М.В.		20.03.21				

1.4. Головний бенефіціар та потенційні користувачі системи

Головний бенефіціар системи – ДСП «Чорнобильська АЕС».

Користувачами системи є працівники ДСП «Чорнобильська АЕС», а саме:

- системний програміст: 1 особа;
- основні користувачі системи за функціональним напрямком: 2 особи.

Під час виконання проекту, в рамках робіт з визначення ролей користувачів та функціональних компонентів, повинна бути уточнена основна кількість користувачів.

2. Мета та призначення створення системи

2.1. Призначення системи

Основне призначення системи є для отримання оперативної інформації інженерно–технічним персоналом НБК та подальшого планування та проведення робіт з технічного обслуговування.

2.2. Мета створення системи

Основною метою створення системи є реалізація середовища для забезпечення організації, планування та управління технічним обслуговуванням устаткування промислового комплексу.

3. Вимоги до системи

3.1. Вимоги до системи в цілому

- CMMS – НБК повинна розроблятися на базі сучасних комп'ютеризованих систем управління технічним обслуговуванням;
- CMMS – НБК повинна бути побудована на базі клієнт-серверної архітектури. Передача інформації по протоколу TCP/IP;
- CMMS – НБК повинна надавати доступ користувачам до інтерфейсу системи без використання додаткових програмних комплексів та сервісів.

3.1.1. Вимоги до структури та функціонування системи, перелік підсистем

До складу системи повинні входити наступні функціональні компоненти:

					<i>КНТЕУ 121 023-01.МР</i>	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		42

- «Технологічні групи»;
- «Облік інвентарних номерів»;
- «Наявне устаткування»;
- «Встановлене устаткування»;
- «Типи устаткування»;
- «Облік наявних дефектів»;
- «Запасні частини та устаткування»;
- «Річний графік ТО»;
- «Журнал обліку ТО»;
- «Наряд на виконання робіт з ТО».

3.1.1.1. Вимоги до режимів управління системою

CMMS – НБК повинна функціонувати в штатному режимі без зупинки в роботі при вводі в режим планового регламентованого технічного обслуговування.

3.1.2. Вимоги до надійності

3.1.2.1. Склад показників надійності до системи в цілому

Пристосованість системи та рівень надійності повинен здобуватися скоординованим застосуванням організаційно-технічних заходів, програмними та апаратними засобами.

Система повинна відповідати таким параметрам:

- середній час відновлення системи при аварійному режимі в роботі апаратної та програмної частини становить 72 години;
- час напрацювання на відмову повинен бути визначений відповідним показником основних компонентів апаратного забезпечення, що відібрані та погоджені з Замовником.

3.1.2.2. Вимоги до надійності програмного забезпечення

Надійність програмного забезпечення системи повинна забезпечуватися за рахунок:

					<i>КНТЕУ 121 023-01.МР</i>	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		43

- проведення загальних заходів налагодження, налаштування та усунення помилок;
- аналіз журналів системних повідомлень при виникненні помилок для подальшого виправлення та оновлення конфігурації;
- система повинна мати можливість відновлення з резервної копії за короткий термін з мінімальними втратами даних;
- зняття резервної копії з встановленої системи не входить до завдань Розробника.

Показники надійності системи повинні підтверджуватися в процесі випробувань та дослідної експлуатації.

3.1.3. Вимоги до захисту інформації від несанкціонованого доступу

Система має відповідати вимогам та нормам діючого законодавства щодо комплексної системи захисту інформації. Замовник, після впровадження системи, повинен мати змогу отримати атестат відповідності комплексної системи захисту інформації.

3.1.3.1. Вимоги до інформаційної безпеки

Для системи повинна бути надана можливість встановлення захищеного з'єднання між клієнтом та сервером, а також встановлені механізми контролю за повним видаленням даних.

3.1.3.2. Вимоги до антивірусного захисту

Виконання файлів програми системи повинні бути захищені від вірусних сигнатур та забезпечені механізмами перевірки контрольних сум.

3.1.3.3. Розмежування відповідальності ролей при доступі

В системі повинні бути наявні ролі для автентифікації в системі. Роль «Провідний інженер» та «Керівництво».

3.1.4. Вимоги до захисту від впливу зовнішніх факторів

Система повинна функціонувати при коливаннях напруги та забезпечена джерелами безперебійного живлення. Також температурного режиму

					<i>КНТЕУ 121 023-01.МР</i>	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		44

навколишнього середовища, що встановлені виробником апаратних засобів та допустимих радіаційних полів і вібрацій.

3.1.5. Вимоги безпеки

Для нормальної експлуатації обладнання системи необхідно встановити у шафи з автоматизованою системою захисту та аварійною сигналізацією.

3.2. Вимоги до видів забезпечення

3.2.1. Вимоги до математичного забезпечення

Програмне забезпечення повинно бути реалізоване у вигляді функціональних компонентів, що забезпечують виконання основних завдань, що перераховані у даному технічному завданні.

3.2.2. Вимоги до інформаційного забезпечення

3.2.2.1. Вимоги до інформаційної сумісності із суміжними системами

Система має бути спроектована таким чином, щоб зменшити навантаження на мережевий трафік між клієнтом і сервером, що знизить вимоги до каналів зв'язку.

3.2.2.2. Вимоги щодо застосування систем управління базами даних

Необхідно застосовувати систему управління базами даних SQL Server Management 2012 для організації та зберігання даних системи. Розробник не відповідає за закупівлю та надання пакету програми Замовнику.

3.2.2.3. Вимоги до процедури надання юридичної сили документам, що продукуються технічними засобами системи

Щодо вимог до надання юридичної сили документам, що продукуються технічними засобами системи можуть бути використані – електронний цифровий підпис або паперовий документ з печаткою та власноручним підписом правоздатної та уповноваженої особи.

4. Вимоги до програмного забезпечення

Для забезпечення захисту робочої станції та серверної частини розробленої системи необхідно використовувати ліцензоване антивірусне програмне

					<i>КНТЕУ 121 023-01.МР</i>	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		45

забезпечення. Операційні системи обох станцій та система управління базою даних повинна бути забезпечена відповідними ліцензіями.

5. Вимоги до технічного забезпечення

Для забезпечення функціонування системи Набувач надає необхідні технічні засоби для серверної частини та робочої станції. Параметри серверного апаратного забезпечення внесені до Таблиці 1.

Таблиця 1.

Параметри серверного апаратного забезпечення

Назва характеристики	Значення характеристики
Виробник	Dell
Модель	PowerEdge T40
Процесор	Intel Xeon E5-2600 v3
Операційна система	Microsoft Windows Server 2012
Набір мікросхем	Intel серії C610
Пам'ять	DIMM DDR4, 384 Гбайт, 16 Гбайт
Накопичувачі	SAS, SATA, NL-SAS
Контролери RAID	PERC H730
Мережевий контролер	1GbE
Відеоплата	Matrox G200 з iDRAC8, 16 Мбайт,

Параметри апаратного забезпечення робочої станції внесені до Таблиці 2.

Таблиця 2.

Параметри апаратного забезпечення робочої станції

Назва характеристики	Значення характеристики
Виробник	Dell
Модель	Precision
Процесор	Intel Core i7-6700 (Skylake)
Операційна система	Microsoft Windows 7 або вище
Набір мікросхем	Intel серії C610
Пам'ять	DDR4, 1000 (HDD), 8 Гбайт
Відеоплата	Інтегрована Intel HD Graphics 530

ПРОГРАМА ТА МЕТОДИКА ТЕСТУВАННЯ

З метою забезпечення коректності роботи та якості розробленого програмного забезпечення було розроблено тестові сценарії перевірки системи управління технічного обслуговування «СММС–НБК».

На основі вимог до програмного забезпечення за основу було обрано метод модульного тестування (Unit Testing), що в свою чергу тестує процедури чи окремі елементи системи, що підпадають під процес тестування Цей метод надає розробнику інформацію та визначає чи присутні безпосередні проблеми з програмним кодом та незалежними модулями, що тестуються [11]. Наведемо приклад одного з тестового сценарію системи методом модульного тестування.

Unit-test – Перевірка логіна та пароля:

```
using System;
using Microsoft.VisualStudio.TestTools.UnitTesting;
namespace LoginPass
public class Unitest
{
    [TestMethod()]
    public void login_pass_fail()
    {
        var accs = new Accesss();
        var actual = accs.Login("admin");
        var expected = false;
        Assert.AreEqual(actual, expected);
        var chek = new Pascheck();
        var result = chek.Password("1234");
        Assert.thisFalse(result);
    }
    [TestMethod()]
    public void login_pass_succ()
    {
        var loginService = new LoginService();
        var actual = loginService.Login("Administrator");
        var expected = true;
        Assert.AreEqual(actual, expected);
        var chek = new Pascheck();
        var result = chek.Password("pass");
        Assert.thisTrue(result);
    }
}
```

Даний тестовий сценарій містить логіку, що визначає чи правильно був переданий пароль чи ні у наявний елемент інтерфейсу кнопку фоходу до системи.

Зм.	Аркуш	№ докум.	Підпис	Дата	<i>КНТЕУ 121 02з-01.МР</i>			
Зав. каф.		Криворучко О.В.		18.10.21	Проектування програмного забезпечення системи управління технічним обслуговуванням промислового комплексу <i>Програма та методика тестування</i>	Стадія	Аркуш	Аркушів
Керівник		Рзасва С.Л.		18.10.21		ПМТ	47	40
Гарант		Токар В.В.		18.10.21		Факультет інформаційних технологій		
Розробив		Балюн М.В.		18.10.21		2 курс, 2з група		

Результатом виконання тесту буде наявність правильно введеного паролю користувача.

На основі вимог до програмного забезпечення зі сторони користувача, було обрано метод ручного тестування, що проводиться тестувальником (користувачем), який використовує набір сценаріїв. Сценарії включають послідовність дій, які повинен виконати оператор для аналізу результату тестування. Основою такого методу є розробка таблиці з кроками сценарію, очікуваною реакцією та співпадіння чи не співпадіння з заявленою реакцією [11]. Приклад проведення такого методу тестування наведено у Таблиці 3.

Таблиця 3.

Результати проведення ручного тестування вікна
«Технологічні групи – Оновлення інформації – Зона– Зберігання»

№ п/п	Дія	Реакція системи	Результат
1.	Запустіть програму «СММС-НБК»	З'являється стартове вікно програми (Рис.3.1)	Вірно
2.	Введіть логін та пароль та натисніть «Вхід»	Перехід до діалогового вікна (Рис.3.1)	Вірно
3.	Вікно «СММС-НБК» активне	Інформація присутня на інтерфейсі (Рис.3.3)	Вірно
4.	Виберіть і натисніть на компонент «Технологічні групи»	Перехід до діалогового вікна (Рис.3.4)	Вірно
5.	Перейдіть по дереву структури до «Зони»	Інформація присутня на інтерфейсі, присутні переходи (Рис 3.5.)	Вірно
6.	Внести інформацію в текстові поля та зберегти інформацію натиснувши кнопку «Зберегти»	Наявне діалогове вікно. Інформація записана до бази даних (Рис 3.5.)	Вірно
7.	Натиснути кнопку «Зберегти» без введення інформації у текстові поля	Текстові поля виділені червоним з написом «введіть данні»	Вірно

Таким чином, з огляду на виконані тестування, система є правильно реалізованою та може вводиться в експлуатацію.

КЕРІВНИЦТВО КОРИСТУВАЧА

Для коректної роботи з інтуїтивно зрозумілим інтерфейсом програми «СМMS – НБК» необхідно:

1. Розмістити отриманий файл СМMS.exe в обраному каталозі і виконати запуск файлу. Рекомендовано не відкривати більше одного екземпляру програми;
2. Рекомендовано вводити необхідну інформацію однією мовою;
3. У процесі авторизації користувач повинен ввести наданий логін та пароль та натиснути кнопку «Вхід» щоб перейти до наступного вікна програми. Для виходу з програми необхідно натиснути кнопку «Вихід». У разі виникнення помилки некоректності введення даних для входу до програми необхідно повторити вищезазначені дії. У разі виникнення помилки відсутності зв'язку з базою даних – звернутися до системного програміста;

4. Вікно «СМMS – НБК» – дає можливість користувачу обрати функціональний компонент системи натиснувши на обраний об'єкт;

5. Вікно «Технологічні групи» – користувач отримує інформацію щодо структури промислового комплексу НБК та можливість оновлення інформації у разі потреби щодо об'єктів технологічного процесу. Для виконання операцій необхідно натиснути на обраний об'єкт та заповнити відповідні поля вводу інформації. Користувач з роллю «Керівництво» не може проводити зміни інформації щодо технологічних груп;

6. Вікно «Типи устаткування» – при обранні необхідного типу устаткування користувачу надається інформація щодо наявних типів устаткування на промисловому комплексі НБК, а також дає можливість занести та встановити нове обладнання. У цьому вікні передбачено прямий вихід до пошуку інформації щодо встановленого та наявного устаткування на промисловому

Зм.	Аркуш	№ докум.	Підпис	Дата	<i>КНТЕУ 121 02з-01.МР</i>			
Зав. каф.		Криворучко О.В.		21.10.21	<i>Проектування програмного забезпечення системи управління технічним обслуговуванням промислового комплексу</i>	Стадія	Аркуш	Аркушів
Керівник		Рзасва С.Л.		21.10.21		КК	49	40
Гарант		Токар В.В.		21.10.21		<i>Факультет інформаційних технологій 2 курс, 2з група</i>		
Розробив		Баллон М.В.		21.10.21				
					<i>Керівництво користувача</i>			

комплексі за допомогою вікон «Наявне устаткування» та «Встановлене устаткування». Користувачу необхідно заповнити необхідні поля і натиснути кнопку «Зберегти» або «Встановити». Користувач з роллю «Керівництво» не може додавати нове устаткування;

7. Вікна «Наявне устаткування» та «Встановлене устаткування» – користувач може обрати з випадаючого списку необхідну для пошуку характеристику, ввести інформацію яку потрібно знайти, після чого натиснувши Enter отримати шукану інформацію;

8. Вікно «Облік інвентарних номерів» – користувачу надається можливість одразу встановити устаткування за допомогою інформації щодо резервних інвентарних номерів (резервного устаткування). Для виконання даної операції необхідно обрати та клацнути на рядок з устаткуванням та натиснути кнопку «Встановити»;

9. Вікно «Облік наявних дефектів» – користувач за допомогою пошуку за інвентарним номером устаткування, натиснувши кнопку «Пошук» та клацнувши на рядок з інформацією, присвоює код дефекту обладнанню, що вийшло з експлуатації і заносить його до дефектів, натиснувши на кнопку «Додати до дефекту»;

10. Вікно «Запасні частини та устаткування» – користувачу надається можливість за допомогою пошуку за інвентарним номером устаткування або частин обладнання, клацнувши на рядок з інформацією, додати його до резерву, натиснувши на кнопку «Додати до резерву»;

11. Вікна «Річний графік ТО», «Журнал обліку ТО», «Наряд на виконання робіт з ТО» – за допомогою даних вікон користувач може експортувати та роздрукувати річний звіт технічного обслуговування на наступний рік, журнал обліку технічного обслуговування обравши місяць проведення ТО, єдиний наряд допуск згідно обраної дати виконання ТО. Для користувача з роллю «Керівництво» вікно річного графіку є неактивним.

					<i>КНТЕУ 121 023-01.МР</i>	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		50

КЕРІВНИЦТВО СИСТЕМНОГО ПРОГРАМІСТА

Для функціонування програми «СММС–НБК» необхідна встановлена на робочу станцію технічного обслуговування операційна система Windows 7 або вищої версії. Програма розроблена на об'єктно-орієнтованій мові програмування С#, а база даних mdf.СММС спроектована у Microsoft SQL Server Management 2012. В процесі проектування використовувалося середовище розробки Microsoft Visual Studio Community 2019 року з пакетом Microsoft .NET Framework 4.5. та пакет Visual C # 2019.

У пакеті програми присутній виконуючий файл СММС.exe та база даних mdf. СММС, що знаходиться під керуванням MS SQL. Для установки програми необхідно скопіювати файл у відповідну теку робочої станції технічного обслуговування.

На сервері необхідно контролювати працездатність системи управління базою даних та цілісність база даних mdf. СММС. На початку необхідно проінсталювати систему управління базою даних Microsoft SQL Server Management 2012 та створити сервер з ім'ям, логіном та паролем наданим у пакеті програми згідно технічного завдання.

Для користувача сервера MS SQL в Security – Logins повинно бути:

- а) General: User Login: ім'я користувача;
 - відмічено SQL Server authentication та введено пароль;
 - знято відмітку з «User must change password at next login»;
 - Default database: СММС;
- б) Server Roles:
 - системний програміст являється адміністратором бази даних.

Для нього в Server Roles повинно бути відмічено sysadmin.

Зм.	Аркуш	№ докум.	Підпис	Дата	<i>КНТЕУ 121 02з-01.МР</i>			
Зав. каф.		Криворучко О.В.		21.10.21	<i>Проектування програмного забезпечення системи управління технічним обслуговуванням промислового комплексу</i>	Стадія	Аркуш	Аркушів
Керівник		Рзаєва С.Л.		21.10.21		КСП	51	40
Гарант		Токар В.В.		21.10.21		<i>Факультет інформаційних технологій</i>		
Розробив		Баллон М.В.		21.10.21				
					<i>Керівництво системного програміста</i>	<i>2 курс, 2з група</i>		

- User Mapping: відмічено в Database: CMMS, Default Schema: dbo;
- Database role: db_datareader, db_datawriter, public.

Наступним кроком необхідно приєднати базу даних mdf. CMMS до сервера. Обов'язковим є періодичне створення резервної копії бази даних.

Дана програма розроблена без необхідності окремої інсталяції. Вона не використовує переривання та системні функції, що є негативними для працездатності операційної систем.

Для реалізації додаткового функціоналу необхідно звернутися до розробника програми відповідно до технічного завдання.

					<i>КНТЕУ 121 023-01.МР</i>	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		52

ДОДАТКИ

ДОДАТОК А

Код класу Access

```
public partial class Access : MetroFramework.Forms.MetroForm
{
    public Access()
    {
        InitializeComponent();
    }
    private Main Main
    {
        get
        {
            throw new System.NotImplementedException();
        }
        set
        {
        }
    }
    private void metroButtonEnter_Click_1(object sender, EventArgs e)
    {
        SqlConnection openacc = new SqlConnection(@"Data Source=WIN-CTJO8OTQQ5Z\SQLEXPRESS;
        Initial Catalog=CMMS;Integrated Security=True");
        SqlDataAdapter sda = new SqlDataAdapter("Select [Вхід].[Роль] from [Вхід] where [Вхід].[Логін]='" + textBoxlog.
        Text + "' and [Вхід].[Пароль]='" + textBoxpass.Text + "'", openacc);
        DataTable dt = new System.Data.DataTable();
        sda.Fill(dt);
        try
        {
            openacc.Open();
        }
        catch
        {
            MetroFramework.MetroMessageBox.Show(this, "Неможливо підключитися до бази даних. Будь ласка,
            зверніться до адміністратора", "Помилка підключення до бази даних", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
        if (dt.Rows.Count == 1)
        {
            Main ss = new Main(dt.Rows[0][0].ToString());
            this.Hide();
            ss.Show();
        }
        else
        {
            MetroFramework.MetroMessageBox.Show(this, "Не вірний логін чи пароль", "Помилка",
            MessageBoxButtons.OK, MessageBoxIcon.Warning);
        }
    }
}
```

Код класу Main

```
public partial class Main : MetroFramework.Forms.MetroForm
{
    public Main(string role)
    {
        InitializeComponent();
        User_account.Text = role;
    }
    private void User_image_Click(object sender, EventArgs e)
    {
        this.Hide();
        Access acc = new Access();
        acc.ShowDialog();
    }
    private void mTech_Groups_Click(object sender, EventArgs e)
    {
        TechnGroups tech = new TechnGroups();
        tech.Main = this;
        tech.ShowDialog();
    }
    private void mType_Equipment_Click(object sender, EventArgs e)
    {
        Equipment_Type eqt = new Equipment_Type();
        eqt.Main = this;
        eqt.ShowDialog();
    }
    private void mInvenNumb_Click(object sender, EventArgs e)
    {
        Audit_Inventar inv = new Audit_Inventar();
        inv.Main = this;
        inv.ShowDialog();
    }
    private void mEquipment_Click(object sender, EventArgs e)
    {
        Equipment eq = new Equipment();
        eq.Main = this;
        eq.ShowDialog();
    }
    private void mInstall_Equipment_Click(object sender, EventArgs e)
    {
        Installed_Equipment inst = new Installed_Equipment();
        inst.Main = this;
        inst.ShowDialog();
    }
    private void mDefect_Click(object sender, EventArgs e)
    {
        Defects def = new Defects();
        def.Main = this;
        def.ShowDialog();
    }
    private void mYear_report_Click(object sender, EventArgs e)
    {
        Report_Year year = new Report_Year();
        year.ShowDialog();
    }
    private void mReserv_Click(object sender, EventArgs e)
    {
        Reserv res = new Reserv();
        res.Main = this;
        res.ShowDialog();
    }
    private void mMonth_report_Click(object sender, EventArgs e)
    {
        Report_Month month = new Report_Month();
        month.Main = this;
    }
}
```


Продовження дод. Б

```
month.ShowDialog();  
}  
private void mNaryad_Click(object sender, EventArgs e)  
{  
    Work_order work = new Work_order();  
    work.Main = this;  
    work.ShowDialog();  
}  
}
```

Код класу TechNCS

```
public partial class TechNCS : Form
{
    public TechNCS()
    {
        InitializeComponent();
    }

    private void TechNCS_Load(object sender, EventArgs e)
    {
        this.об_ектиTableAdapter.Fill(this.cMMS_DataSet.Об_екти);
        this.технологічні_групиTableAdapter.Fill(this.cMMS_DataSet.Технологічні_групи);
        this.системиTableAdapter.Fill(this.cMMS_DataSet.Системи);
        this.зонаTableAdapter.Fill(this.cMMS_DataSet.Зона);
    }

    private void User_account_Click(object sender, EventArgs e)
    {
        this.зонаTableAdapter.Update(this.cMMS_DataSet.Зона);
        this.системиTableAdapter.Update(this.cMMS_DataSet.Системи);
        this.технологічні_групиTableAdapter.Update(this.cMMS_DataSet.Технологічні_групи);
        this.об_ектиTableAdapter.Update(this.cMMS_DataSet.Об_екти);
    }
}
```

Код класу TechnGroups

```

public partial class TechnGroups : MetroFramework.Forms.MetroForm
{
    public TechnGroups()
    {
        InitializeComponent();
    }

    private Form f;

    private void treeView1_AfterSelect_1(object sender, TreeViewEventArgs e)
    {
        TreeNode node = treeView1.SelectedNode;
        switch (node.Text)
        {
            case "НБК":
                f.Dispose();
                f = new TechNCS();
                f.TopLevel = false;
                mMainPanelFG.Controls.Add(f);
                f.FormBorderStyle = System.Windows.Forms.FormBorderStyle.None;
                f.Dock = DockStyle.Fill;
                f.Show();
                break;
            case "Зони":
                f.Dispose();
                f = new Zones_TG();
                f.TopLevel = false;
                mMainPanelFG.Controls.Add(f);
                f.FormBorderStyle = System.Windows.Forms.FormBorderStyle.None;
                f.Dock = DockStyle.Fill;
                f.Show();
                break;
            case "Системи":
                f.Dispose();
                f = new Systems_TG();
                f.TopLevel = false;
                mMainPanelFG.Controls.Add(f);
                f.FormBorderStyle = System.Windows.Forms.FormBorderStyle.None;
                f.Dock = DockStyle.Fill;
                f.Show();
                break;
            case "Технологічні групи":
                f.Dispose();
                f = new Groups_TG();
                f.TopLevel = false;
                mMainPanelFG.Controls.Add(f);
                f.FormBorderStyle = System.Windows.Forms.FormBorderStyle.None;
                f.Dock = DockStyle.Fill;
                f.Show();
            case "Об'єкти":
                f.Dispose();
                f = new Objects_TG();
                f.TopLevel = false;
                mMainPanelFG.Controls.Add(f);
                f.FormBorderStyle = System.Windows.Forms.FormBorderStyle.None;
                f.Dock = DockStyle.Fill;
                f.Show();
                break;
        }
    }

    public Main Main { get; internal set; }
}

```


Продовження дод. Г

```
private void TechnGroups_Load_1(object sender, EventArgs e)
{
    User_account.Text = ((Form)this.Main).Controls["User_account"].Text;
    this.treeView1.Nodes[0].Expand();

    f = new TechNCS();
    f.TopLevel = false;
    mMainPanelFG.Controls.Add(f);
    f.FormBorderStyle = System.Windows.Forms.FormBorderStyle.None;
    f.Dock = DockStyle.Fill;
    f.Show();
}
private void mMainPanelFG_Paint(object sender, PaintEventArgs e)
{
    DialogResult = DialogResult.OK;
}
}
```

Код класу Equipment_Type

```

public partial class Equipment_Type : MetroFramework.Forms.MetroForm
{
    SqlConnection openacc = new SqlConnection(@"Data Source=WIN-CTJO8OTQQ5Z\SQLEXPRESS;
Initial Catalog=CMMS;Integrated Security=True");
    SqlCommand cmd;
    SqlDataAdapter adpt;
    DataTable dt;
    public Equipment_Type()
    {
        InitializeComponent();
        displayDataSIT();
        displayDataAITZ();
        displayDataTSVT();
    }
    public Main Main { get; internal set; }
    public Equipment Equipment { get; internal set; }
    private void User_account_LinkClicked(object sender, EventArgs e)
    {
        DialogResult = DialogResult.OK;
    }
    private void Equipment_Type_Load(object sender, EventArgs e)
    {
        this.наявне_устаткуванняTableAdapter.Fill(this.cMMS_DataSet.Наявне_устаткування);
        User_account.Text = ((Form)this.Main).Controls["User_account"].Text;

        if (((Form)this.Main).Controls["User_account"].Text != "Administrator")
        {
            metroButton1.Enabled = true;
            metroButton2.Enabled = true;
            metroButton5.Enabled = true;
            metroButton6.Enabled = true;
            metroButton8.Enabled = true;
            metroButton9.Enabled = true;
            metroButton11.Enabled = true;
        }
        //Для ЗВТ
        adpt = new SqlDataAdapter("Select * From [Вид калібрування]",openacc);
        dt = new DataTable("Вид калібрування");
        adpt.Fill(dt);
        comboBox2.DataSource = dt;
        comboBox2.DisplayMember = "Код калібрування";
        comboBox2.ValueMember = "Код калібрування";

        adpt = new SqlDataAdapter("Select * From [Тип устаткування] WHERE [Код_типу ] LIKE '%'",openacc);
        dt = new DataTable("Тип устаткування");

        adpt.Fill(dt);

        comboBox3.DataSource = dt;
        comboBox3.DisplayMember = "Код_типу"
        adpt = new SqlDataAdapter("Select * From [Об'єкти]",openacc);
        dt = new DataTable("Об'єкти");

        adpt.Fill(dt);

        comboBox4.DataSource = dt;
        comboBox4.DisplayMember = "Код об'єкту";
        comboBox4.ValueMember = "Код об'єкту";

        adpt = new SqlDataAdapter("Select * From [Наявне устаткування] WHERE [Інв_№] BETWEEN 1000 AND
2000",openacc);
        dt = new DataTable("Наявне устаткування");
    }
}

```

Продовження дод. Д

```
comboBox5.DataSource = dt;
comboBox5.DisplayMember = "Інв_№";
comboBox5.ValueMember = "Інв_№";

adpt = new SqlDataAdapter("Select * From [Приміщення]",орепасс);
dt = new DataTable("Приміщення");

adpt.Fill(dt);

//для АгаТЗ

adpt = new SqlDataAdapter("Select * From [Вид калібрування]",орепасс);
dt = new DataTable("Вид калібрування");

adpt.Fill(dt);

comboBox11.DataSource = dt;
comboBox11.DisplayMember = "Код калібрування";
comboBox11.ValueMember = "Код калібрування";

adpt = new SqlDataAdapter("Select * From [Тип устаткування] WHERE [Код_типу] LIKE 'P%' ",орепасс);
dt = new DataTable("Тип устаткування");

adpt.Fill(dt);

comboBox10.DataSource = dt;
comboBox10.DisplayMember = "Код_типу";
comboBox10.ValueMember = "Код_типу";

adpt = new SqlDataAdapter("Select * From [Об'єкти]",орепасс);
dt = new DataTable("Об'єкти");

adpt.Fill(dt);

comboBox18.DataSource = dt;
comboBox18.DisplayMember = "Код об'єкту";
comboBox18.ValueMember = "Код об'єкту";

adpt = new SqlDataAdapter("Select * From [Наявне устаткування] WHERE [Інв№] BETWEEN 3000 AND
4000",орепасс);
dt = new DataTable("Наявне устаткування");

adpt.Fill(dt);

comboBox17.DataSource = dt;
comboBox17.DisplayMember = "Інв_№";
comboBox17.ValueMember = "Інв_№";

adpt = new SqlDataAdapter("Select * From [Приміщення]",орепасс);
}
```


Код класу Audit_Inventar

```

public partial class Audit_Inventar : MetroFramework.Forms.MetroForm
{
    SqlConnection openacc = new SqlConnection(@"Data Source=WIN-CTJO8OTQ5Z \SQLEXPRESS;
Initial Catalog=CMMS;Integrated Security=True");
    SqlCommand cmd;
    SqlDataAdapter adpt;
    DataTable dt;
    public Audit_Inventar()
    {
        InitializeComponent();
        displayDataReserv();
    }

    public void displayDataReserv()
    {
        openacc.Open();
        adpt = new SqlDataAdapter(@"
SELECT [Інв_№], Найменування, [Зав_№], [Рік виготовлення], Код_типу, Діапазон, Тип
FROM [Резервне устаткування]", openacc);
        dt = new DataTable();
        adpt.Fill(dt);
        metroGridtxt.DataSource = dt;
        openacc.Close();
    }

    public Main Main { get; internal set; }

    private void metroGridtxt_CellDoubleClick(object sender, DataGridViewCellEventArgs e)
    {
    }

    public void Clear()
    {
        textBoxinv.Text = "";
        textBoxnaim.Text = "";
        textBoxzav.Text = "";
        textBoxgod.Text = "";
        textBoxdiap.Text = "";
        textBoxtip.Text = "";
    }
    private void mInsert_button_Click(object sender, EventArgs e)
    {
        try
        {
            if (MetroFramework.MetroMessageBox.Show(this, "Встановити устаткування?", "Збереження",
                MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
            {
                if (con.State == ConnectionState.Closed) openacc.Open();

                cmd = new SqlCommand(@"
INSERT INTO
[dbo].[Встановлене устаткування]
VALUES (@inv, @naimen)          INSERT INTO
[dbo].[Наявне устаткування]
VALUES (@zavnom, @god, @kodtura, @diapazon);
INSERT INTO [dbo].[Тип устаткування] VALUES (@tip);
DELETE FROM [dbo].[Резервне устаткування] WHERE [Інв_№] = " + textBoxinv.Text + " ", openacc);
                cmd.Parameters.AddWithValue("inv", textBoxinv.Text);
                cmd.Parameters.AddWithValue("naimen", textBoxnaim.Text);
                cmd.Parameters.AddWithValue("zavnom", textBoxzav.Text);
            }
        }
        catch { }
    }
}

```

Продовження дод. Е

```
cmd.Parameters.AddWithValue("god", textBoxgod.Text);
cmd.Parameters.AddWithValue("kodtypa", textBoxtyp.Text);
cmd.Parameters.AddWithValue("diapazon", textBoxdiap.Text);
cmd.Parameters.AddWithValue("tip", textBoxtip.Text);
cmd.ExecuteNonQuery();
openacc.Close();
displayDataReserv();
Clear();
MetroFramework.MetroMessageBox.Show(this, "Дані збережено!", "Збереження", MessageBoxButtons.OK,
MessageBoxIcon.Information);
}
}
catch (Exception mb)
{
    MessageBox.Show(mb.Message);
}
finally
{
    if (con.State == connectionState.Open)openacc.Close();
}
}

private void Useraccount_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
    DialogResult = DialogResult.OK;
}

private void metroGridtxt_CellContentClick(object sender, DataGridViewCellEventArgs e)
{
    textBoxinv.Text = metroGridtxt.Rows[e.RowIndex].Cells[0].Value.ToString();
    textBoxnaim.Text = metroGridtxt.Rows[e.RowIndex].Cells[1].Value.ToString();
    textBoxzav.Text = metroGridtxt.Rows[e.RowIndex].Cells[2].Value.ToString();
    textBoxgod.Text = metroGridtxt.Rows[e.RowIndex].Cells[3].Value.ToString();
    textBoxtyp.Text = metroGridtxt.Rows[e.RowIndex].Cells[4].Value.ToString();
    textBoxdia.Text = metroGridtxt.Rows[e.RowIndex].Cells[5].Value.ToString();
    textBoxtip.Text = metroGridtxt.Rows[e.RowIndex].Cells[6].Value.ToString();
}

private void Audit_Inventar_Load(object sender, EventArgs e)
{
    User_account.Text = ((Form)this.Main).Controls["User_account"].Text;
}
}
```

Код класу Installed_Equipment

```

public partial class Installed_Equipment: MetroFramework.Forms.MetroForm
{
    public Installed_Equipment()
    {
        InitializeComponent();
    }

    public Main Main { get; internal set; }
    public Equipment_Type Equipment_Type { get; internal set; }

    private void Installed_Equipment_Load_1(object sender, EventArgs e)
    {
        User_account.Text = ((Form)this.Main).Controls["User_account"].Text;
    }

    private void metroButtonInst_Click(object sender, EventArgs e)
    {
        DialogResult = DialogResult.OK;
    }

    private void Useraccount_Click(object sender, EventArgs e)
    {
        DialogResult = DialogResult.OK;
    }

    private void textBox1_TextChanged_1(object sender, EventArgs e)
    {
        SqlConnection openacc = new SqlConnection(@"Data Source=WIN-CTJO8OTQQ5Z\SQLEXPRESS;
Initial Catalog=CMMS;Integrated Security=True");

        SqlDataAdapter adpt;
        DataTable dt;

        if (comboBox1.Text == "Позиція")
        {
            openacc.Open();
            adpt = new SqlDataAdapter(@"SELECT Позиція, [Найменування позиції], Код_елементу, [Код об'єкту],
[Інв_№], Приміщення, Відмітка, Ряд, Вісь, ДНОП, [Не обслуговується]
FROM [Встановлене устаткування] where [Встановлене устаткування].[Позиція] like '" + textBox1.Text + "%'",openacc);
            dt = new DataTable();
            adpt.Fill(dt);
            metroGridtxt.DataSource = dt;
        }
        else if (comboBox1.Text == "Найменування позиції")
        {
            adpt = new SqlDataAdapter(@"SELECT Позиція, [Найменування позиції], Код_елементу, [Код об'єкту],
[Інв_№], Приміщення, Відмітка, Ряд, Вісь, ДНОП, [Не обслуговується]
FROM [Встановлене устаткування] where [Встановлене устаткування].[Найменування позиції] like '" + textBox1.Text +
%"',openacc);
            dt = new DataTable();
            adpt.Fill(dt);
            metroGridtxt.DataSource = dt;
        }
        else if (comboBox1.Text == "Код_елементу")
        {
            openacc.Open();
            adpt = new SqlDataAdapter(@"SELECT Позиція, [Найменування позиції], Код_елементу, [Код об'єкту],
[Інв_№], Приміщення, Відмітка, Ряд, Вісь, ДНОП, [Не обслуговується]
FROM [Встановлене устаткування] where [Встановлене устаткування].[Код_елементу] like '" + textBox1.Text + "%'",
con);
            dt = new DataTable();
        }
    }
}

```


Продовження дод. Ж

```
        adpt.Fill(dt);
        metroGridtxt.DataSource = dt;
    }
    else if (comboBox1.Text == "Код об'єкту")
    {
        FROM [Встановлене устаткування] where [Встановлене устаткування].[Код_елементу] like "" + textBox1.Text + "%' ",
        con);
        dt = new DataTable();
        adpt.Fill(dt);
        metroGridtxt.DataSource = dt;
    }
    else if (comboBox1.Text == "Код об'єкту")
    {
        openacc.Open();
        adpt = new SqlDataAdapter(@"SELECT Позиція, [Найменування позиції], Код_елементу, [Код об'єкту],
[Інв_№], Приміщення, Відмітка, Ряд, Вісь, ДНОП, [Не обслуговується]
FROM [Встановлене устаткування] where [Встановлене устаткування].[Код об'єкту] like "" + textBox1.Text + "%'
",openacc);
        dt = new DataTable();
        adpt.Fill(dt);
        metroGridtxt.DataSource = dt;
    }
    else if (comboBox1.Text == "Інв_№")
    {
        openacc.Open();
        adpt = new SqlDataAdapter(@"SELECT Позиція, [Найменування позиції], Код_елементу, [Код об'єкту],
[Інв_№], Приміщення, Відмітка, Ряд, Вісь, ДНОП, [Не обслуговується]
FROM [Встановлене устаткування] where [Встановлене устаткування].[Інв_№] like "" + textBox1.Text + "%' ",openacc);
        dt = new DataTable();
        adpt.Fill(dt);
        metroGridtxt.DataSource = dt;
    }
    else if (comboBox1.Text == "Приміщення")
    {
        openacc.Open();
        adpt = new SqlDataAdapter(@"SELECT Позиція, [Найменування позиції], Код_елементу, [Код об'єкту],
[Інв_№], Приміщення, Відмітка, Ряд, Вісь, ДНОП, [Не обслуговується]
FROM [Встановлене устаткування] where [Встановлене устаткування].[Приміщення] like "" + textBox1.Text + "%' ",
openacc);
        dt = new DataTable();
        adpt.Fill(dt);
        metroGridtxt.DataSource = dt;
    }
    else if (comboBox1.Text == "Не обслуговується")
    {
        openacc.Open();
        adpt = new SqlDataAdapter(@"SELECT Позиція, [Найменування позиції], Код_елементу, [Код об'єкту],
[Інв_№], Приміщення, Відмітка, Ряд, Вісь, ДНОП, [Не обслуговується]
FROM [Встановлене устаткування] where [Встановлене устаткування].[Не обслуговується] like "" + textBox1.Text + "%'
",openacc);
        dt = new DataTable();
        adpt.Fill(dt);
        metroGridtxt.DataSource = dt;
    }
    else
    {
    }
    }
}
```

Код класу Defects

```

public partial class Defects : MetroFramework.Forms.MetroForm
{
    SqlConnection openacc = new SqlConnection(@"Data Source=WIN-CTJO8OTQQ5Z\SQLEXPRESS;
Initial Catalog=CMMS;Integrated Security=True");
    SqlCommand cmd;
    SqlDataAdapter adpt;
    DataTable dt;

    public Main Main { get; internal set; }

    public Defects()
    {
        InitializeComponent();
        displayDataDept();
    }

    public void displayDataDept()
    {
        openacc.Open();
        adpt = new SqlDataAdapter("SELECT [Дефект].[Код_дефекту], [Відомість дефектації].Дата,
Відомість дефектації.Позиція, [Відомість дефектації].Код_елементу, [Відомість дефектації].[Інв_№],
[Відомість дефектації].[Характер дефекту], [Дефект].[Безпосередня причина], [Дефект].[Основна причина],
[Відомість дефектації].Код_типу, [Відомість дефектації].Тип, [Відомість дефектації].Найменування,
[Відомість дефектації].[Виконання робіт] FROM[Відомість дефектації] INNER JOIN [Дефект]
ON[Відомість дефектації].[Код_дефекту] = [Дефект].[Код_дефекту]",openacc);
        dt = new DataTable();
        adpt.Fill(dt);
        metroGridtxt.DataSource = dt;
        openacc.Close();
    }

    private void Useraccount_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
    {
        DialogResult = DialogResult.OK;
    }

    private void Defects_Load_1(object sender, EventArgs e)
    {
        User_account.Text = ((Form)this.Main).Controls["User_account"].Text;
    }

    private void mSearchthisButton_Click(object sender, EventArgs e)
    {
        openacc.Open();
        adpt = new SqlDataAdapter("SELECT [Наявне устаткування].[Інв_№], [Наявне устаткування].[Код типу],
[Тип устаткування].Тип, [Встановлене устаткування].позиція, [Встановлене устаткування].[Найменування позиції],
[Встановлене устаткування].Код_елементу FROM[Наявне устаткування] INNER JOIN [Тип устаткування] ON
[Наявне устаткування].[Код типу] = [Тип устаткування].Код_типу INNER JOIN [Встановлене устаткування]
ON[Наявне устаткування].[Інв_№] = [Встановлене устаткування].[Інв_№] WHERE [Наявне устаткування].[Інв_№] = "
+ int.Parse(textBox1.Text) + " ",openacc);
        adpt.Fill(dt);
        metroGrid2.DataSource = dt;
        openacc.Close();
    }

    private void mInsDeptButton_Click(object sender, EventArgs e)
    {
        try
        {
            if (MetroFramework.MetroMessageBox.Show(this, "Додати до дефекту?", "Збереження",
MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
            {
                if (con.State == connectionState.Closed)openacc.Open();
            }
        }
    }
}

```

Продовження дод. И

```
cmd = new SqlCommand(@"  
INSERT INTO  
[dbo].[Дефект]  
VALUES(@coddept, @prichina, @korenprich);  
INSERT INTO  
[dbo].[Відомість дефектації]  
VALUES (@date, @pozition, @elementcode,@invcode, @harkter,@typekod, @type, @naimen,  
@coddept, @vykrob);  
DELETE FROM [dbo].[Наявне устаткування] WHERE [Інв_№] = " + textBox6.Text + " ;  
DELETE FROM [dbo].[Встановлене устаткування] WHERE [Інв_№] = " + textBox6.Text + " ;openacc);  
  
cmd.Parameters.AddWithValue("date", dateTimePicker1.Value);  
cmd.Parameters.AddWithValue("pozition", textBox4.Text);  
cmd.Parameters.AddWithValue("elementcode", textBox5.Text);  
cmd.Parameters.AddWithValue("invcode", textBox6.Text);  
cmd.Parameters.AddWithValue("harkter", textBox7.Text);  
cmd.Parameters.AddWithValue("typekod", textBox10.Text);  
cmd.Parameters.AddWithValue("type", textBox11.Text);  
cmd.Parameters.AddWithValue("naimen", textBox12.Text);  
  
cmd.Parameters.AddWithValue("vykrob", textBox3.Text);  
cmd.Parameters.AddWithValue("coddept", textBox2.Text);  
cmd.Parameters.AddWithValue("prichina", textBox8.Text);  
cmd.Parameters.AddWithValue("korenprich", textBox9.Text);  
  
cmd.ExecuteNonQuery();  
  
openacc.Close();  
displayDataDept();  
Clear();  
  
MetroFramework.MetroMessageBox.Show(this, "Дані збережено!", "Збереження",  
MessageBoxButtons.OK,  
MessageBoxIcon.Information);  
}  
catch (Exception ex)  
{  
MetroFramework.MetroMessageBox.Show(this, "Дані не збережено!", "Збереження", MessageBoxButtons.OK,  
MessageBoxIcon.Information);  
}  
finally  
{  
if (con.State == openaccnecionState.Open)openacc.Close();  
}  
}
```


Код класу Reserv

```

public partial class Reserv : MetroFramework.Forms.MetroForm
{
    SqlConnection openacc = new SqlConnection(@"Data Source=WIN-CTJO8OTQQ5Z\SQLEXPRESS;
Initial Catalog=CMMS;Integrated Security=True");
    SqlCommand cmd;
    SqlDataAdapter adpt;
    DataTable dt;
    public Reserv()
    {
        InitializeComponent();
        displayDataReserv();
    }
    public void displayDataReserv()
    {
        openacc.Open();
        adpt = new SqlDataAdapter(@"
SELECT [Инв_№], Найменування, [Зав_№], [Рік виготовлення], Код_типу, Діапазон, Тип
FROM [Резервне устаткування]",openacc);
        dt = new DataTable();
        adpt.Fill(dt);
        metroGridtxt.DataSource = dt;
        openacc.Close();
    }
    public Main Main{ get; internal set; }
    public void Clear()
    {
        textBoxinv.Text = "";
        textBoxnaim.Text = "";
        textBoxzav.Text = "";
        textBoxgod.Text = "";
        textBoxdiap.Text = "";
        textBoxtip.Text = "";
    }

    private void mSearch_button_Click(object sender, EventArgs e)
    {
        openacc.Open();
        adpt = new SqlDataAdapter(@"SELECT
        [Встановлене устаткування].[Инв_№], [Тип устаткування].Найменування,
[Наявне устаткування].[Зав_№], [Наявне устаткування].[Рік випуску],
[Тип устаткування].Код_типу, [Наявне устаткування].Діапазон, [Тип устаткування].Тип
FROM[Встановлене устаткування] INNER JOIN
[Наявне устаткування] ON[Встановлене устаткування].[Инв_№] = [Наявне устаткування].[Инв_№]
INNER JOIN
[Тип устаткування] ON[Наявне устаткування].[Код типу] = [Тип устаткування].Код_типу
WHERE [Наявне устаткування].[Инв_№] = "" + int.Parse(textBox1.Text) + """,openacc);
        dt = new DataTable();
        openacc.Close();
    }

    private void mInsert_button_Click(object sender, EventArgs e)
    {
        try
        {
            if (MetroFramework.MetroMessageBox.Show(this, "Додати до резерву?", "Збереження",
            MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
            {
                if (con.State == openaccnnectionState.Closed)openacc.Open();
                cmd = new SqlCommand(@"

```

Продовження дод. К

```
INSERT INTO
[dbo].[Резервне устаткування]
VALUES (@inv, @naimen, @zavnom, @god, @kodtyra, @diapazon, @tip);
DELETE FROM [dbo].[Встановлене устаткування] WHERE [Пнв_№] = "" + textBox2.Text + "",openacc);

cmd.Parameters.AddWithValue("inv", textBox2.Text);
cmd.Parameters.AddWithValue("naimen", textBox3.Text);
cmd.Parameters.AddWithValue("zavnom", textBox4.Text);
cmd.Parameters.AddWithValue("god", textBox5.Text);
cmd.Parameters.AddWithValue("kodtyra", textBox6.Text);
cmd.Parameters.AddWithValue("diapazon", textBox7.Text);
cmd.Parameters.AddWithValue("tip", textBox8.Text);

cmd.ExecuteNonQuery();

openacc.Close();
displayDataReserv();
Clear();
MetroFramework.MetroMessageBox.Show(this, "Данні збережено", "Збереження",
MessageBoxButtons.OK,
MessageBoxIcon.Information);
}
}
catch (Exception ex)
{
MetroFramework.MetroMessageBox.Show(this, "Дані не збережено!", "Збереження", MessageBoxButtons.OK,
MessageBoxIcon.Information);
}
finally
{
if (con.State ==connectionState.Open)openacc.Close();
}
}
private void User_account_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
DialogResult = DialogResult.OK;
}
private void metroGridtxt_CellContentClick(object sender, DataGridViewCellEventArgs e)
{
textBoxinv.Text = metroGridtxt.Rows[e.RowIndex].Cells[0].Value.ToString();
textBoxnaim.Text = metroGridtxt.Rows[e.RowIndex].Cells[1].Value.ToString();
textBoxzav.Text = metroGridtxt.Rows[e.RowIndex].Cells[2].Value.ToString();
textBoxgod.Text = metroGridtxt.Rows[e.RowIndex].Cells[3].Value.ToString();
textBoxoty.Text = metroGridtxt.Rows[e.RowIndex].Cells[4].Value.ToString();
textBoxdia.Text = metroGridtxt.Rows[e.RowIndex].Cells[5].Value.ToString();
textBoxtip.Text = metroGridtxt.Rows[e.RowIndex].Cells[6].Value.ToString();
}
private void Reserv_Load_1(object sender, EventArgs e)
{
User_account.Text = ((Form)this.Main).Controls["User_account"].Text;
}
}
```

Код класу Report_Year

```

public partial class Report_Year : MetroFramework.Forms.MetroForm
{
    public Report_Year()
    {
        InitializeComponent();
    }

    public Main Main { get; internal set; }
    private void Report_Year_Load(object sender, EventArgs e)
    {
        User_account.Text = ((Form)this.Main).Controls["User_account"].Text;
    }

    private void User_account_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
    {
        DialogResult = DialogResult.OK;
    }
}

```

Код класу Report_Month

```

public partial class Report_Month : MetroFramework.Forms.MetroForm
{
    ReportDocument cryprt = new ReportDocument();
    public Report_Month()
    {
        InitializeComponent();
    }
    public Main Main { get; internal set; }
    private void Report_Month_Load(object sender, EventArgs e)
    {
        User_account.Text = ((Form)this.Main).Controls["User_account"].Text;
    }

    private void Openrep_Click(object sender, EventArgs e)
    {
        cryprt.Load(@"MonthReport.rpt");
        SqlConnection openacc = new SqlConnection(@"Data Source=WIN-CTJO8OTQQ5Z\SQLEXPRESS;Initial
        Catalog=CMMS;Integrated Security=True");
        openacc.Open();
        DataSet dst = new DataSet();
        SqlDataAdapter sda = new SqlDataAdapter(@"Select * FROM dbo.[Журнал_TO_Місяць] where [Expr1] between '" +
        dateTimePicker1.Value + "' and '" + dateTimePicker2.Value + "'", openacc);
        sda.Fill(dst, "Журнал_TO_Місяць");
        cryprt.SetDataSource(dst);
        cryprt.SetParameterValue("@FromDate", dateTimePicker1.Value);
        cryprt.SetParameterValue("@ToDate", dateTimePicker2.Value);
        crystalReportMonth.ReportSource = cryprt;
    }
    private void User_account_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
    {
        DialogResult = DialogResult.OK;
    }
}

```


Продовження дод. Л

Код класу WorkOrder

```
public partial class Work_order : MetroFramework.Forms.MetroForm
{
    ReportDocument cryprt = new ReportDocument();
    public Work_order()
    {
        InitializeComponent();
    }
    public Main Main { get; internal set; }
    private void Work_order_Load(object sender, EventArgs e)
    {
        User_account.Text = ((Form)this.Main).Controls["User_account"].Text;
    }

    private void Openorder_Click(object sender, EventArgs e)
    {
        cryprt.Load(@"WorkOrder.rpt");
        SqlConnection openacc = new SqlConnection(@"Data Source=WIN-CTJO8OTQQ5Z\SQLEXPRESS;Initial
        Catalog=CMMS;Integrated Security=True");
        openacc.Open();
        DataSet dst = new DataSet();
        SqlDataAdapter sda = new SqlDataAdapter(@"Select * FROM dbo.[Наряд_на_виконання_робіт] where [Дата
        обслуговування] = " + dateTimePicker1.Value + " ",openacc);
        sda.Fill(dst, "Наряд_на_виконання_робіт");
        // SqlDataAdapter sda = new SqlDataAdapter(@"Select * FROM dbo.[Наявне устаткування] where [Інв_№] = " +
        textBox1.Text + " ",openacc);
        // sda.Fill(dst, "Наявне устаткування");
        cryprt.SetDataSource(dst);
        cryprt.SetParameterValue("@invent", dateTimePicker1.Value);
        //cryprt.SetParameterValue("@invent", textBox1.Text);
        crystalReportWork.ReportSource = cryprt;
    }
    private void User_account_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
    {
        DialogResult = DialogResult.OK;
    }
}
```

Запит до бази даних Річний_графік_TO

```

SELECT    MONTH(dbo.[Наявне устаткування].[Дата обслуговування]) AS [Місяць TO],
          dbo.[Встановлене устаткування].[Код об'єкту] + '-' + dbo.[Встановлене устаткування].Позиція AS [Позиція у графіку],
          dbo.[Тип устаткування].Найменування, dbo.[Тип устаткування].Тип, dbo.[Наявне устаткування].[Дата обслуговування],
          dbo.[Тип устаткування].Цикл_KP, dbo.[Тип устаткування].Цикл_TP, dbo.[Тип устаткування].Цикл_TO, CASE WHEN [Вид
ремонту] IS NOT NULL THEN " + [Вид ремонту] + " + RIGHT([Дата обслуговування], 20)

ELSE [Вид ремонту] END AS Останній_ремонт, CASE WHEN [Тип устаткування].[Цикл_TO] <> 0 THEN str([Тип
устаткування].[Цикл_TO], 2, 2) + ' - ' + CASE WHEN [Тип устаткування].Цикл_TP <> 0 THEN str([Тип устаткування]. ' END
END END AS Планове_TO, CASE Year(DateAdd(MM, [Цикл_TP], [Дата обслуговування]))
  WHEN 1 THEN 'TP' END ELSE CASE Month(DateAdd(MM, [Цикл_TO], [Дата обслуговування])) WHEN 1 THEN 'TO'
ELSE CASE Month(DateAdd(MM, 2 * [Цикл_TO], [Дата обслуговування])) WHEN 1 THEN 'TO' ELSE CASE
Month(DateAdd(MM, 3 * [Цикл_TO], [Дата обслуговування])) WHEN 1 THEN 'TO' ELSE CASE Month(DateAdd(MM, 4 *
[Цикл_TO], [Дата обслуговування]))
  WHEN 1 THEN 'TO' ELSE " END END END END AS [1], CASE Year(DateAdd(MM, [Цикл_TP], [Дата
обслуговування])) WHEN 2 THEN 'TP' END ELSE CASE Month(DateAdd(MM, [Цикл_TO], [Дата обслуговування])) WHEN 2
THEN 'TO' ELSE CASE Month(DateAdd(MM, 2 * [Цикл_TO], [Дата обслуговування]))
  WHEN 2 THEN 'TO' ELSE CASE Month(DateAdd(MM, 3 * [Цикл_TO], [Дата обслуговування])) WHEN 2 THEN 'TO'
ELSE CASE Month(DateAdd(MM, 4 * [Цикл_TO],
[Дата обслуговування])) WHEN 2 THEN 'TO' ELSE " END END END END AS [2], CASE Year(DateAdd(MM,
[Цикл_TP], [Дата обслуговування]))
  WHEN 3 THEN 'TP' END ELSE CASE Month(DateAdd(MM, [Цикл_TO], [Дата обслуговування])) WHEN 3 THEN 'TO'
ELSE CASE Month(DateAdd(MM, 2 * [Цикл_TO], [Дата обслуговування]))
  WHEN 3 THEN 'TO' ELSE CASE Month(DateAdd(MM, 3 * [Цикл_TO], [Дата обслуговування])) WHEN 3 THEN 'TO'
ELSE CASE Month(DateAdd(MM, 4 * [Цикл_TO],
[Дата обслуговування])) WHEN 3 THEN 'TO' ELSE " END END END END AS [3], CASE Year(DateAdd(MM,
[Цикл_TP], [Дата обслуговування]))
  WHEN 4 THEN 'TP' END ELSE CASE Month(DateAdd(MM,
[Цикл_TO], [Дата обслуговування])) WHEN 4 THEN 'TO' ELSE CASE Month(DateAdd(MM, 2 * [Цикл_TO], [Дата
обслуговування]))
  WHEN 4 THEN 'TO' ELSE CASE Month(DateAdd(MM, 3 * [Цикл_TO], [Дата обслуговування])) WHEN 4 THEN 'TO'
ELSE CASE Month(DateAdd(MM, 4 * [Цикл_TO],
[Дата обслуговування])) WHEN 4 THEN 'TO' ELSE " END END END END AS [4], CASE Year(DateAdd(MM,
[Цикл_TP], [Дата обслуговування]))
  WHEN 5 THEN 'TP' END ELSE CASE Month(DateAdd(MM,
[Цикл_TO], [Дата обслуговування])) WHEN 5 THEN 'TO' ELSE CASE Month(DateAdd(MM, 2 * [Цикл_TO], [Дата
обслуговування]))
  WHEN 5 THEN 'TO' ELSE CASE Month(DateAdd(MM, 3 * [Цикл_TO], [Дата обслуговування])) WHEN 5 THEN 'TO'
ELSE CASE Month(DateAdd(MM, 4 * [Цикл_TO],
[Дата обслуговування])) WHEN 5 THEN 'TO' ELSE " END END END END AS [5], CASE Year(DateAdd(MM,
[Цикл_TP], [Дата обслуговування]))
  WHEN 6 THEN 'TP' END ELSE CASE Month(DateAdd(MM,
[Цикл_TO], [Дата обслуговування])) WHEN 6 THEN 'TO' ELSE CASE Month(DateAdd(MM, 2 * [Цикл_TO], [Дата
обслуговування]))
  WHEN 6 THEN 'TO' ELSE CASE Month(DateAdd(MM, 3 * [Цикл_TO], [Дата обслуговування])) WHEN 6 THEN 'TO'
ELSE CASE Month(DateAdd(MM, 4 * [Цикл_TO],
[Дата обслуговування])) WHEN 6 THEN 'TO' ELSE " END END END END AS [6], CASE Year(DateAdd(MM,
[Цикл_TP], [Дата обслуговування]))
  WHEN 7 THEN 'TP' END ELSE CASE Month(DateAdd(MM,
[Цикл_TO], [Дата обслуговування])) WHEN 7 THEN 'TO' ELSE CASE Month(DateAdd(MM, 2 * [Цикл_TO], [Дата
обслуговування]))
  WHEN 7 THEN 'TO' ELSE CASE Month(DateAdd(MM, 3 * [Цикл_TO], [Дата обслуговування])) WHEN 7 THEN 'TO'
ELSE CASE Month(DateAdd(MM, 4 * [Цикл_TO],
[Дата обслуговування])) WHEN 7 THEN 'TO' ELSE " END END END END AS [7], CASE Year(DateAdd(MM,
[Цикл_TP], [Дата обслуговування]))
  WHEN 8 THEN 'TP' END ELSE CASE Month(DateAdd(MM,
[Цикл_TO], [Дата обслуговування])) WHEN 8 THEN 'TO' ELSE CASE Month(DateAdd(MM, 2 * [Цикл_TO], [Дата
обслуговування]))
  WHEN 8 THEN 'TO' ELSE CASE Month(DateAdd(MM, 3 * [Цикл_TO], [Дата обслуговування])) WHEN 8 THEN 'TO'
ELSE CASE Month(DateAdd(MM, 4 * [Цикл_TO]
[Дата обслуговування])) WHEN 8 THEN 'TO' ELSE " END END END END AS [8], CASE Year(DateAdd(MM,
[Цикл_TP], [Дата обслуговування]))
  WHEN 9 THEN 'TP' END ELSE CASE Month(DateAdd(MM,
[Цикл_TO], [Дата обслуговування])) WHEN 9 THEN 'TO' ELSE CASE Month(DateAdd(MM, 2 * [Цикл_TO], [Дата
обслуговування]))),

```

Продовження дод. М

```
WHEN 9 THEN 'TO' ELSE CASE Month(DateAdd(ММ, 3 * [Цикл_TO], [Дата обслуговування])) WHEN 9 THEN 'TO'
ELSE CASE Month(DateAdd(ММ, 4 * [Цикл_TO],
[Дата обслуговування])) WHEN 9 THEN 'TO' ELSE " END END END END END AS [9], CASE Year(DateAdd(ММ,
[Цикл_TP], [Дата обслуговування]))
WHEN 10 THEN 'TP' END ELSE CASE Month(DateAdd(ММ,
[Цикл_TO], [Дата обслуговування])) WHEN 10 THEN 'TO' ELSE CASE Month(DateAdd(ММ, 2 * [Цикл_TO], [Дата
обслуговування]))
WHEN 10 THEN 'TO' ELSE CASE Month(DateAdd(ММ, 3 * [Цикл_TO], [Дата обслуговування])) WHEN 10 THEN
'TO' ELSE CASE Month(DateAdd(ММ, 4 * [Цикл_TO],
[Дата обслуговування])) WHEN 10 THEN 'TO' ELSE " END END END END END AS [10], CASE Year(DateAdd(ММ,
[Цикл_TP], [Дата обслуговування]))
WHEN 11 THEN 'TP' END ELSE CASE Month(DateAdd(ММ,
[Цикл_TO], [Дата обслуговування])) WHEN 11 THEN 'TO' ELSE CASE Month(DateAdd(ММ, 2 * [Цикл_TO], [Дата
обслуговування]))
WHEN 11 THEN 'TO' ELSE CASE Month(DateAdd(ММ, 3 * [Цикл_TO], [Дата обслуговування])) WHEN 11 THEN '
TO' ELSE CASE Month(DateAdd(ММ, 4 * [Цикл_TO],
[Дата обслуговування])) WHEN 11 THEN 'TO' ELSE " END END END END END AS [11], CASE Year(DateAdd(ММ,
[Цикл_TP], [Дата обслуговування]))
WHEN 12 THEN 'TP' END ELSE CASE Month(DateAdd(ММ,
[Цикл_TO], [Дата обслуговування])) WHEN 12 THEN 'TO' ELSE CASE Month(DateAdd(ММ, 2 * [Цикл_TO], [Дата
обслуговування]))
WHEN 12 THEN 'TO' ELSE CASE Month(DateAdd(ММ, 3 * [Цикл_TO], [Дата обслуговування])) WHEN 12 THEN '
TO' ELSE CASE Month(DateAdd(ММ, 4 * [Цикл_TO], [Дата обслуговування]))
WHEN 12 THEN 'TO' ELSE " END END END END END AS [12], dbo.[Тип устаткування].Норма_TP, dbo.[Тип
устаткування].Норма_TO
```

```
FROM      dbo.[Тип устаткування] INNER JOIN dbo.[Наявне устаткування] ON dbo.[Тип устаткування].
Код_типу = dbo.[Наявне устаткування].[Код типу] INNER JOIN dbo.[Встановлене устаткування]
ON dbo.[Наявне устаткування].[Инв_№] = dbo.[Встановлене устаткування].[Инв_№]
```

```
WHERE      (dbo.[Встановлене устаткування].[Не обслуговується] <> - 1)
```

Запит до бази даних Журнал_TO_Місяць

```
SELECT      TOP (100) PERCENT dbo.[Тип устаткування].Найменування, dbo.[Тип устаткування].Тип,
dbo.[Тип устаткування].Норма_TP, dbo.[Тип устаткування].Норма_TO,
dbo.[Встановлене устаткування].Приміщення, dbo.[Встановлене устаткування].Позиція,
DATEADD(М, 1, dbo.[Наявне устаткування].[Дата обслуговування])
AS Expr1, dbo.[Наявне устаткування].[Вид ремонту], dbo.[Встановлене устаткування].[Не обслуговується]
FROM      dbo.[Встановлене устаткування] INNER JOIN
dbo.[Наявне устаткування] ON dbo.[Встановлене устаткування].[Инв_№] =
dbo.[Наявне устаткування].[Инв_№] INNER JOIN
dbo.[Тип устаткування] ON dbo.[Наявне устаткування].[Код типу] = dbo.[Тип устаткування].Код_типу
WHERE      (dbo.[Встановлене устаткування].[Не обслуговується] = 0)
ORDER BY   dbo.[Встановлене устаткування].Приміщення
```

Запит до бази даних Наряд_на_виконання_робіт

```
SELECT      dbo.[Встановлене устаткування].[Найменування позиції],
dbo.[Встановлене устаткування].Позиція, dbo.[Встановлене устаткування].[Инв_№],
dbo.[Встановлене устаткування].[Код об'єкту], dbo.[Наявне устаткування].[Дата обслуговування]
FROM      dbo.[Встановлене устаткування] INNER JOIN
dbo.[Наявне устаткування] ON dbo.[Встановлене устаткування].[Инв_№] = dbo.[Наявне устаткування].[Инв_№]
```