

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Навчально-науковий центр заочної форми навчання

Кафедра КІТАР

Рівень вищої освіти перший (бакалаврський)

Спеціальність 151 Автоматизація та комп'ютерно-інтегровані технології

Тип програми Освітньо-професійна

Освітня програма Автоматизація та комп'ютерно-інтегровані технології

(шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри КІТАР

(підпис)

«___» _____ 2024 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Масічу Максиму Вадимовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Розробка системи автоматизації процесу зберігання та видачі електричних інструментів

Затверджена наказом по університету від 03.06.2024 р. № 544 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 12.06.2024 р.

3. Вихідні дані до роботи 3.1 Операційна система – Windows 10

3.2 Об'єм відеопам'яті – 6 ГБ, об'єм оперативної пам'яті – 16 ГБ

3.3 Мова написання програми – C#

4. Перелік питань, що потрібно опрацювати в роботі 4.1 Вступ

4.2 Аналіз технічного завдання

4.3 Розробка та верифікація програмного забезпечення

4.4 Висновки та перелік джерела посилань

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій

Презентація PowerPoint на 15 слайдів у форматі .pptx;

6. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз технічного завдання	11.05.2024	
2	Виконання розділу 1	17.05.2024	
3	Виконання розділу 2	26.05.2024	
4	Розробка програмного забезпечення	10.05 – 01.06.24	
5	Виконання розділу 3	01.06.2024	
6	Подання роботи на перевірку на плагіат	11.06.2024	
7	Оформлення пояснювальної записки	11.06.2024	
8	Подання роботи на рецензію	12.06.2024	
9	Подання роботи на підпис зав. кафедри	12.06.2024	
10	Подання атестаційної роботи в ЕК	12.06.2024	

Дата видачі завдання 11.05.2024 р.

Студент _____ Масіч М. В.

(підпис)

Керівник роботи _____ доц. Замірець О.М.

(підпис)

(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 83 с., 14 табл., 43 рис., 2 дод., 20 джерел.

АВТОМАТИЗАЦІЯ, ЗБЕРІГАННЯ, ВИДАЧА, ЕЛЕКТРИЧНІ
ІНСТРУМЕНТИ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, УПРАВЛІННЯ.

Об'єкт розробки: процес зберігання та видачі електричних інструментів у технічних та ремонтних підрозділах.

Мета роботи – розробка системи автоматизації процесу зберігання та видачі електричних інструментів для забезпечення ефективного обліку, контролю, обслуговування та звітності щодо використання інструментів.

Методи розробки – забезпечення ефективного обліку, контролю, обслуговування та звітності щодо використання інструментів.

В кваліфікаційній роботі розглянуто актуальні питання автоматизації обліку електричних інструментів, запропоновані рішення для вдосконалення процесів зберігання та видачі. Розроблено програмне забезпечення, яке включає модулі для реєстрації, редагування, видалення та обслуговування інструментів, а також механізми авторизації користувачів і формування різноманітних звітів. Проведено тестування системи, що підтвердило її функціональність та надійність.

Розроблена система може використовуватись у різних галузях, де необхідно забезпечити ефективне управління електричними інструментами, а також у навчальному процесі для підготовки фахівців у сфері автоматизації та управління.

ABSTRACT

Explanatory note: 83 pp., 14 tables, 43 figures, 2 appendices, 20 sources.

AUTOMATION, STORAGE, ISSUING, POWER TOOLS, SOFTWARE, MANAGEMENT.

The object of the research: the process of storing and issuing electric tools in technical and repair units.

The purpose of the work is to develop a system for automating the process of storing and issuing electric tools to increase the efficiency of resource management and reduce operating costs.

Research methods – analysis of modern automated accounting systems, software design, modular testing and experimental verification of system functionality.

In the certification work, the actual issues of automating the accounting of electric tools were considered, solutions were proposed for improving the storage and issuing processes. Software has been developed that includes modules for registration, editing, deletion and maintenance of tools, as well as mechanisms for user authorization and generation of various reports. The system was tested, which confirmed its functionality and reliability.

The developed system can be used in various industries, where it is necessary to ensure effective control of electric tools, as well as in the educational process for training specialists in the field of automation and control.

ЗМІСТ

Перелік умовних позначень та скорочень	8
Вступ.....	9
1 Аналіз сучасних засобів та систем автоматизованого обліку.....	12
1.1 Аналіз предметної області	12
1.2 Аналіз сучасних систем автоматизованого обліку.....	16
1.3 Постановка задачі.....	21
1.4 Висновок.....	24
2 Вибір технологій, проектування та особливості архітектури програмного забезпечення.....	25
2.1 Вибір технологій	25
2.2 Аналіз вимог. Use-case діаграми. Основні прецеденти.....	28
2.3 Проектування бази даних	33
2.4 Архітектура проекту	37
2.5 Висновок.....	46
3 Розробка та верифікація програмного забезпечення	47
3.1 Розробка основних алгоритмів	47
3.2 Розробка програмних модулів системи	51
3.3 Експериментальна перевірка та тестування	59
3.4 Інструкція користувача	64
3.5 Висновок.....	76
4 Автоматизація системи	78
4.1 Технічні аспекти автоматизації.....	78
4.2 Переваги автоматизованого обліку.....	79
4.3 Інтеграція системи RFID у програму	80
5 Охорона праці.....	82
5.1 Загальні питання охорони праці	82
5.2 Вплив автоматизації на безпеку праці.....	82
5.3 Стандарти безпеки при інсталяції та роботі автоматизованих систем.....	83

Висновки.....	84
Перелік джерел посилань	86
Додаток А Скрипт бази даних	89
Додаток Б Лістинги програми	93
Додаток В Демонстраційний матеріал у виді презентації	107
Додаток Г Відомість кваліфікаційної роботи	108

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

БД – база даних

СУБД – система управління базами даних

C# – об'єктно-орієнтована мова програмування, розроблена корпорацією Microsoft, використовується для створення різноманітних бізнес-додатків.

DBMS (Database Management System) – система управління базами даних.

ERP (Enterprise Resource Planning) – планування ресурсів підприємства.

GigaTrak Tool Tracking – система управління інструментами для підприємств.

IoT (Internet of Things) – Інтернет речей, мережа фізичних об'єктів, які взаємодіють між собою через Інтернет.

MS SQL Server – реляційна система управління базами даних, розроблена корпорацією Microsoft.

RFID (Radio Frequency Identification) – радіочастотна ідентифікація.

SQL (Structured Query Language) – мова структурованих запитів для роботи з базами даних.

UC (Use Case) – варіант використання системи, описує конкретний сценарій взаємодії користувача з системою.

ВСТУП

У сучасному світі ефективне управління ресурсами є ключовим фактором успіху будь-якої організації, зокрема у сфері технічного обслуговування та ремонту. Одним з важливих аспектів цього процесу є зберігання та видача електричних інструментів, які використовуються для виконання різноманітних завдань. Незважаючи на те, що багато компаній вже впровадили різні методи для організації даного процесу, залишаються певні проблеми та виклики, які потребують подальшого вдосконалення.

Сучасний стан проблеми свідчить про наявність значної кількості ручних операцій при обліку та видачі інструментів, що призводить до підвищення ризику людських помилок, втрат обладнання та неефективного використання ресурсів. Часто відсутність централізованого контролю та автоматизованих систем обліку ускладнює моніторинг стану інструментів, їх своєчасне обслуговування та ремонт, що, у свою чергу, може призвести до затримок у виконанні робіт та підвищення експлуатаційних витрат.

Світові тенденції у вирішенні цієї проблеми орієнтовані на впровадження автоматизованих систем управління ресурсами, які дозволяють значно підвищити ефективність процесів зберігання та видачі інструментів. Використання технологій штучного інтелекту, інтернету речей (IoT) та систем управління базами даних (DBMS) дозволяє створювати інтелектуальні системи, які автоматично відстежують місцезнаходження інструментів, контролюють їх технічний стан, планують обслуговування та забезпечують своєчасне поповнення запасів. Такі системи також сприяють зниженню експлуатаційних витрат та підвищенню рівня безпеки на виробництві.

Актуальність теми кваліфікаційної роботи зумовлена потребою в розробці сучасних рішень для автоматизації процесу зберігання та видачі електричних інструментів. Це не лише дозволить підвищити ефективність роботи організацій, але й сприятиме зниженню витрат, підвищенню рівня безпеки та зменшенню

ризиків втрат інструментів. Впровадження автоматизованих систем є важливим кроком на шляху до побудови інтелектуальних виробничих процесів, що відповідають вимогам сучасного ринку.

Мета роботи: розробка системи автоматизації процесу зберігання та видачі електричних інструментів для підвищення ефективності управління ресурсами та зменшення операційних витрат.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- виконати аналіз сучасних методів, засобів та систем автоматизованого обліку, включаючи аналіз предметної області та існуючих систем автоматизованого обліку, для визначення переваг та недоліків існуючих рішень;

- обрати відповідні технології, спроектувати базу даних та розробити архітектуру програмного забезпечення, з урахуванням вимог до системи, для забезпечення надійності, масштабованості та зручності використання;

- розробити та верифікувати програмне забезпечення, включаючи створення основних алгоритмів, програмних модулів системи, а також проведення експериментальної перевірки та тестування;

- розробити інструкцію користувача для забезпечення ефективного впровадження та використання системи, а також для забезпечення підтримки та навчання користувачів у процесі експлуатації програмного забезпечення.

Об'єкт розробки: процес зберігання та видачі електричних інструментів у технічних та ремонтних підрозділах, що включає організацію обліку, контролю, збереження та використання інструментів для забезпечення безперебійної роботи та ефективного управління ресурсами.

Предмет розробки: методи та технології автоматизації обліку і управління зберіганням та видачею електричних інструментів, які забезпечують підвищення точності обліку, зменшення витрат на управління ресурсами, підвищення безпеки та зниження ризику втрат інструментів через людські помилки або недбалість.

Кваліфікаційну роботу оформлено згідно з ДСТУ 3008:2015 [1], а також з рекомендаціями з підготовки і оформлення кваліфікаційної роботи здобувачами першого (бакалаврського) рівня вищої освіти [2,3].

1 АНАЛІЗ СУЧАСНИХ ЗАСОБІВ ТА СИСТЕМ АВТОМАТИЗОВАНОГО ОБЛІКУ

1.1 Аналіз предметної області

Особливості обліку електричних інструментів вимагають ретельного підходу до організації та управління цим процесом. Електричні інструменти є важливими ресурсами для технічних та ремонтних підрозділів, і їх правильне зберігання та видача безпосередньо впливають на ефективність виконання робіт. Відсутність належного обліку може призвести до низки проблем, таких як втрата інструментів, їх пошкодження або неналежне використання.

Зберігання електричних інструментів повинно бути організоване таким чином, щоб забезпечити швидкий доступ до необхідного обладнання, мінімізуючи час на його пошук. Для цього можуть використовуватись спеціальні стелажі, шафи або контейнери, обладнані засобами для маркування та ідентифікації інструментів. Важливо також забезпечити умови, що запобігають пошкодженню інструментів, такі як контроль вологості та температури в приміщеннях для зберігання.

Ефективний облік електричних інструментів включає не лише фіксацію їх місцезнаходження, але й контроль за станом та строками обслуговування. Ведення журналів або використання електронних таблиць для цього завдання часто є недостатньо надійним через високу ймовірність людських помилок. Тому все більше підприємств звертаються до автоматизованих систем управління, які дозволяють централізовано зберігати та обробляти дані про інструменти.

На рис. 1.1 представлено діаграму розподілу основних особливостей зберігання електричних інструментів.



Рисунок 1.1 – Особливості зберігання електричних інструментів

Важливими аспектами є зберігання інструментів є:

- безпека зберігання. Інструменти повинні зберігатися в умовах, які виключають можливість їх пошкодження або крадіжки;
- організація простору. Зберігання повинно бути організоване так, щоб забезпечити швидкий доступ до необхідних інструментів;
- облік і контроль. Важливо мати точний облік усіх інструментів для запобігання їх втраті та забезпечення своєчасного обслуговування.

Основні виклики, з якими стикаються підприємства при управлінні електричними інструментами, включають декілька ключових аспектів. Перш за все, використання ручного обліку, що часто здійснюється за допомогою паперових журналів або електронних таблиць, призводить до підвищеної ймовірності помилок і втрати даних. Це ускладнює процес відстеження інструментів, їхнього стану та наявності. По-друге, неправильне використання або розміщення інструментів без централізованого контролю може призвести до неефективного використання ресурсів і втрат обладнання. По-третє, відсутність належного контролю за станом інструментів ускладнює своєчасне проведення

технічного обслуговування та ремонтів, що підвищує ризик несправностей і нещасних випадків.

Для детальнішого розуміння проблематики потрібно розглянути деякі аспекти управління інструментами, які є важливими для побудови ефективної системи автоматизації. На рис. 1.2 представлено діаграму, яка допоможе візуалізувати основні виклики в управлінні інструментами.



Рисунок 1.2 – Основні виклики в управлінні інструментами

Існує кілька основних методів ведення обліку електричних інструментів, кожен з яких має свої переваги та недоліки. Ручний облік, що здійснюється за допомогою паперових журналів або електронних таблиць, є простим у використанні, але має високу ймовірність людських помилок та трудомісткість процесу. Штрихкодний облік використовує штрихкоди для ідентифікації інструментів та забезпечує швидке зчитування даних, однак потребує початкових інвестицій у обладнання. RFID-облік пропонує безконтактне зчитування даних і високу точність, проте вимагає значних витрат на впровадження технології. Автоматизовані системи обліку, такі як ERP-системи, дозволяють централізовано управляти даними і інтегрувати їх з іншими бізнес-процесами, але є дорогими у впровадженні та обслуговуванні. Хмарні системи обліку забезпечують

доступність з будь-якої точки світу та зменшують витрати на інфраструктуру, але залежать від якості інтернет-з'єднання і мають потенційні ризики безпеки даних. На рис. 1.3 представлено діаграму розподілу ефективних методів ведення обліку електричних інструментів.

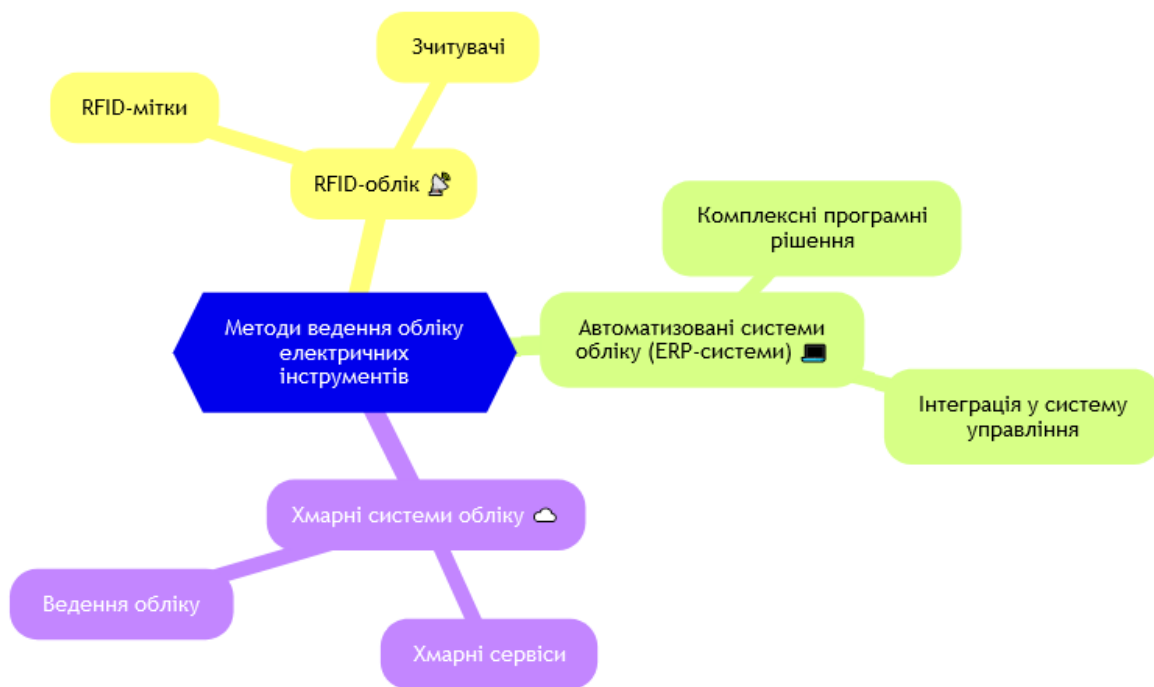


Рисунок 1.3 – Основні методи ведення обліку електричних інструментів

Аналіз предметної області показує, що особливості обліку електричних інструментів вимагають ретельної організації процесу зберігання та видачі. Безпечне зберігання, швидкий доступ до інструментів та точний облік їхнього стану є критичними аспектами, що впливають на ефективність виробничих і ремонтних підрозділів [4]. Використання автоматизованих систем обліку стає необхідністю для мінімізації людських помилок, втрат обладнання та забезпечення своєчасного технічного обслуговування. Дослідження сучасних методів та технологій, таких як системи маркування та автоматизовані облікові системи, допоможе створити більш ефективні рішення для управління електричними інструментами.

1.2 Аналіз сучасних систем автоматизованого обліку

Аналіз аналогічних програмних рішень є важливим етапом у розробці власної системи автоматизації обліку електричних інструментів. Це дозволяє виявити сучасні тенденції, ефективні підходи та технології, що використовуються у вже існуючих рішеннях, а також визначити їх сильні та слабкі сторони. Вивчення наявних систем допомагає уникнути поширених помилок, оптимізувати процес розробки та забезпечити конкурентоспроможність майбутнього продукту. Основна мета цього аналізу – оцінити функціональні можливості, зручність використання, масштабованість та надійність існуючих рішень, що дозволить сформулювати вимоги до власної системи та визначити шляхи її подальшого вдосконалення.

GigaTrak Tool Tracking призначений для ефективного управління електричними інструментами та обладнанням на підприємствах (рис. 1.4). Цей застосунок дозволяє автоматизувати процеси обліку, зберігання, видачі та повернення інструментів, знижуючи ризик їх втрати або неправильного використання. Система забезпечує точне відстеження місцезнаходження інструментів, їх стану та історії обслуговування, що сприяє підвищенню продуктивності та безпеки праці [4].

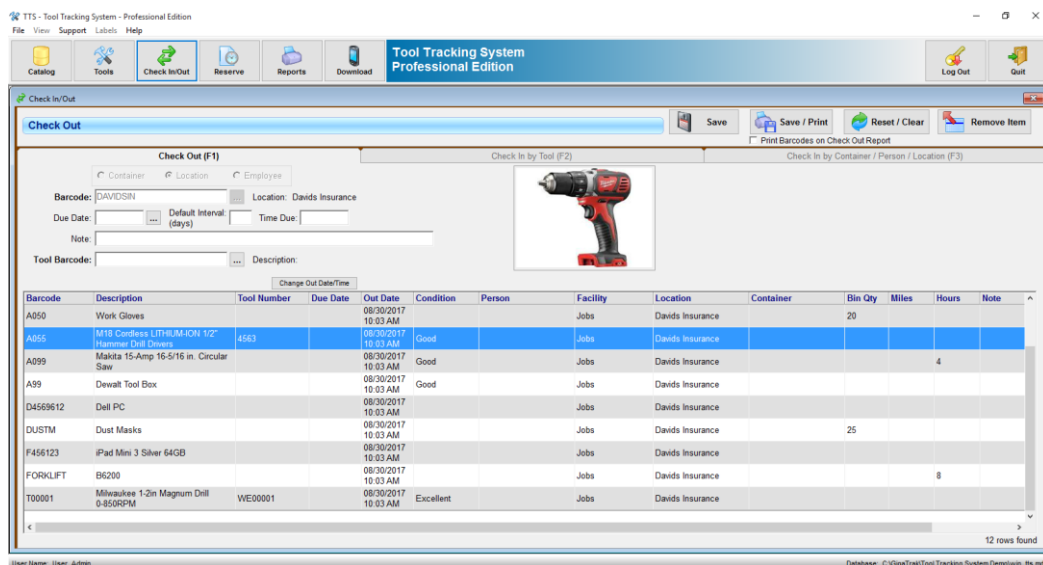


Рисунок 1.4 – Приклад інтерфейсу «GigaTrak Tool Tracking»

Система побудована на основі клієнт-серверної архітектури, що включає центральну базу даних для зберігання інформації про інструменти та сервер, який обробляє запити клієнтів. Клієнтська частина системи представлена у вигляді веб-інтерфейсу та мобільних додатків, що забезпечує доступ до функціоналу з будь-якого пристрою з підключенням до інтернету. Система використовує штрихкоди або RFID-мітки для ідентифікації інструментів, а сканери та зчитувачі забезпечують швидке збирання даних. Інтерфейс користувача дозволяє легко управляти інструментами, створювати звіти та контролювати їх переміщення. Інтеграція з іншими системами управління підприємством можлива через API, що забезпечує безперервний обмін даними та покращує загальну ефективність управління ресурсами [5].

Переваги:

- точне відстеження місцезнаходження та стану інструментів;
- автоматизація процесів обліку та управління інструментами;
- доступність з будь-якого пристрою завдяки веб-інтерфейсу та мобільним додаткам;
- інтеграція з іншими системами управління через API.

Недоліки:

- високі початкові витрати на впровадження технології;
- потреба в навчанні персоналу для роботи з системою;
- залежність від якості інтернет-з'єднання для доступу до функціоналу;
- потенційні ризики безпеки даних при використанні хмарних сервісів [6].

Отже, GigaTrak Tool Tracking є потужним інструментом для автоматизації управління електричними інструментами, що забезпечує високу точність обліку та ефективність процесів. Використання штрихкодів або RFID-міток дозволяє значно знизити ризик втрат та неправильного використання інструментів. Незважаючи на певні недоліки, такі як високі початкові витрати та потреба в навчанні персоналу, система пропонує значні переваги, які можуть суттєво покращити управління ресурсами на підприємстві.

Asset Panda призначений для управління активами, включаючи електричні інструменти, обладнання та інші ресурси підприємства (рис. 1.5). Цей застосунок допомагає відстежувати місцезнаходження, стан та історію використання активів, забезпечуючи точний облік і контроль за їхнім використанням. Система дозволяє автоматизувати процеси інвентаризації, обслуговування та аудиту, що підвищує ефективність управління ресурсами і знижує операційні витрати [7].

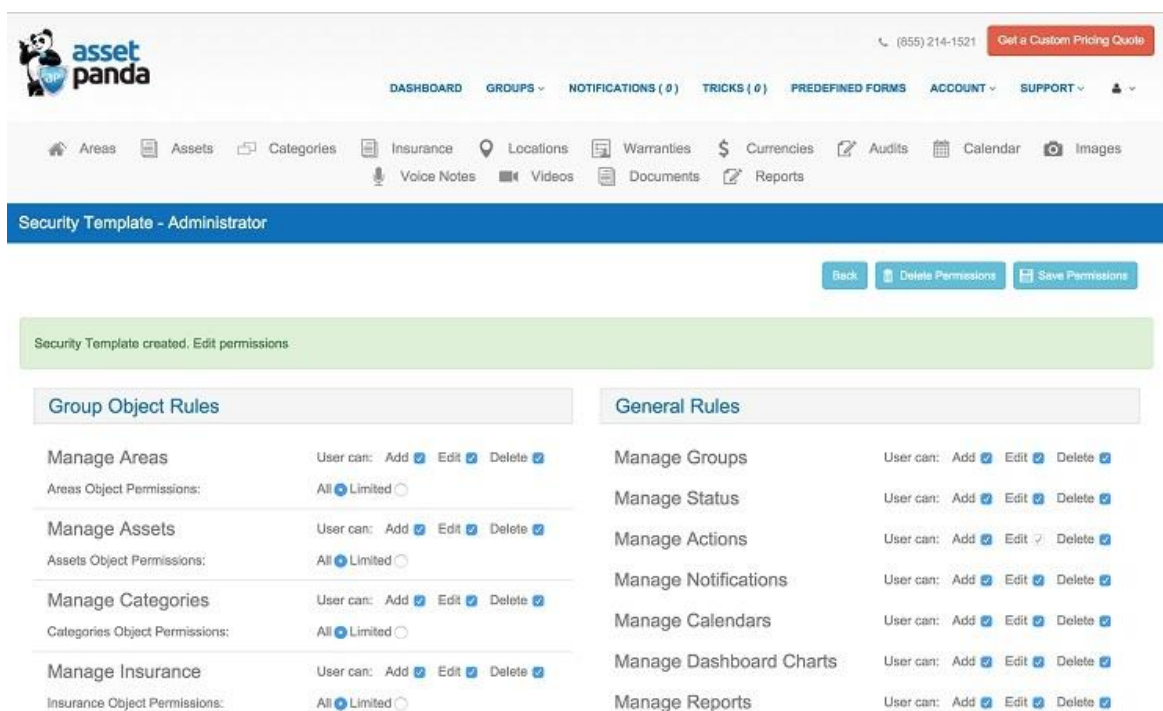


Рисунок 1.5 – Приклад інтерфейсу «Asset Panda»

Опис архітектури: Asset Panda побудована на основі хмарної архітектури, що забезпечує доступ до системи з будь-якого пристрою з підключенням до інтернету. Центральна база даних зберігає всю інформацію про активи, а сервери обробляють запити клієнтів. Клієнтська частина представлена у вигляді веб-інтерфейсу та мобільних додатків, що дозволяє користувачам зручно управляти активами в режимі реального часу. Система використовує штрихкоди, QR-коди та RFID-мітки для ідентифікації активів, що забезпечує швидке та точне збирання даних. Інтерфейс користувача є інтуїтивно зрозумілим і легко налаштовується під потреби підприємства. Інтеграція з іншими бізнес-системами можлива через API,

що дозволяє забезпечити безперервний обмін даними та інтегрувати управління активами з іншими бізнес-процесами [8].

Переваги:

- хмарна архітектура забезпечує доступ до системи з будь-якого пристрою з інтернетом;
- підтримка штрихкодів, QR-кодів та RFID-міток для швидкого та точного обліку;
- інтуїтивно зрозумілий інтерфейс, що легко налаштовується під потреби користувача;
- інтеграція з іншими бізнес-системами через API, що забезпечує безперервний обмін даними.

Недоліки:

- залежність від якості інтернет-з'єднання для доступу до системи;
- потенційні ризики безпеки даних при використанні хмарних сервісів;
- регулярні витрати на підписку, що можуть бути значними для великих підприємств;
- потреба в навчанні персоналу для ефективного використання системи [9].

Отже, Asset Panda є потужним інструментом для управління активами, включаючи електричні інструменти та обладнання. Завдяки хмарній архітектурі та підтримці сучасних методів ідентифікації активів, таких як штрихкоди, QR-коди та RFID-мітки, система забезпечує високу точність обліку та зручність використання. Незважаючи на певні недоліки, такі як залежність від інтернет-з'єднання та потреба в навчанні персоналу, Asset Panda пропонує значні переваги, які можуть суттєво покращити управління ресурсами на підприємстві.

Cheqroom призначений для управління інвентарем, включаючи електричні інструменти та обладнання (рис. 1.6). Застосунок допомагає підприємствам відстежувати місцезнаходження, стан та доступність своїх активів у режимі реального часу. Він автоматизує процеси видачі та повернення інструментів, знижуючи ризик їх втрати або неправильного використання. Система дозволяє

створювати детальні звіти та аналітику, що сприяє оптимізації управління ресурсами. Cheqroom забезпечує зручний та інтуїтивно зрозумілий інтерфейс для користувачів, що полегшує процес впровадження та використання [10].

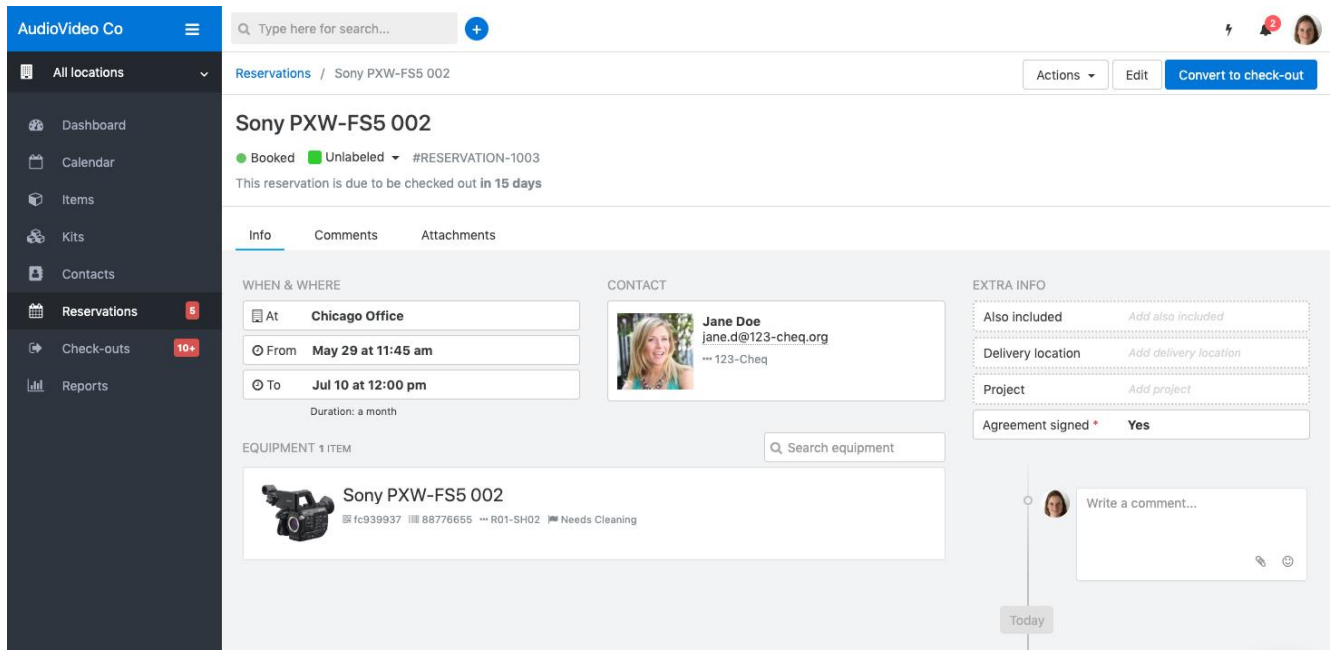


Рисунок 1.6 – Приклад інтерфейсу «Cheqroom»

Cheqroom побудована на основі хмарної архітектури, що забезпечує доступ до системи з будь-якого пристрою з підключенням до інтернету. Центральна база даних зберігає всю інформацію про активи, а сервери обробляють запити користувачів, забезпечуючи високу швидкість та надійність роботи системи. Клієнтська частина представлена у вигляді веб-інтерфейсу та мобільних додатків, що дозволяє користувачам легко керувати активами незалежно від їхнього місцезнаходження [11].

Переваги:

- хмарна архітектура забезпечує доступ до системи з будь-якого пристрою з інтернетом;
- автоматизація процесів видачі та повернення інструментів знижує ризик їх втрати;

- інтуїтивно зрозумілий інтерфейс, що полегшує впровадження та використання системи;

- можливість створення детальних звітів та аналітики для оптимізації управління ресурсами.

Недоліки:

- залежність від якості інтернет-з'єднання для доступу до системи;

- потенційні ризики безпеки даних при використанні хмарних сервісів;

- вартість підписки може бути значною для деяких підприємств [12];

- потреба в навчанні персоналу для ефективного використання системи.

Отже, Cheqroom є потужним інструментом для управління інвентарем, що забезпечує високу точність обліку та ефективність процесів. Хмарна архітектура та підтримка мобільних додатків роблять систему доступною та зручною для користувачів з будь-якого місця. Незважаючи на деякі недоліки, такі як залежність від інтернет-з'єднання та потреба в навчанні персоналу, Cheqroom пропонує значні переваги, які можуть суттєво покращити управління активами на підприємстві.

1.3 Постановка задачі

Основною метою системи є забезпечення ефективного обліку, контролю, обслуговування та звітності щодо використання інструментів. Для досягнення цієї мети необхідно вирішити низку завдань, пов'язаних з управлінням інструментами, довідниками, пошуком, звітністю та системними функціями, зокрема:

а) Управління.

Система повинна забезпечувати ефективне управління інструментами, що включає їх видачу, обслуговування, ведення історії видачі та обслуговування. Це дозволить оптимізувати процеси використання інструментів, зменшити ризик втрат та пошкоджень, а також забезпечити своєчасне технічне обслуговування.

Основні функції управління включають:

- 1) видачу інструментів працівникам та їх повернення;
- 2) ведення записів про технічне обслуговування та ремонти інструментів;
- 3) Зберігання історії всіх операцій з інструментами, що дозволяє відстежувати їх використання та обслуговування;

б) Довідники.

Для забезпечення належного обліку інструментів система повинна містити довідники, що включають інформацію про категорії інструментів, самі інструменти та працівників, які мають доступ до інструментів. Це дозволить організувати дані у зручній формі для подальшого використання та аналізу.

Основні довідники включають:

- 1) категорії інструментів, що дозволяють класифікувати інструменти за типами та призначенням;
- 2) інформацію про кожен інструмент, включаючи назву, інвентарний номер, технічні характеристики та поточний стан;
- 3) дані про працівників, що видають або отримують інструменти, включаючи їхні облікові записи та ролі;

в) Пошук.

Система повинна забезпечувати зручний пошук інструментів за назвою та інвентарним номером. Це дозволить швидко знаходити необхідні інструменти, перевіряти їх доступність та стан, а також здійснювати інші операції з ними. Пошук повинен бути інтуїтивно зрозумілим та швидким, забезпечуючи високу продуктивність користувачів;

г) Звітність.

Для ефективного управління інструментами необхідно мати можливість генерувати різноманітні звіти, що дозволяють аналізувати використання інструментів, витрати на їх обслуговування та інші важливі аспекти. Основні типи звітів включають:

- 1) загальну вартість обслуговування кожного інструменту, що дозволяє контролювати витрати та планувати бюджети;

- 2) використання інструментів за категоріями, що допомагає виявити найбільш популярні та часто використовувані інструменти;
- 3) звіти про інструменти, що видані за певний період, що дозволяє відстежувати їх використання та аналізувати продуктивність працівників;

д) Адміністрування.

Система повинна мати функції для управління обліковими записами користувачів, відстеження подій у системі та персоналізації налаштувань. Це дозволить забезпечити безпеку даних, контроль за доступом та зручність використання. Основні системні функції включають:

- 1) управління обліковими записами користувачів, включаючи створення, редагування та видалення записів;
- 2) відстеження подій у системі, що дозволяє контролювати всі дії користувачів та забезпечувати безпеку даних;
- 3) персоналізацію налаштувань для кожного користувача, що дозволяє налаштовувати систему відповідно до індивідуальних потреб та уподобань;

е) База даних.

База даних є основою системи, яка зберігає всю необхідну інформацію для ефективного управління інструментами. Основні таблиці бази даних включають:

- 1) таблиця «Інструмент», що містить основну інформацію про електричні інструменти, зокрема їх назву, інвентарний номер, технічні характеристики та поточний стан;
- 2) таблиця «Категорія», що містить інформацію про категорії інструментів, що дозволяє класифікувати інструменти за типами та призначенням;
- 3) таблиця «Працівник», що містить інформацію про працівників, які видають або отримують інструменти, включаючи їхні облікові записи та ролі;
- 4) таблиця «Операції з інструментами», що містить інформацію про всі операції з інструментами, такі як видача, повернення або переміщення;
- 5) таблиця «Обслуговування інструментів», що містить інформацію про технічне обслуговування та ремонти інструментів.

Таким чином, постановка задачі передбачає створення комплексної системи автоматизації управління інструментами, яка забезпечить ефективний облік, контроль, обслуговування та звітність щодо використання електричних інструментів на підприємстві.

1.4 Висновок

У рамках даного розділу було проведено детальний аналіз сучасних методів, засобів та систем автоматизованого обліку електричних інструментів. Розглянуто основні особливості зберігання електричних інструментів, включаючи безпеку зберігання, організацію простору та облік і контроль, що представлено на відповідних діаграмах. Виявлено ключові виклики в управлінні інструментами, такі як використання ручного обліку, неправильне використання та відсутність належного контролю. Також проаналізовано основні методи ведення обліку, серед яких штрихкодний облік, автоматизовані системи обліку та хмарні системи обліку, що дозволило визначити їх переваги та недоліки.

Проаналізовано сучасні системи автоматизованого обліку, такі як GigaTrak Tool Tracking, Asset Panda та Cheqroom. Для кожної з цих систем виділено ключові переваги та недоліки, що дозволило отримати глибше розуміння можливостей та обмежень існуючих рішень. На основі цього аналізу сформульовано основні завдання для розробки власної системи автоматизації обліку електричних інструментів, включаючи управління видачею та обслуговуванням інструментів, ведення довідників, забезпечення пошуку, формування звітності та системні функції для управління обліковими записами.

Результати розгляду та аналізу, отримані у даному розділі, дозволяють чітко визначити вимоги до майбутньої системи автоматизації, що будуть використані у наступному розділі для вибору технологій, проектування бази даних та архітектури програмного забезпечення.

2 ВИБІР ТЕХНОЛОГІЙ, ПРОЕКТУВАННЯ ТА ОСОБЛИВОСТІ АРХІТЕКТУРИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Вибір технологій

Вибір технологій є ключовим етапом у процесі розробки системи автоматизації управління зберіганням та видачею електричних інструментів. Від правильного вибору технологій залежить успішність реалізації проекту, продуктивність, масштабованість та зручність використання системи. При виборі технологій необхідно враховувати специфіку задач, вимоги до системи та наявні ресурси.

C# є сучасною об'єктно-орієнтованою мовою програмування, розробленою корпорацією Microsoft. Вона часто використовується для створення різноманітних бізнес-додатків, зокрема на платформі .NET, що забезпечує високу продуктивність і зручність інтеграції з іншими продуктами Microsoft. C# також підтримує розробку веб-додатків, мобільних додатків та ігрових програм завдяки своєму потужному синтаксису та широким можливостям [13-14].

Java є популярною об'єктно-орієнтованою мовою програмування, яка відома своєю платформи-незалежністю завдяки технології Java Virtual Machine (JVM). Ця мова широко використовується для розробки корпоративних додатків, веб-сервісів та мобільних додатків для платформи Android. Java має велику екосистему бібліотек та фреймворків, що дозволяє швидко розробляти складні та масштабовані системи [15].

C++ є потужною мовою програмування, яка поєднує високий рівень абстракції з можливістю низькорівневого управління пам'яттю. Вона використовується для розробки програмного забезпечення, яке потребує високої продуктивності, таких як ігри, системне програмне забезпечення та драйвери. C++ забезпечує високу ефективність виконання та контроль над апаратними ресурсами, що робить її ідеальною для розробки ресурсомістких додатків [16].

У табл. 2.1 проведено порівняльний аналіз мов програмування

Таблиця 2.1 – Порівняльний аналіз мов програмування

Характеристика	C#	Java	C++
Підтримка платформи	.NET (Windows, cross-platform з .NET Core)	JVM (платформо-незалежна)	Крос-платформенна, але залежна від компілятора
Зручність розробки	Висока (простий синтаксис, інструменти)	Середня (простий синтаксис)	Середня (складніший синтаксис)
Розробка бізнес-додатків	Висока	Висока	Середня
Інструментарій та IDE	Visual Studio (потужна IDE з широкими можливостями)	Eclipse, IntelliJ IDEA (потужні IDE)	Visual Studio, Code::Blocks, CLion
Підтримка асинхронного програмування	Вбудована підтримка (async/await)	Вбудована підтримка (з версії 8, CompletableFuture)	Складніше (через бібліотеки, наприклад, Boost)
Автоматичне управління пам'яттю (GC)	Вбудована (Garbage Collector)	Вбудована (Garbage Collector)	Відсутня (необхідно вручну керувати пам'яттю)

Вибір C# для розробки даної системи обґрунтований його високою інтеграцією з продуктами Microsoft, що забезпечує зручність та ефективність розробки на платформі .NET. Завдяки простому синтаксису та потужним інструментам, таким як Visual Studio, C# дозволяє швидко та ефективно створювати надійні та масштабовані бізнес-додатки [17]. Вбудована підтримка асинхронного програмування значно спрощує розробку багатозадачних систем, що є важливим для управління великим обсягом даних та операцій. Крім того, автоматичне управління пам'яттю в C# допомагає зменшити ймовірність помилок, пов'язаних з керуванням ресурсами, що підвищує надійність системи.

Від обраної СУБД залежить продуктивність, надійність, масштабованість та безпека збереження і обробки даних. При виборі СУБД необхідно враховувати специфіку задач, обсяг даних, що обробляються, вимоги до продуктивності та інтеграції з іншими компонентами системи.

MS SQL Server є потужною реляційною СУБД, розробленою компанією Microsoft, що забезпечує високу продуктивність та надійність. MySQL у свою чергу є популярною відкритою реляційною СУБД, яка забезпечує високу продуктивність та гнучкість. PostgreSQL є потужною відкритою реляційною СУБД, відомою своєю відповідністю стандартам SQL та багатим набором функцій [18-21].

У табл. 2.2 проведено порівняльний аналіз розглянутих СУБД, що дозволить вибрати оптимальну із них.

Таблиця 2.2 – Порівняльний аналіз СУБД

Характеристика	MS SQL Server	MySQL	PostgreSQL
Інтеграція з продуктами Microsoft	Висока	Низька	Низька
Підтримка складних запитів та аналітики	Висока	Середня	Висока
Безпека та управління доступом	Висока (вбудовані функції безпеки)	Середня (менш розвинені функції)	Висока (розширені налаштування)
Підтримка транзакцій	Висока	Висока	Висока
Інструменти для розробників	Вбудовані (SQL Server Management Studio)	Середня (менше інтегрованих інструментів)	Висока (розширені інструменти)
Продуктивність під навантаженням	Висока	Висока	Висока
Підтримка аналітичних функцій	Вбудовані аналітичні служби (SSAS)	Середня (обмежена підтримка)	Висока (підтримка аналітичних запитів)

Вибір MS SQL Server для розробки даної системи обґрунтований його високою інтеграцією з продуктами Microsoft, що забезпечує зручність та

ефективність розробки на платформі .NET. Ця СУБД пропонує розвинену підтримку складних запитів та аналітичних функцій, що дозволяє ефективно обробляти великі обсяги даних і виконувати складні аналітичні завдання. Вбудовані функції безпеки та управління доступом гарантують захист даних, що є критично важливим для корпоративних застосунків. Крім того, потужні інструменти для розробників, такі як SQL Server Management Studio, значно спрощують процес адміністрування та розробки баз даних, підвищуючи загальну продуктивність роботи системи [22].

2.2 Аналіз вимог. Use-case діаграми. Основні прецеденти

Аналіз вимог є важливим етапом у процесі розробки системи автоматизації управління зберіганням та видачею електричних інструментів. На цьому етапі визначаються функціональні та нефункціональні вимоги до системи, які дозволяють сформулювати чітке уявлення про її можливості та обмеження.

Функціональні вимоги відображають конкретні дії, які система повинна виконувати для задоволення потреб користувачів. Вони включають деталі щодо управління інструментами, видачі та обслуговування, пошуку, звітності та інших важливих аспектів роботи системи. У табл. 2.3 представлені функціональні вимоги до застосунку, що дозволяють забезпечити ефективне управління електричними інструментами та оптимізувати відповідні бізнес-процеси.

Таблиця 2.3 – Функціональні вимоги до застосунку

Вимоги	Опис
1	2
REQ-1	Програма повинна забезпечувати функцію видачі та повернення інструментів, включаючи облік всіх операцій.
REQ-2	Система має підтримувати функції технічного обслуговування інструментів, включаючи планування та реєстрацію ремонтів.
REQ-3	Має бути можливість перегляду історії видачі та обслуговування інструментів.

Продовження табл. 2.3

1	2
REQ-4	Програма повинна забезпечувати ведення довідників категорій інструментів, інструментів та працівників.
REQ-6	Програма має надавати можливість створення звітів щодо загальної вартості обслуговування інструментів, використання інструментів за категоріями та інструментів, виданих за певний період.

Нефункціональні вимоги відіграють ключову роль у забезпеченні якості фінального програмного продукту. Ці вимоги торкаються елементів системи, які не є частиною її основних функцій, але впливають на ефективність та комфорт її використання. Нefункціональні вимоги критично важливі, оскільки вони забезпечують стабільність системи та відповідають очікуванням користувачів, що підтверджено в табл. 2.4.

Таблиця 2.4 – Нefункціональні вимоги до застосунку

Вимоги	Опис
NFR-1	Система повинна забезпечувати швидкий відгук на запити користувачів, з часом відгуку не більше 2 секунд для основних операцій.
NFR-2	Програма повинна забезпечувати високу надійність, з мінімальним часом простою, не більше 0,1% на місяць.
NFR-3	Система повинна підтримувати безпечний доступ, включаючи аутентифікацію користувачів та шифрування даних.
NFR-4	Програма повинна мати інтуїтивно зрозумілий інтерфейс користувача, що забезпечує легкість навчання та використання.
NFR-5	Система повинна бути масштабованою, дозволяючи легко додавати нові функціональні можливості та підтримувати збільшення кількості користувачів без значного зниження продуктивності.
NFR-6	Програма повинна забезпечувати сумісність з іншими системами, що використовуються на підприємстві, через стандартизовані API та протоколи інтеграції.

Актори системи визначають користувачів або зовнішні системи, які взаємодіють із програмним забезпеченням для виконання певних функцій. Вони відіграють ключову роль у визначенні функціональних можливостей системи та впливають на її архітектуру і дизайн. У табл. 2.5 представлені основні актори системи та їхні ролі.

Таблиця 2.5 – Ролі та цілі учасників застосунку

Актори	Цілі
Системний адміністратор	Управління обліковими даними, моніторинг подій, а також виконання всіх функцій, доступних користувачам.
Користувач	Виконання основних функцій системи, таких як видача інструментів, їх обслуговування, пошук, ведення довідників та створення звітів.

Варіанти використання (use-case) описують різні сценарії взаємодії користувачів з системою для досягнення конкретних цілей. Вони допомагають зрозуміти вимоги до функціональності системи та визначити, які дії повинні бути підтримані програмним забезпеченням. У табл. 2.6 представлені основні варіанти використання для даної системи.

Таблиця 2.6 – Опис варіантів використання системи

Варіант	Ім'я	Опис
1	2	3
UC1	Реєстрація користувачів	Забезпечує можливість реєстрації нових користувачів у системі
UC2	Вхід у систему	Забезпечує аутентифікацію користувачів і доступ до їхніх облікових записів
UC3	Огляд зареєстрованих користувачів	Надає системним адміністраторам можливість переглядати список усіх зареєстрованих користувачів
UC4	Додавання користувачів	Дозволяє адміністраторам додавати нових користувачів до системи
UC5	Оновлення даних користувача	Дозволяє оновлювати інформацію про користувачів у системі

Продовження табл. 2.6

1	2	3
UC6	Перегляд категорій інструментів	Надає можливість перегляду всіх категорій інструментів у системі
UC7	Реєстрація категорії	Дозволяє вводити нові категорії інструментів до системи
UC8	Оновлення даних категорії	Дозволяє змінювати інформацію про існуючі категорії інструментів
UC9	Перегляд списку інструментів	Дозволяє користувачам переглядати всі доступні інструменти в каталозі
UC10	Додавання інструменту	Дозволяє адміністратору вводити нові інструменти до системи
UC11	Оновлення даних інструменту	Дозволяє редагувати інформацію про інструменти, вибрані зі списку
UC12	Перегляд каталогу працівників	Дає можливість користувачам переглядати всіх працівників
UC13	Додавання працівника	Дозволяє адміністратору додавати нових працівників до системи
UC14	Оновлення даних працівника	Дозволяє редагувати інформацію про працівників, вибраних зі списку
UC15	Обслуговування інструментів	Уможливорює створення записів про технічне обслуговування та ремонти інструментів
UC16	Видача інструментів	Забезпечує процес видачі інструментів працівникам з реєстрацією операції у системі
UC17	Пошук інструментів	Дозволяє користувачам здійснювати пошук інструментів за назвою або інвентарним номером
UC18	Формування звітності	Надає можливість створювати звіти щодо різних аспектів використання та обслуговування інструментів

Функціональні та нефункціональні вимоги до системи визначають основні аспекти її роботи, включаючи управління інструментами, довідниками, пошук, звітність та системні функції. Визначення цих вимог є критичним етапом, що забезпечує чітке розуміння необхідної функціональності та якісних характеристик

системи. Вимоги формують основу для розробки та подальшого тестування системи.

На основі вимог до системи розроблено use-case діаграми прецедентів для двох ролей: системного адміністратора та користувача. Діаграми прецедентів представлені на рис. 2.1 та рис. 2.2 відповідно.

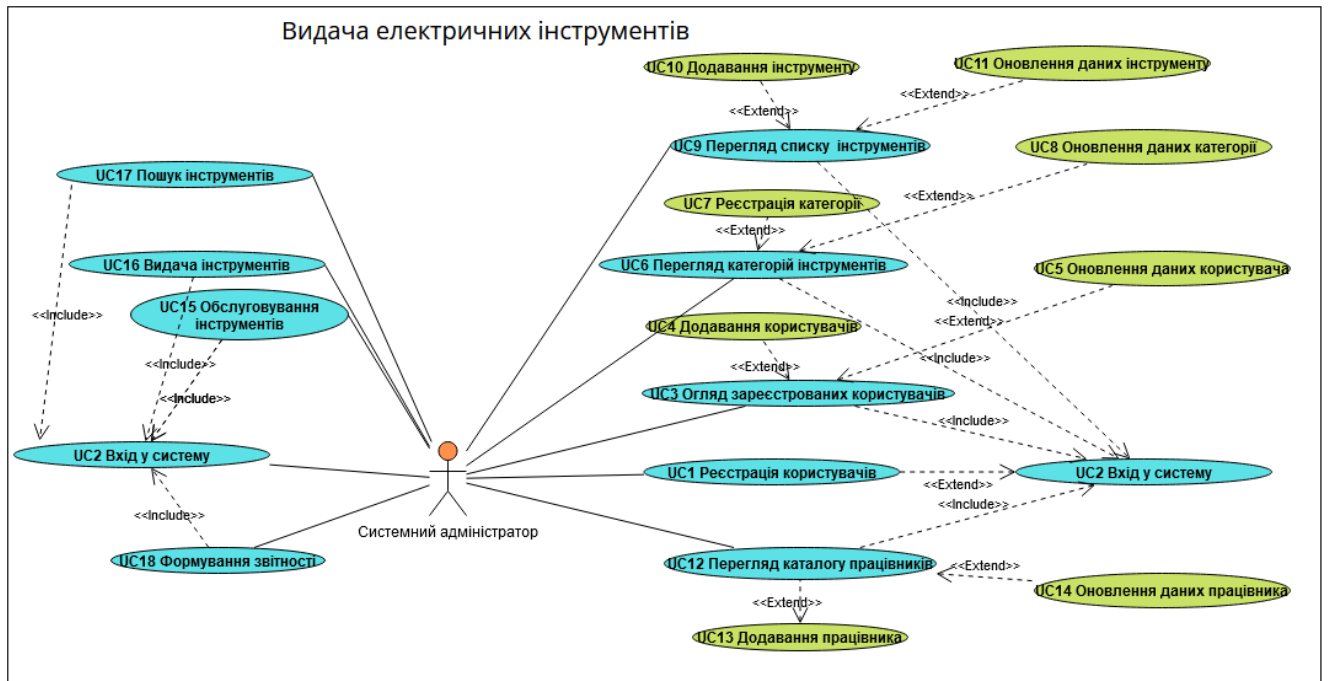


Рисунок 2.1 – Діаграма use-case для ролі системного адміністратора



Рисунок 2.2 – Діаграма use-case для ролі користувача

На представлених діаграмах варіантів використання системи відображено взаємозв'язки між акторами та основними функціями системи. Кожен варіант використання, від UC1 до UC18, ілюструє конкретний шлях взаємодії користувача з системою, включаючи реєстрацію, авторизацію, управління користувачами, інструментами, тощо.

2.3 Проектування бази даних

Проектування бази даних є одним з ключових етапів розробки системи автоматизації управління зберіганням та видачею електричних інструментів. На цьому етапі визначаються структура бази даних, основні таблиці та зв'язки між ними, що забезпечує ефективне зберігання, обробку та доступ до даних. Правильне проектування бази даних гарантує надійність, продуктивність і масштабованість системи, а також спрощує її подальше обслуговування та розширення.

Для системи реалізовано наступні таблиці бази даних.

Таблиця «Users» призначена для зберігання інформації про користувачів системи, включаючи їхні облікові дані та ролі (табл. 2.7). Ця таблиця є ключовою для управління доступом і безпекою в системі, дозволяючи адміністратору контролювати користувачів та їхні права.

Таблиця 2.7 – Атрибути сутності «Users»

№	Поле	Тип даних	Призначення поля
1	UserId	INT	Унікальний ідентифікатор користувача
2	FirstName	NVARCHAR(100)	Ім'я користувача
3	LastName	NVARCHAR(100)	Прізвище користувача
4	UserName	NVARCHAR(100)	Ім'я користувача для входу в систему
5	UsersPassword	NVARCHAR(200)	Пароль користувача для входу в систему
6	RoleId	INT	Ідентифікатор ролі користувача
7	Description	NVARCHAR(MAX)	Додаткова інформація про користувача
8	Email	NVARCHAR(200)	Електронна пошта користувача

Таблиця «ToolTransaction» призначена для зберігання записів про всі транзакції з інструментами, включаючи їх видачу та повернення (табл. 2.8). Ця таблиця є ключовою для відстеження історії використання інструментів і взаємодії з працівниками.

Таблиця 2.8 – Атрибути сутності «ToolTransaction»

№	Поле	Тип даних	Призначення поля
1	ToolTransactionId	INT	Унікальний ідентифікатор транзакції
2	ToolId	INT	Ідентифікатор інструменту
3	EmployeeId	INT	Ідентифікатор працівника, що бере інструмент
4	TransactionDate	DATETIME	Дата видачі інструменту
5	ReturnDate	DATETIME	Дата повернення інструменту
6	Description	NVARCHAR(MAX)	Опис транзакції
7	Status	BIT	Статус транзакції (виконано/не виконано)

Таблиця «Tool» призначена для зберігання інформації про всі інструменти, що знаходяться в системі (табл. 2.9). Вона містить деталі про кожен інструмент, такі як назва, опис, інвентарний номер, ціна, категорія та статус.

Таблиця 2.9 – Атрибути сутності «Tool»

№	Поле	Тип даних	Призначення поля
1	ToolId	INT	Унікальний ідентифікатор інструменту
2	ToolName	NVARCHAR(100)	Назва інструменту
3	Description	NVARCHAR(MAX)	Опис інструменту
4	ToolNumber	NVARCHAR(20)	Інвентарний номер інструменту
5	ToolPrice	FLOAT	Ціна інструменту
6	CategoryId	INT	Ідентифікатор категорії інструменту
7	ToolStatusId	INT	Ідентифікатор статусу інструменту

Таблиця «Logs» призначена для зберігання записів про події та дії, виконані користувачами в системі (табл. 2.10). Вона є важливою для моніторингу активності, аудиту безпеки та аналізу поведінки користувачів.

Таблиця 2.10 – Атрибути сутності «Logs»

№	Поле	Тип даних	Призначення поля
1	LogsId	INT	Унікальний ідентифікатор журналу подій
2	UsersId	INT	Ідентифікатор користувача, який виконав подію
3	EventNameShow	NVARCHAR(MAX)	Назва події
4	EventDate	DATETIME	Дата та час події
5	UsersName	NVARCHAR(250)	Ім'я користувача, який виконав подію

Таблиця «Category» призначена для зберігання інформації про різні категорії інструментів (табл. 2.11). Ця таблиця є важливою для класифікації та організації інструментів у системі, дозволяючи впорядковувати їх за певними групами.

Таблиця 2.11 – Атрибути сутності «Category»

№	Поле	Тип даних	Призначення поля
1	CategoryId	INT	Унікальний ідентифікатор категорії
2	CategoryName	NVARCHAR(100)	Назва категорії
3	Description	NVARCHAR(MAX)	Опис категорії

Таблиця «Employee» призначена для зберігання записів про працівників, які взаємодіють із системою (табл. 2.12). Ця таблиця є ключовою для обліку та управління персоналом у системі, включаючи їхні контактні дані та посади.

Таблиця 2.12 – Атрибути сутності «Employee»

№	Поле	Тип даних	Призначення поля
1	EmployeeId	INT	Унікальний ідентифікатор працівника
2	FirstName	NVARCHAR(50)	Ім'я працівника
3	LastName	NVARCHAR(50)	Прізвище працівника
4	Position	NVARCHAR(100)	Посада працівника
5	Email	NVARCHAR(150)	Електронна пошта працівника
6	Phone	NVARCHAR(20)	Контактний номер телефону працівника

Таблиця «ToolMaintenance» призначена для зберігання записів про технічне обслуговування та ремонти інструментів (табл. 2.13). Вона є ключовою для відстеження стану інструментів, їх обслуговування та пов'язаних витрат.

Таблиця 2.13 – Атрибути сутності «ToolMaintenance»

№	Поле	Тип даних	Призначення поля
1	MaintenanceId	INT	Унікальний ідентифікатор запису обслуговування
2	ToolId	INT	Ідентифікатор інструменту, що обслуговується
3	MaintenanceDate	DATETIME	Дата проведення обслуговування або ремонту
4	Description	NVARCHAR(MAX)	Опис проведених робіт
5	Cost	FLOAT (53)	Вартість обслуговування або ремонту
6	Status	BIT	Статус обслуговування (виконано/не виконано)

Після створення всіх таблиць, було побудовано діаграму бази даних, яка представлена на рис. 2.3.

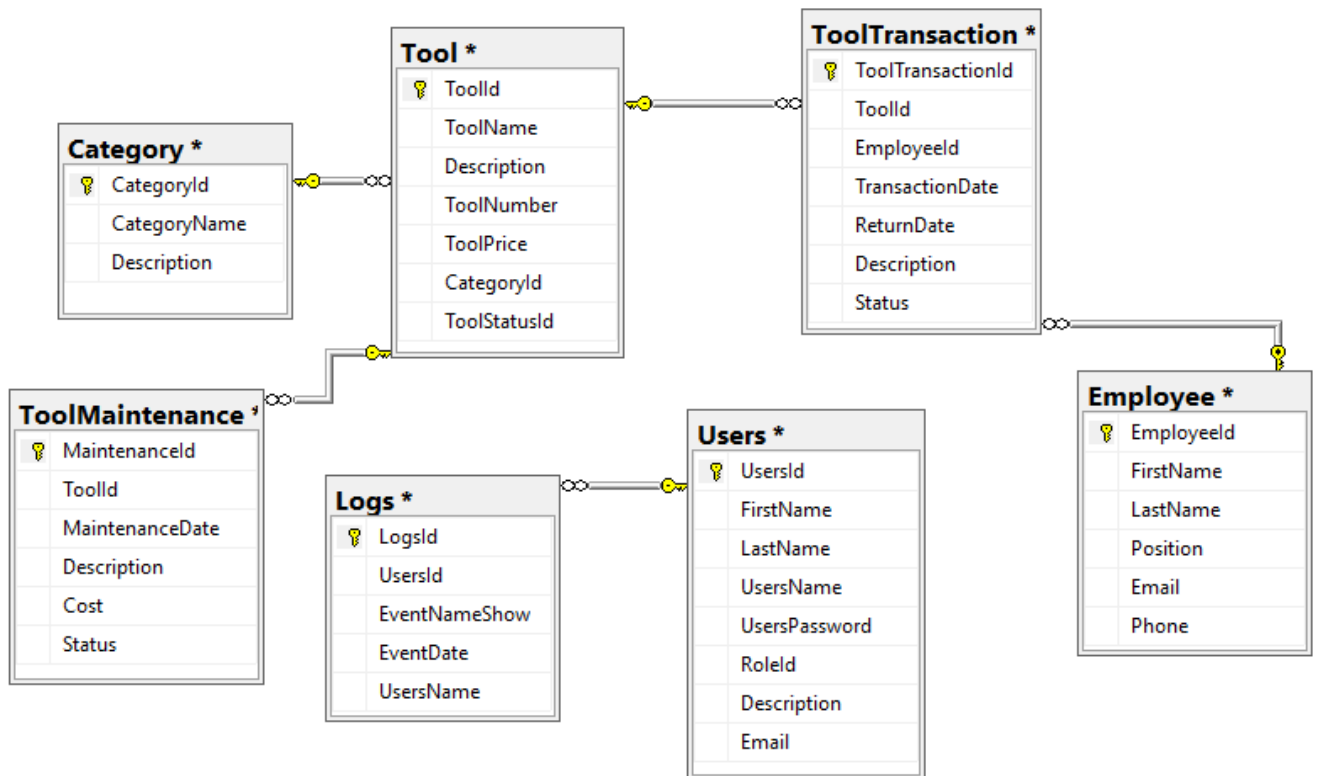


Рисунок 2.3 – Фізична модель бази даних

Метою розробки структури бази даних було не лише забезпечення її цілісності, а й оптимізація процесів зберігання та управління даними всередині системи, що сприяє ефективній обробці інформації та забезпечує швидкий доступ до необхідних даних для користувачів.

2.4 Архітектура проекту

Архітектура проекту є фундаментальною частиною процесу розробки, яка визначає структурні аспекти системи та забезпечує ефективну взаємодію між її компонентами. Вибір правильної архітектури впливає на продуктивність, масштабованість, надійність та підтримуваність програмного забезпечення. Для розробки системи автоматизації управління зберіганням та видачею електричних інструментів було обрано трьохрівневу архітектуру.

Трьохрівнева архітектура складається з рівня презентації, рівня бізнес-логіки та рівня даних, що забезпечує чітке розмежування функцій та відповідальностей між різними частинами системи. Рівень презентації відповідає за взаємодію з користувачем, надаючи зручний та інтуїтивно зрозумілий інтерфейс. Рівень бізнес-логіки реалізує основні правила та процеси, що керують функціональністю системи, забезпечуючи централізоване управління логікою додатка. Рівень даних відповідає за зберігання та управління інформацією, забезпечуючи надійний та ефективний доступ до бази даних.

Вибір трьохрівневої архітектури обґрунтований її численними перевагами. Вона забезпечує модульність та масштабованість системи, дозволяючи легко додавати нові функції або змінювати існуючі без значного впливу на інші компоненти. Розділення бізнес-логіки від інтерфейсу користувача сприяє більшій гнучкості у розробці та підтримці, а також покращує безпеку системи, зменшуючи ризик несанкціонованого доступу до даних. Крім того, така архітектура полегшує тестування та відлагодження, оскільки кожен рівень може бути перевірений окремо, забезпечуючи високу якість кінцевого продукту.

На початку розробки було створено шар доступу до даних, який забезпечує надійне та ефективне зберігання та обробку інформації. Цей шар відповідає за взаємодію з базою даних, включаючи виконання запитів та управління транзакціями. Діаграма класів цього рівня представлена на рис. 2.4.

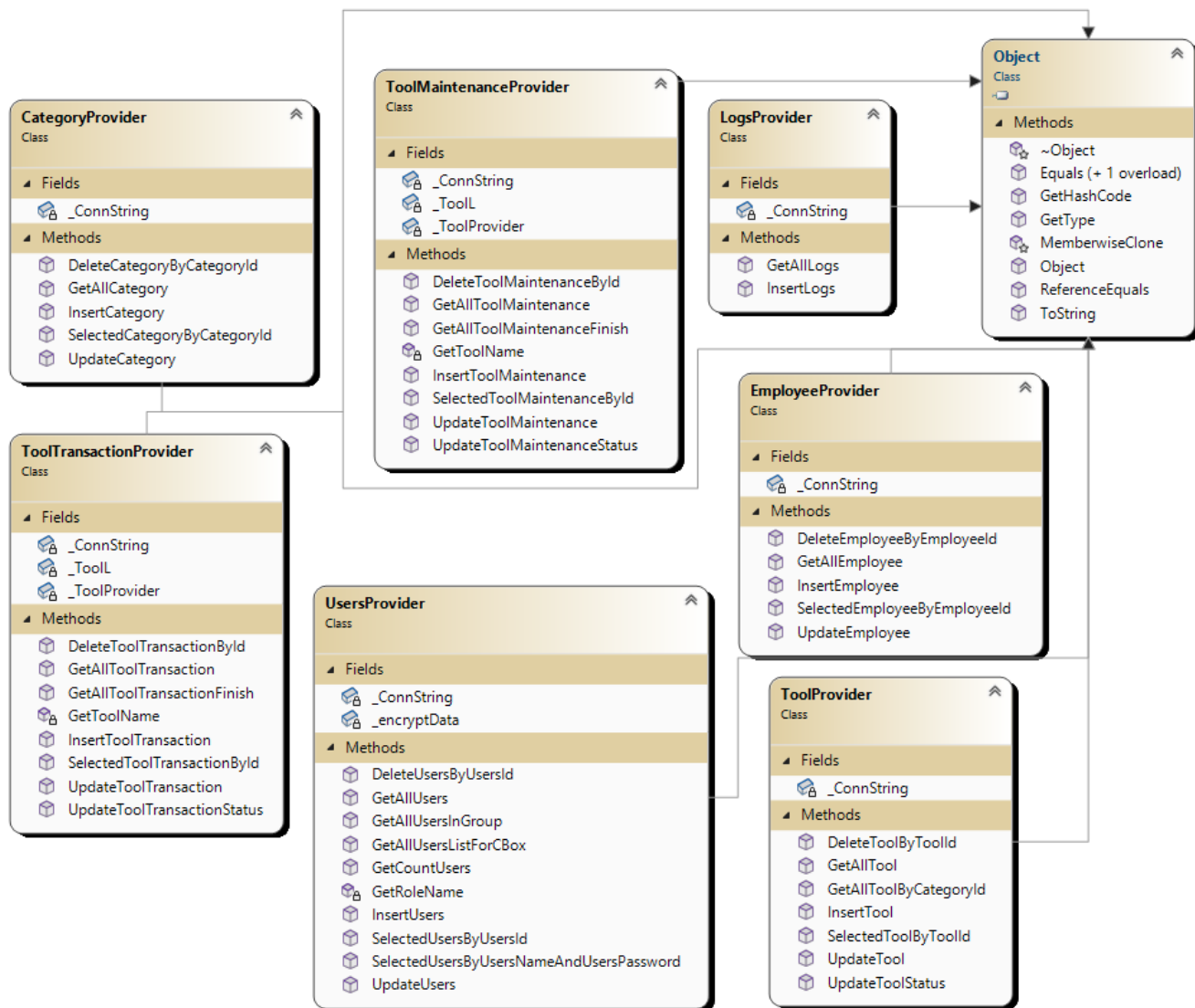


Рисунок 2.4 – Діаграма класів рівня даних

Діаграма даного рівня складається із наступних класів:

– клас `CategoryProvider` відповідає за управління категоріями інструментів. Він включає методи для створення нових категорій, оновлення даних існуючих категорій, видалення категорій з системи, а також методи для отримання детальної інформації про категорії, включно з фільтрацією за різними критеріями;

– клас `EmployeeProvider` відповідає за управління даними про працівників. Він містить методи для додавання нових працівників, оновлення інформації про існуючих працівників, видалення записів працівників, а також методи для отримання інформації про працівників, включаючи фільтрацію за різними параметрами;

– клас `LogsProvider` відповідає за управління записами подій у системі. Включає методи для додавання нових подій, отримання записів про події з фільтрацією за різними критеріями, а також видалення старих або непотрібних записів подій;

– клас `ToolMaintenanceProvider` відповідає за управління даними про обслуговування інструментів. Включає методи для реєстрації нових випадків обслуговування, оновлення даних про обслуговування, видалення записів обслуговування, а також методи для отримання інформації про проведені обслуговування з можливістю фільтрації за різними параметрами;

– клас `ToolProvider` відповідає за управління даними про інструменти. Він містить методи для додавання нових інструментів, оновлення інформації про існуючі інструменти, видалення інструментів з системи, а також методи для отримання детальної інформації про інструменти, включаючи фільтрацію за різними критеріями;

– клас `ToolTransactionProvider` відповідає за управління процесом видачі інструментів. Включає методи для реєстрації нових транзакцій видачі, оновлення даних про існуючі транзакції, видалення записів транзакцій, а також методи для отримання детальної інформації про транзакції видачі з можливістю фільтрації за різними параметрами;

– клас `UsersProvider` відповідає за управління даними про користувачів системи. Він містить методи для створення нових облікових записів користувачів, оновлення інформації про користувачів, видалення облікових записів, а також методи для отримання детальної інформації про користувачів з можливістю фільтрації за різними критеріями.

До рівня користувацького інтерфейсу належать наступні класи:

- клас `IssuanceMDI` є основним інтерфейсом користувача, який забезпечує доступ до всіх функцій системи. Він дозволяє користувачам навігувати між різними модулями, такими як управління інструментами, видача, обслуговування та звітність. Головне вікно також відображає інформацію про поточний статус системи та надає можливості для швидкого доступу до основних операцій;

- клас `LoginForm` відповідає за процес входу користувачів до системи. Він включає поля для введення імені користувача та пароля, а також методи для перевірки облікових даних та надання доступу до системи. У разі успішної авторизації користувачі перенаправляються до головного вікна системи;

- клас `PersonalizationForm` дозволяє користувачам налаштовувати інтерфейс системи відповідно до своїх потреб та уподобань. Він включає методи для зміни налаштувань профілю, таких як тема інтерфейсу, мова та інші персональні параметри. Цей клас забезпечує зручність та інтуїтивність користування системою;

- клас `SystemLogForm` забезпечує відображення та аналіз записів подій у системі. Він дозволяє адміністраторам переглядати лог подій, фільтрувати записи за різними критеріями та виконувати аналіз для виявлення проблем або підозрілих дій. Цей клас є важливим інструментом для забезпечення безпеки та надійності системи;

- клас `UpdateUsersForm` надає адміністраторам можливість редагувати інформацію про користувачів та видаляти облікові записи. Він включає поля для зміни даних користувача, таких як ім'я, електронна пошта, роль та інші атрибути. Цей клас забезпечує ефективне управління користувачами в системі;

- клас `UsersForm` дозволяє адміністраторам додавати нових користувачів до системи та переглядати інформацію про існуючих користувачів. Він включає поля для введення даних нового користувача та методи для збереження цієї інформації в базі даних. Клас також забезпечує функції перегляду та пошуку користувачів за різними критеріями;

– клас `ToolMaintenanceForm` дозволяє користувачам реєструвати випадки технічного обслуговування інструментів. Він включає поля для введення інформації про обслуговування, такі як дата, опис робіт та вартість. Цей клас забезпечує ефективний облік та управління обслуговуванням інструментів;

– клас `ToolMaintenanceHistForm` відображає історію обслуговування інструментів. Він дозволяє користувачам переглядати записи про проведені ремонти та обслуговування, включаючи дату, опис робіт та витрати. Цей клас є важливим для моніторингу стану інструментів та планування подальших робіт;

– клас `ToolTransactionForm` забезпечує процес видачі інструментів працівникам. Він включає поля для введення даних про інструмент, працівника та дату видачі, а також методи для збереження цієї інформації в базі даних. Цей клас дозволяє ефективно керувати процесом видачі та повернення інструментів;

– клас `ToolTransactionHistForm` призначений для відображення історії видачі інструментів. Він дозволяє користувачам переглядати записи про всі транзакції видачі, включаючи інформацію про інструменти, працівників, дати видачі та повернення. Цей клас є важливим для моніторингу використання інструментів та аналізу їх обігу;

– клас `CategoryForm` дозволяє користувачам додавати нові категорії інструментів та переглядати існуючі категорії. Він включає поля для введення назви категорії та опису, а також методи для збереження цієї інформації в базі даних. Клас також забезпечує функції перегляду та пошуку категорій за різними критеріями;

– клас `EmployeeForm` дозволяє адміністраторам додавати нових працівників до системи та переглядати інформацію про існуючих працівників. Він включає поля для введення даних про працівника, таких як ім'я, прізвище, посада, електронна пошта та контактний номер. Клас також забезпечує функції перегляду та пошуку працівників за різними критеріями;

– клас `ToolForm` призначений для додавання нових інструментів до системи та перегляду інформації про існуючі інструменти. Він включає поля для введення даних про інструмент, таких як назва, опис, інвентарний номер, ціна та категорія.

Клас також забезпечує функції перегляду та пошуку інструментів за різними критеріями;

- клас `UpdateCategoryForm` дозволяє користувачам редагувати інформацію про існуючі категорії інструментів та видаляти їх із системи. Він включає поля для зміни назви категорії та опису, а також методи для збереження змін у базі даних. Цей клас забезпечує ефективне управління категоріями інструментів;

- клас `UpdateEmployeeForm` дозволяє адміністраторам редагувати інформацію про існуючих працівників та видаляти їх із системи. Він включає поля для зміни даних про працівника, таких як ім'я, прізвище, посада, електронна пошта та контактний номер. Цей клас забезпечує ефективне управління працівниками у системі;

- клас `UpdateToolForm` дозволяє користувачам редагувати інформацію про існуючі інструменти та видаляти їх із системи. Він включає поля для зміни даних про інструмент, таких як назва, опис, інвентарний номер, ціна та категорія. Цей клас забезпечує ефективне управління інструментами у системі.

Ці класи інтерфейсу забезпечують важливі функціональні можливості для управління продукцією, категоріями, а також для ведення звітності про виготовлені та продані товари, дозволяючи користувачам ефективно взаємодіяти з даними в системі.

У структурі системи бізнес-логіка відіграє вирішальну роль, будучи основою для виконання всіх бізнес-операцій. Вона встановлює правила для обробки даних, їх аналізу та формування ключових висновків, що є критичними для ефективності системи. Бізнес-логіка реалізована як набір модулів, що дозволяє легко адаптувати систему до змінних вимог, спрощує масштабування і оновлення компонентів. Такий підхід також сприяє відділенню бізнес-правил від інтерфейсу користувача та взаємодії з базою даних, що полегшує розробку за рахунок концентрації на ключових аналітичних та обробних функціях.

На архітектурній схемі (рис. 2.6) видно, як бізнес-логіка координує взаємодію з іншими частинами системи та визначає області відповідальності кожного компонента. Ця діаграма дозволяє глибше зрозуміти структуру програми

та особливості взаємодії її елементів, вказуючи на значний вплив бізнес-логіки на загальну продуктивність і функціональність системи.

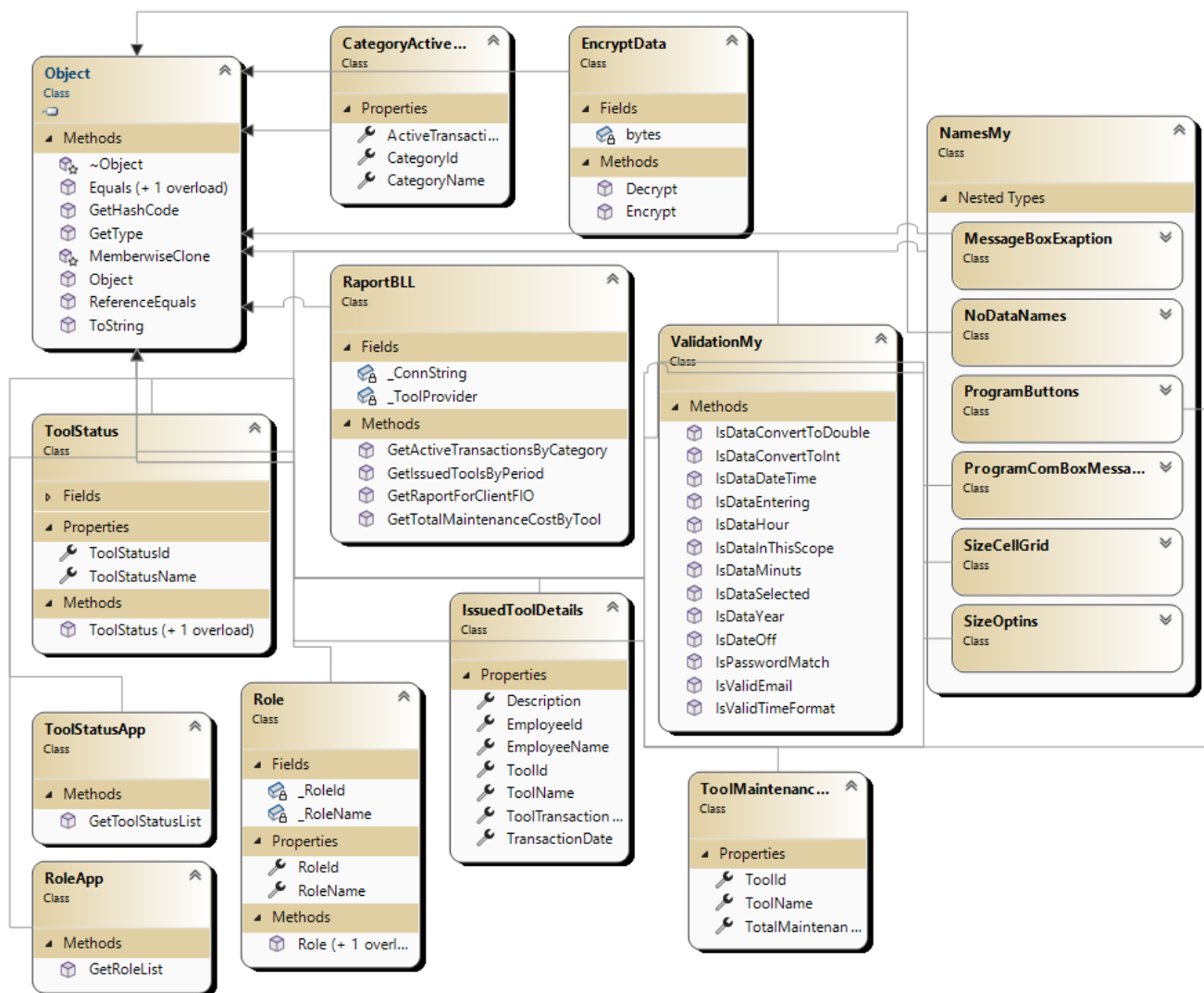


Рисунок 2.6 – Діаграма класів бізнес логіки

Діаграма класів бізнес логіки складається із наступних класів:

- **ValidationMy** відповідає за перевірку введених користувачами даних на відповідність встановленим правилам та обмеженням. Він включає методи для валідації полів форм, таких як електронна пошта, номери телефонів, ідентифікатори та інші дані, забезпечуючи правильність та цілісність інформації перед її збереженням у базі даних;
- **NamesMy** забезпечує роботу з іменами користувачів та інших об'єктів системи. Він включає методи для нормалізації та форматування імен, а також для

перевірки унікальності імен у системі, що допомагає уникнути дублювання та забезпечує зручність використання;

- EncryptData відповідає за шифрування та дешифрування конфіденційної інформації у системі. Він включає методи для безпечного зберігання та передачі даних, таких як паролі, особисті дані користувачів та інші чутливі інформаційні об'єкти, забезпечуючи високий рівень безпеки;

- RoleApp керує ролями та правами доступу користувачів у системі. Він включає методи для створення, редагування та видалення ролей, а також для призначення прав доступу до різних функцій та ресурсів системи, забезпечуючи контроль доступу відповідно до встановлених політик безпеки;

- ToolMaintenanceCost відповідає за обчислення та управління витратами на технічне обслуговування інструментів. Він включає методи для розрахунку вартості обслуговування, створення звітів про витрати та аналізу ефективності витрат, що допомагає контролювати бюджети та планувати обслуговування;

- CategoryActiveTransactions забезпечує відстеження активних транзакцій за категоріями інструментів. Він включає методи для обліку та моніторингу активних транзакцій, таких як видача та повернення інструментів, що дозволяє отримувати актуальну інформацію про використання інструментів у різних категоріях;

- IssuedToolDetails керує деталями виданих інструментів. Він включає методи для обліку інформації про видані інструменти, дати видачі та повернення, відповідальних працівників та статус інструментів, що дозволяє ефективно відстежувати та контролювати процес видачі;

- ToolStatusApp відповідає за управління статусами інструментів у системі. Він включає методи для зміни та оновлення статусів інструментів, таких як доступний, виданий, у ремонті, та інші, що забезпечує точну інформацію про поточний стан кожного інструменту у системі.

Перелічені класи спільно забезпечують важливий функціонал бізнес-логіки в системі, сприяючи її надійності, безпеці та корисності для кінцевих користувачів.

2.5 Висновок

У рамках даного розділу було здійснено всебічний аналіз та вибір технологій, необхідних для розробки системи автоматизації управління зберіганням та видачею електричних інструментів. Проаналізовано різні мови програмування, серед яких C#, Java та C++, та обрано C# як найбільш підходящу для реалізації проекту. Також було розглянуто декілька систем управління базами даних, зокрема MS SQL Server, MySQL та PostgreSQL, і зроблено вибір на користь MS SQL Server завдяки її інтеграції з платформою .NET та високої продуктивності.

Проведено детальний аналіз вимог до системи, що включає функціональні та нефункціональні вимоги, а також визначено основних акторів та описано варіанти використання системи. Розроблено use-case діаграми для двох ключових ролей: системного адміністратора та користувача, що дозволяє чітко окреслити взаємодію користувачів із системою та їхні потреби.

Проектування бази даних включало опис таблиць та побудову фізичної моделі бази даних, що забезпечує надійне зберігання та обробку інформації. Описано структуру таблиць, необхідних для зберігання даних про інструменти, категорії, працівників, транзакції та обслуговування.

Обрано трьохрівневу архітектурну модель, що включає рівні презентації, бізнес-логіки та даних. Детально описано діаграми класів для кожного рівня, що забезпечує чітке розмежування функцій та дозволяє легко масштабувати та підтримувати систему.

Результати проведеного аналізу та проектування дозволяють перейти до етапу розробки системи, використовуючи вибрані технології та архітектурні рішення.

3 РОЗРОБКА ТА ВЕРИФІКАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Розробка основних алгоритмів

Процес розробки системи автоматизації процесу зберігання та видачі електричних інструментів включає кілька ключових етапів. Один з найважливіших аспектів – створення алгоритмів, що забезпечують надійну та ефективну роботу системи. Основним завданням є розробка алгоритмів, які включають процедури авторизації, обробки запитів на видачу та повернення інструментів, а також управління складськими операціями. Першим кроком в даному процесі є алгоритм авторизації користувача, що дозволяє забезпечити захищений доступ до системи та запобігти несанкціонованим діям. На рис. 3.1 представлено блок-схему алгоритму авторизації користувача.



Рисунок 3.1 – Блок-схема алгоритму авторизації користувача

Авторизація користувача починається з введення логіну та паролю користувачем, після чого відбувається натискання кнопки «Увійти». Дані передаються в базу даних для перевірки, і отримана відповідь визначає подальші дії. Якщо введені дані коректні, користувач отримує доступ до головного вікна програми, у випадку помилки виводиться повідомлення про помилку, і процес авторизації повторюється.

На рис. 3.2 представлено блок-схему алгоритму додавання інформації про електричний інструмент у базу даних системи.

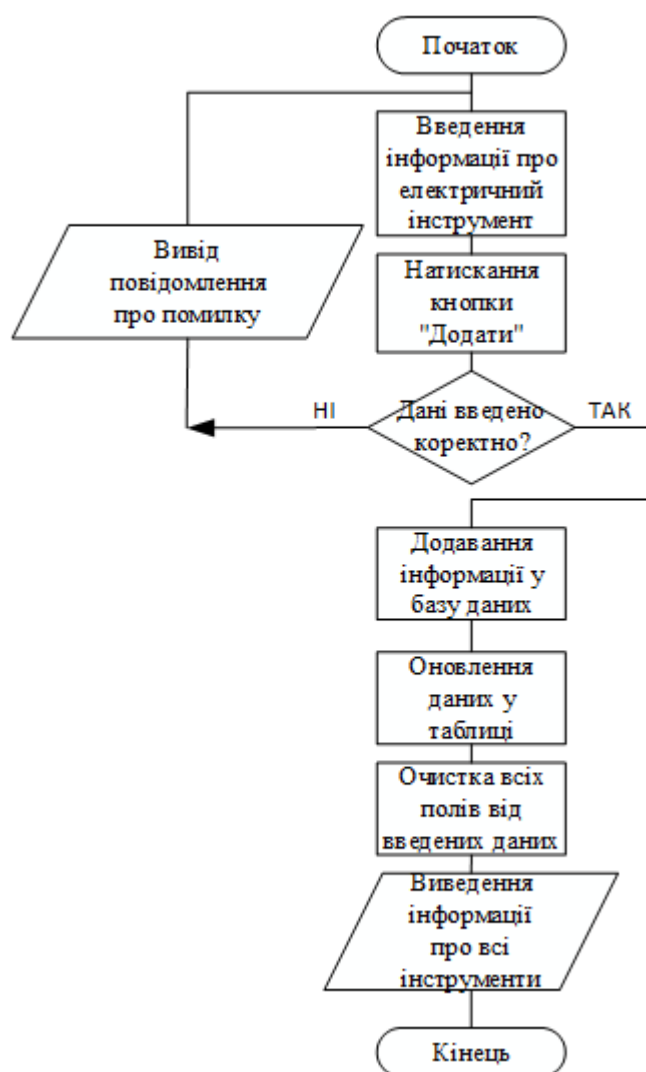


Рисунок 3.2 – Блок-схема алгоритму додавання електричного інструменту

Алгоритму додавання електричного інструменту у базу даних починається з введення користувачем необхідної інформації про інструмент та натискання

кнопки «Додати». Після цього відбувається перевірка введених даних, і в разі їх коректності інформація додається до бази даних, оновлюються дані у відповідних таблицях, очищуються поля введення, і виводиться інформація про всі інструменти. Якщо введені дані некоректні, виводиться повідомлення про помилку, і процес повторюється.

Рис. 3.3 відображає блок-схему алгоритму редагування обраного із списку електричного інструменту та збереження зміненої інформації у базі даних системи.

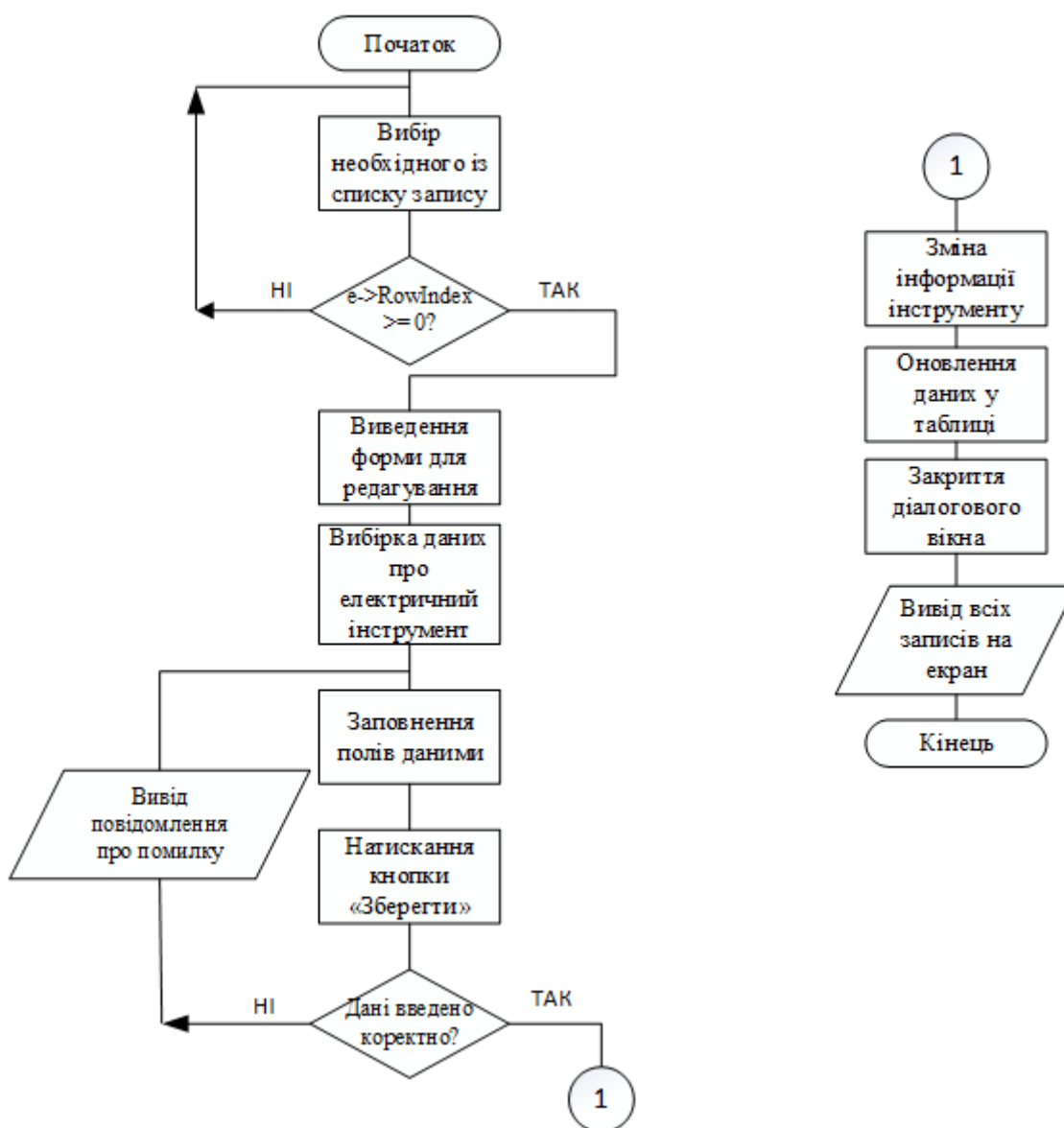


Рисунок 3.3 – Блок-схема алгоритму редагування електричного інструменту

Блок-схема алгоритму редагування електричного інструменту починається з вибору необхідного запису зі списку. Якщо вибраний запис є валідним, відкривається форма для редагування, де користувач може змінити дані про інструмент. Після заповнення полів даними та натискання кнопки «Зберегти» відбувається перевірка введених даних; у разі успішної перевірки інформація змінюється у базі даних та оновлюються відповідні таблиці, і всі записи виводяться на екран. Якщо дані введено некоректно, виводиться повідомлення про помилку, і процес повторюється.

На рис. 3.4 представлено блок-схему алгоритму видалення інформації електричного інструменту із бази даних.

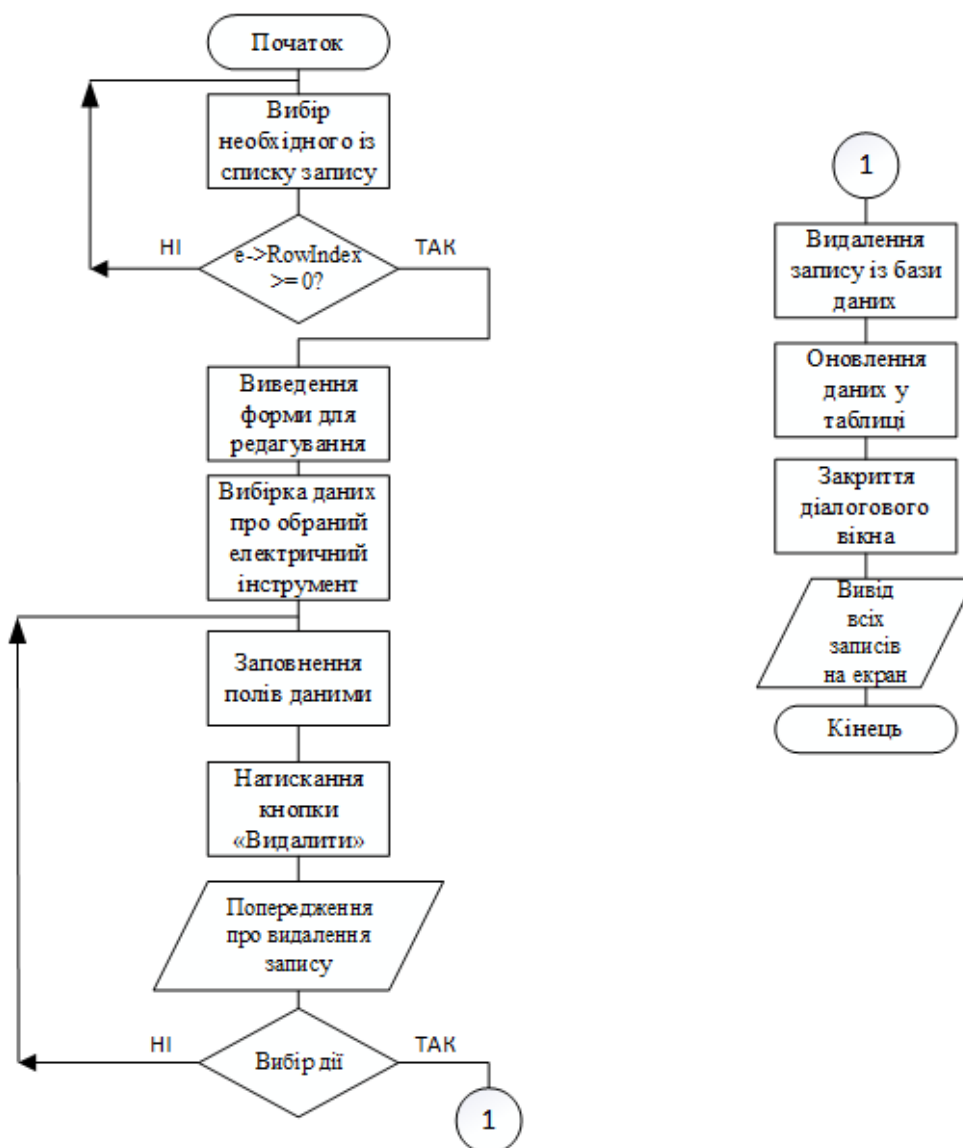


Рисунок 3.4 – Блок-схема алгоритму видалення електричного інструменту

Блок-схема алгоритму видалення електричного інструменту починається з вибору необхідного запису зі списку. Якщо вибраний запис є валідним, відкривається форма для редагування, де натискає кнопку «Видалити». Після чого він отримує попередження про видалення запису. Якщо підтверджується дія, запис видаляється з бази даних, оновлюються таблиці, закривається діалогове вікно, і всі записи виводяться на екран.

3.2 Розробка програмних модулів системи

Розробка програмних модулів для системи автоматизації процесу зберігання та видачі електричних інструментів включає створення різноманітних компонентів, що забезпечують функціональність та інтеграцію всіх частин системи. Основними завданнями є розробка модулів для обробки даних, інтерфейсів користувача, а також механізмів взаємодії з базою даних. Одним з ключових аспектів є налаштування параметрів підключення до бази даних, що гарантує надійний та безпечний обмін інформацією між компонентами системи. На рис. 3.5 наведено приклад коду параметрів підключення до бази даних.

```

<!-- Підключення до бази даних -->
<appSettings>
  <add key="CONNECT"
      value="Data Source=(LocalDB)\MSSQLLocalDB;
            AttachDbFilename=|DataDirectory|\DB.mdf;
            Integrated Security=True" />

```

Рисунок 3.5 – Параметри підключення до бази даних

Цей код налаштовує підключення до бази даних у файлі конфігурації програми. В секції `<appSettings>` створюється ключ «CONNECT», якому присвоюється рядок підключення до локальної бази даних SQL Server. Рядок підключення вказує, що використовується локальна база даних (LocalDB)\MSSQLLocalDB, база даних розміщується у файлі DB.mdf, що

знаходиться в каталозі даних програми, та використовується інтегрована аутентифікація.

На початку розробки системи автоматизації процесу зберігання та видачі електричних інструментів було створено головне вікно програми, яке забезпечує доступ до основних функцій системи. Інтерфейс головного вікна включає меню з розділами для управління, довідників, пошуку, звітності та адміністрування, що дозволяє користувачам ефективно взаємодіяти з системою. На рис. 3.6 зображено головне вікно програми, яке демонструє структуру інтерфейсу користувача.

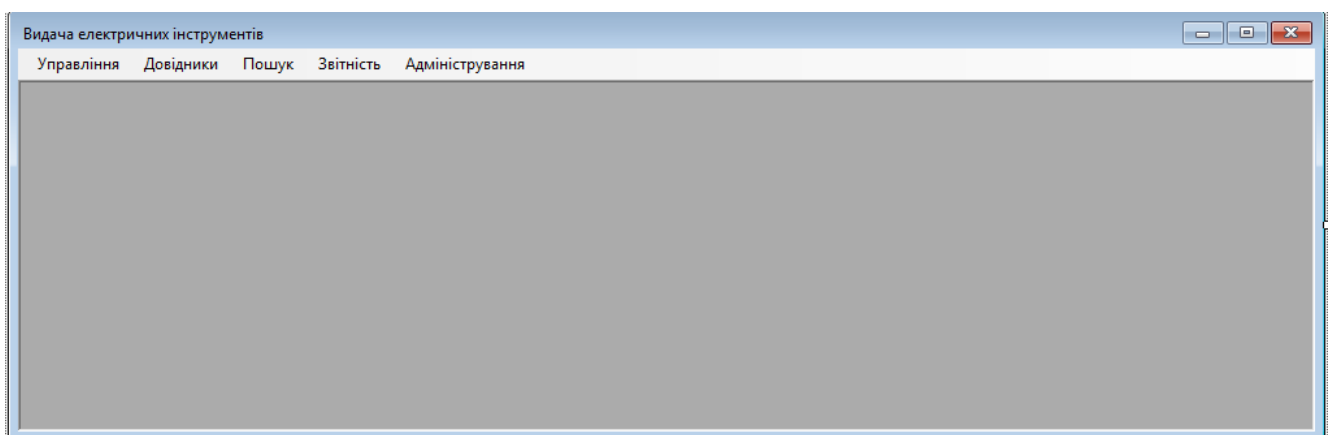


Рисунок 3.6 – Головне вікно системи

Для реалізації функціоналу видачі інструментів в програмі, обробник події натискання на відповідний пункт меню створює та відображає форму «ToolTransactionForm», яка відкривається в режимі MDI та максимізується для зручності користувача. Такий підхід забезпечує ефективне управління вікнами та спрощує користувачам взаємодію з системою. На рис. 3.7 представлено код обробника події для меню видачі інструментів.

```
private void видачаІнструментівToolStripMenuItem_Click(object sender, EventArgs e) {
    CloseAllWindows();
    ToolTransactionForm toolTransactionForm = new ToolTransactionForm();
    toolTransactionForm.MdiParent = this;
    toolTransactionForm.WindowState = FormWindowState.Maximized;
    toolTransactionForm.Show();
}
```

Рисунок 3.7 – Реалізація коду меню для відкриття форми видачі інструментів

Код спрацьовує, як тільки користувач натискає на цей елемент, спершу закриваються всі відкриті вікна за допомогою методу «CloseAllWindows». Після цього створюється нова форма «ToolTransactionForm», яка встановлюється як дочірня до поточного головного вікна. Далі, нова форма максимізується) та відображається на екрані.

Форма для видачі інструментів є важливим компонентом системи, що дозволяє користувачам зручно вводити та обробляти дані про інструменти, які видаються. Інтерфейс форми включає поля для вибору категорії, інструменту, працівника, дати видачі та опису проведених робіт, що забезпечує повноту та точність зібраної інформації. На рис. 3.8 зображено розроблену форму для видачі інструментів, яка демонструє основні елементи цього інтерфейсу.

Рисунок 3.8 – Реалізація форми видачі інструментів

У конструкторі форми викликається метод «LoadAllDate», який призначений для даних про працівників та категорії (рис. 3.9).

```

private void LoadAllDate() {
    _EmployeeL = _EmployeeProvider.GetAllEmployee();
    EmployeeCBox.DataSource = _EmployeeL;
    EmployeeCBox.ValueMember = "EmployeeId";
    EmployeeCBox.DisplayMember = "FIO";
    _CategoryList = _CategoryProvider.GetAllCategory();
    CategoryCBox.DataSource = _CategoryList;
    CategoryCBox.ValueMember = "CategoryId";
    CategoryCBox.DisplayMember = "CategoryName";
    _IsCategoryLoad = true;
    CategoryCBox_SelectedValueChanged(CategoryCBox, EventArgs.Empty);
}

```

Рисунок 3.9 – Завантаження даних про працівників та категорії

Цей код відповідає за завантаження даних для випадючих списків у формі. Спочатку отримується список всіх працівників за допомогою методу «GetAllEmployee» провайдера `_EmployeeProvider`, після чого цей список призначається джерелом даних для випадючого списку `EmployeeCBox`. Встановлюються властивості для відображення (`DisplayMember`) і значення (`ValueMember`) полів. Аналогічно, отримується список всіх категорій за допомогою методу «GetAllCategory» провайдера `_CategoryProvider`, який призначається джерелом даних для випадючого списку `CategoryCBox`. Встановлюються властивості для відображення та значення полів для цього списку, а також викликається метод «`CategoryCBox_SelectedValueChanged`», щоб обробити зміни значення категорії.

На рис. 3.10 наведено код методу для додавання запису про видачу інструменту працівнику у базу даних.

```

private void AddBtn_Click(object sender, EventArgs e) {
    if (IsDataCorrect()) {
        _ToolProvider.UpdateToolStatus(2, Convert.ToInt32(ToolCBox.SelectedValue));
        _ToolTransactionProvider.InsertToolTransaction(Convert.ToInt32(ToolCBox.SelectedValue),
            Convert.ToInt32(EmployeeCBox.SelectedValue),
            TransactionDate.Value, TransactionDate.Value.AddDays(2), DescriptionTBox.Text);
        DataLoad();
        LoadAllDate();
        ClearAllControls();
    }
}

```

Рисунок 3.10 – Код методу додавання видачі інструменту працівнику

Цей код визначає обробник події для натискання на кнопку «Додати». Після перевірки коректності даних за допомогою методу «IsDataCorrect», виконується оновлення статусу інструменту в базі даних за допомогою методу «UpdateToolStatus», який встановлює статус 2 для обраного інструменту. Далі створюється новий запис транзакції інструменту за допомогою методу «InsertToolTransaction», який записує інформацію про вибраний інструмент, працівника, дати транзакції та опис. Після цього дані перезавантажуються за допомогою методів «DataLoad» і «LoadAllDate», а всі елементи управління очищуються для підготовки до введення нових даних.

Після вибору інструменту із випадального списку, викликається подія «ToolCBox_SelectedValueChanged», код якого представлено на рис. 3.11.

```
private void ToolCBox_SelectedValueChanged(object sender, EventArgs e) {
    if (_IsToolLoad) {
        _selTool = _ToolProvider.SelectedToolByToolId(Convert.ToInt32(ToolCBox.SelectedValue));
        ToolNumberLbl.Text = "(Інв. номер: " + _selTool.ToolNumber + " )";
    }
}
```

Рисунок 3.11 – Код події «ToolCBox_SelectedValueChanged»

Якщо змінна `_IsToolLoad` встановлена в `true`, викликається метод «SelectedToolByToolId» провайдера `_ToolProvider`, який отримує дані про вибраний інструмент за його ідентифікатором. Отриманий інструмент зберігається у змінну `_selTool`, а мітка `ToolNumberLbl` оновлюється, щоб відобразити інвентарний номер вибраного інструменту.

Також розроблено метод, який обробляє події кліку на комірки у таблиці транзакцій інструментів, дозволяючи користувачам виконувати різні операції з транзакціями, такі як оновлення статусу, видалення або редагування даних. Це забезпечує гнучкість і зручність управління транзакціями інструментів у системі. На рис. 3.12 представлено код цього методу.

```

private void ToolTransactionGridView_CellClick(object sender, DataGridViewCellEventArgs e) {
    ToolTransaction selToolTransaction = new ToolTransaction();
    if (ToolTransactionGridView[0, e.RowIndex].Value.ToString() != _ToolTransactionList[0].Message) {
        selToolTransaction =
            _ToolTransactionProvider.SelectedToolTransactionById(Convert.ToInt32(
                ToolTransactionGridView["ToolTransactionId", e.RowIndex].Value.ToString()));
        _selTool = _ToolProvider.SelectedToolByToolId(selToolTransaction.ToolId);
        if (e.ColumnIndex == 4) {
            _ToolProvider.UpdateToolStatus(1, _selTool.ToolId);
            _ToolTransactionProvider.UpdateToolTransactionStatus(selToolTransaction.ToolTransactionId);
        } else if (e.ColumnIndex == 5) {
            _ToolProvider.UpdateToolStatus(1, _selTool.ToolId);
            _ToolTransactionProvider.DeleteToolTransactionById(selToolTransaction.ToolTransactionId);
        } else {
            UpdateToolTransactionForm updatePostsForm =
                new UpdateToolTransactionForm(Convert.ToInt32(ToolTransactionGridView[0, e.RowIndex].Value.ToString()));
            updatePostsForm.ShowDialog();
        }
        DataLoad();
        LoadAllDate();
    }
}

```

Рисунок 3.12 – Код події для управління виданими інструментами

Код визначає обробник події кліку на комірку у «ToolTransactionGridView», який реагує на взаємодію користувача з таблицею транзакцій інструментів. Спочатку створюється новий об'єкт «ToolTransaction». Після цього перевіряється, чи значення у першій комірці вибраного рядка не збігається з повідомленням у першому елементі списку транзакцій `_ToolTransactionList`. Якщо значення відрізняються, завантажується обрана транзакція інструменту за допомогою методу «SelectedToolTransactionById» провайдера `_ToolTransactionProvider` і зберігається у змінну `selToolTransaction`. Паралельно завантажуються дані про інструмент, пов'язаний з транзакцією, за допомогою методу «SelectedToolByToolId» провайдера `_ToolProvider` і зберігаються у змінну `_selTool`.

Далі код перевіряє, на яку комірку було натиснуто, зокрема:

- якщо натиснуто на комірку в колонці з індексом 4, статус інструменту оновлюється до значення 1 за допомогою методу «UpdateToolStatus,» і статус транзакції оновлюється методом «UpdateToolTransactionStatus»;

- якщо натиснуто на комірку в колонці з індексом 5, статус інструменту також оновлюється до значення 1, але транзакція видаляється за допомогою методу `DeleteToolTransactionById`;

– якщо натиснуто на будь-яку іншу комірку, відкривається форма для оновлення транзакції «UpdateToolTransactionForm», яка ініціалізується з ідентифікатором транзакції, вибраної у першій комірці рядка.

Після виконання відповідної дії дані перезавантажуються за допомогою методів «DataLoad» і «LoadAllDate», щоб відобразити актуальні дані у таблиці та випадаючих списках.

Для формування статистики у системі було реалізовано ряд запитів. Зокрема на рис. 3.13 показано метод «GetTotalMaintenanceCostByTool», який реалізує запит, щоб отримати загальні витрати на обслуговування для кожного інструменту.

```
string sqlString = @"
SELECT
    t.ToolId, t.ToolName,
    SUM(tm.Cost) AS TotalMaintenanceCost
FROM
    Tool t
JOIN
    ToolMaintenance tm ON t.ToolId = tm.ToolId
GROUP BY
    t.ToolId, t.ToolName
ORDER BY
    TotalMaintenanceCost DESC";
```

Рисунок 3.13 – Код запиту для отримання загальних витрат на обслуговування для кожного інструменту

У запиті використовується конструкція SELECT, яка вибирає ідентифікатор інструменту (ToolId), назву інструменту (ToolName) та сумарні витрати на обслуговування (SUM(tm.Cost)), об'єднуючи таблицю інструментів (Tool) з таблицею обслуговування інструментів (ToolMaintenance) за допомогою поля ToolId. Дані групуються за ідентифікатором та назвою інструменту, а результати сортуються за спаданням загальних витрат на обслуговування.

Метод «GetActiveTransactionsByCategory» виконує SQL-запит для отримання кількості активних транзакцій за кожною категорією інструментів (рис. 3.14).

```

string sqlString = @"
SELECT
    c.CategoryId, c.CategoryName,
    COUNT(tt.ToolTransactionId) AS ActiveTransactionsCount
FROM
    Category c
JOIN
    Tool t ON c.CategoryId = t.CategoryId
JOIN
    ToolTransaction tt ON t.ToolId = tt.ToolId
WHERE
    tt.Status = 0
GROUP BY
    c.CategoryId, c.CategoryName
ORDER BY
    ActiveTransactionsCount DESC";

```

Рисунок 3.14 – Код запиту для отримання кількості використання інструментів по кожній категорії

Запит вибирає ідентифікатор категорії (CategoryId), назву категорії (CategoryName) та підраховує кількість активних транзакцій (COUNT(tt.ToolTransactionId)), об'єднуючи таблиці Category, Tool та ToolTransaction за відповідними ключами. Умовою відбору є статус транзакції, який має дорівнювати 0, що вказує на активні транзакції. Результати групуються за ідентифікатором та назвою категорії, а потім сортуються за спаданням кількості активних транзакцій.

Метод «GetIssuedToolsByPeriod» виконує SQL-запит для отримання деталей про видані інструменти в певний період часу (рис. 3.15).

```

string sqlString = @"
SELECT
    t.ToolId, t.ToolName, tt.ToolTransactionId, tt.TransactionDate, e.EmployeeId,
    e.FirstName + ' ' + e.LastName AS EmployeeName,
    tt.Description
FROM
    Tool t
JOIN
    ToolTransaction tt ON t.ToolId = tt.ToolId
JOIN
    Employee e ON tt.EmployeeId = e.EmployeeId
WHERE
    tt.TransactionDate BETWEEN @StartDate AND @EndDate
ORDER BY
    tt.TransactionDate";

```

Рисунок 3.15 – Фрагмент код методу «GetIssuedToolsByPeriod»

Запит вибирає ідентифікатор інструменту (ToolId), назву інструменту (ToolName), ідентифікатор транзакції (ToolTransactionId), дату транзакції (TransactionDate), ідентифікатор працівника (EmployeeId), ім'я та прізвище працівника (EmployeeName), а також опис транзакції (Description). Таблиці Tool, ToolTransaction і Employee об'єднуються за відповідними ключами. Умовою вибірки є дата транзакції, яка повинна бути в межах між заданими датами @StartDate і @EndDate. Результати сортуються за датою транзакції у зростаючому порядку.

3.3 Експериментальна перевірка та тестування

Процес розробки системи автоматизації процесу зберігання та видачі електричних інструментів завершується етапом експериментальної перевірки та тестування. Основною метою цього етапу є виявлення та усунення можливих помилок, а також забезпечення належного рівня функціональності та надійності системи. Особлива увага приділяється тестуванню ключових компонентів системи, зокрема форм введення та обробки даних. Важливим елементом цього процесу є тестування форми «Інструменти», яке включає перевірку різних сценаріїв роботи. У табл. 3.1 наведено тестові сценарії для форми «Інструменти», що охоплюють основні випадки використання системи.

Таблиця 3.1 – Тестові сценарії для форми «Інструменти»

№	Сценарій	Вхідні дані	Очікуваний результат
1	2	3	4
1	Додавання нового інструменту з коректними даними	Назва: Дриль, Номер: DR001, Ціна: 1500, Категорія: Дрилі, Статус: в наявності, Опис: Потужний дриль	Інструмент додається до списку, поля очищуються
2	Спроба додавання інструменту без обов'язкового поля «Назва»	Номер: DR002, Ціна: 1500, Категорія: Дрилі, Статус: в наявності, Опис: Без назви	Поява повідомлення про помилку, інструмент не додається

Продовження табл. 3.1

1	2	3	4
3	Спроба додавання інструменту з невалідним значенням у полі «Ціна»	Назва: Лобзик, Номер: LB001, Ціна: -100, Категорія: Лобзики, Статус: в наявності, Опис: Не валідна ціна	Поява повідомлення про помилку, інструмент не додається
4	Очищення полів форми після введення даних	Назва: Шліфувальна машина, Номер: SM001, Ціна: 2000, Категорія: Шліфувальні машини, Статус: в наявності, Опис: Шліфувальна машина	Поля форми очищуються
5	Вихід з форми	-	Форма закривається
6	Редагування існуючого інструменту	Вибір існуючого інструменту, зміна значень	Зміни зберігаються, список оновлюється
7	Видалення існуючого інструменту	Вибір інструменту, натискання кнопки «Видалити»	Інструмент видаляється зі списку

Результат виконання тестових сценарії представлено на рис. 3.16.

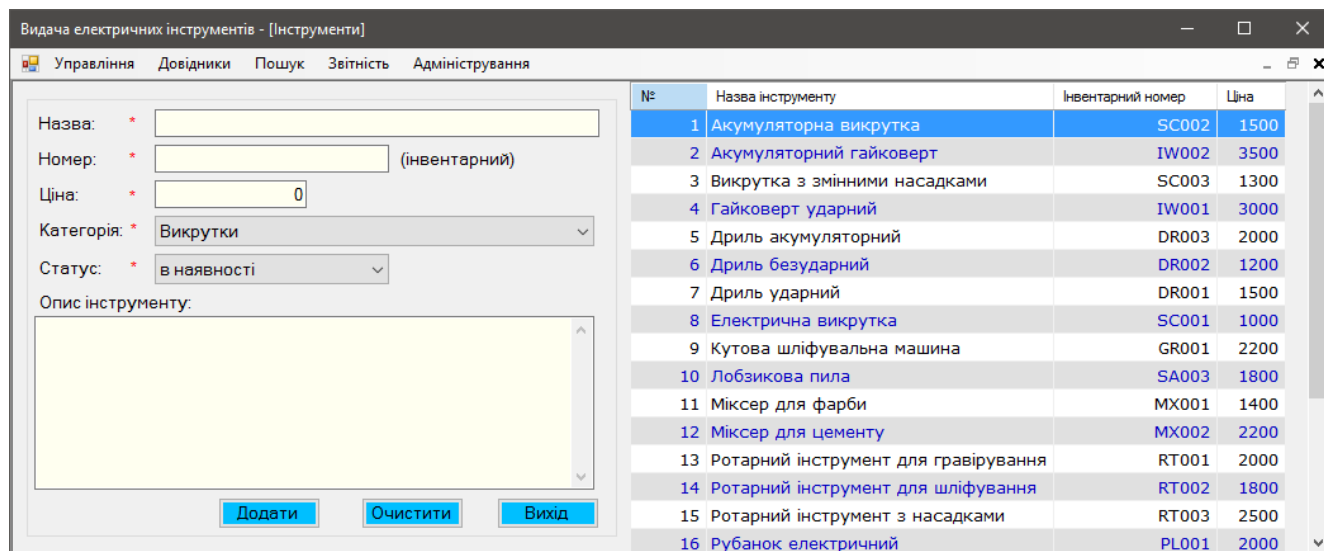


Рисунок 3.16 – Результати тестування форми «Інструменти»

Форма «Видача інструментів» є важливим елементом системи, яка дозволяє користувачам реєструвати видачу інструментів, керувати цими даними та взаємодіяти з базою даних. У табл. 3.2 наведено тестові сценарії для цієї форми.

Таблиця 3.2 – Тестові сценарії для форми «Видача інструментів»

№	Сценарій	Вхідні дані	Очікуваний результат
1	Додавання запису про видачу інструменту з коректними даними	Категорія: Викрутки, Інструмент: Викрутка з насадками, Працівник: Боровська Ярина, Дата видачі: 08.06.2024, Опис: -	Запис додається до списку, поля очищуються
2	Спроба додавання запису без обов'язкового поля «Інструмент»	Категорія: Викрутки, Працівник: Боровська Ярина, Дата видачі: 08.06.2024, Опис: -	Поява повідомлення про помилку, запис не додається
3	Спроба додавання запису з некоректною датою видачі	Категорія: Викрутки, Інструмент: Викрутка з насадками, Працівник: Боровська Ярина, Дата видачі: -	Поява повідомлення про помилку, запис не додається
4	Очищення полів форми після введення даних	Категорія: Викрутки, Інструмент: Викрутка з насадками, Працівник: Боровська Ярина, Дата видачі: 08.06.2024, Опис: -	Поля форми очищуються
5	Вихід з форми	-	Форма закривається
6	Повернення інструменту, відзначеного у списку	Вибір інструменту, натискання кнопки «Повернути»	Статус інструменту оновлюється на «в наявності», запис оновлюється
7	Видалення запису про видачу інструменту зі списку	Вибір інструменту, натискання кнопки «Видалити»	Запис видаляється зі списку

Ці тестові сценарії охопили основні функціональні можливості форми «Видача інструментів» і сприяли виявленню можливих помилок та забезпеченню стабільної роботи системи. На рис. 3.17 представлено результати виконання тестових сценаріїв для форми «Видача інструментів».

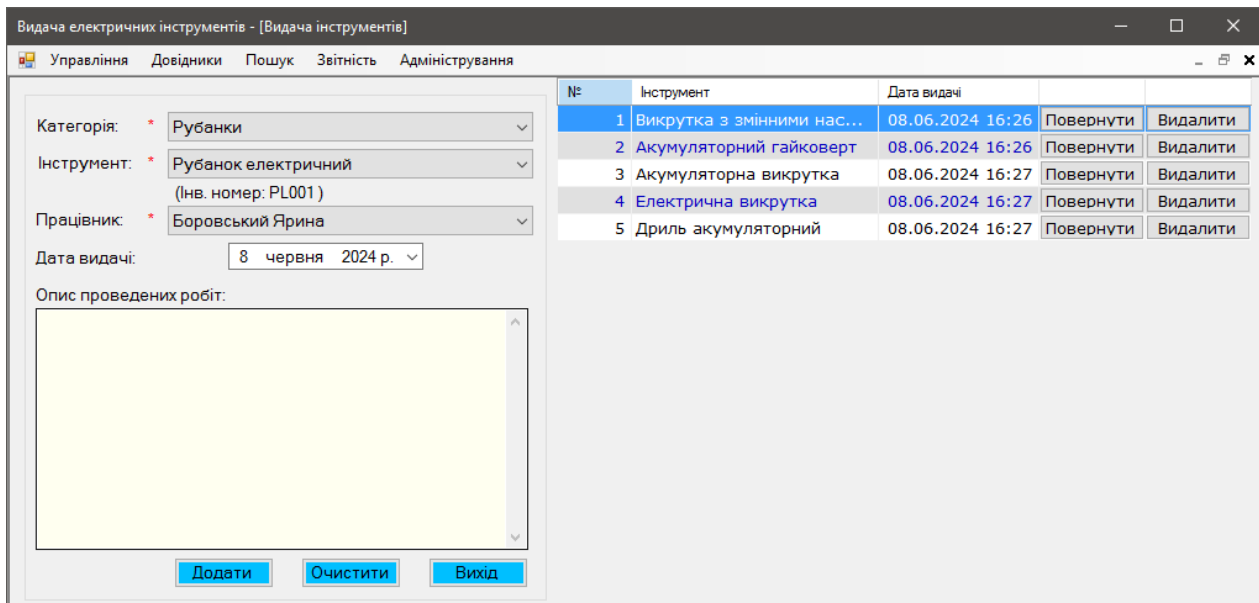


Рисунок 3.17 – Результати тестування форми «Видача інструментів»

Форма «Інструменти, що видані за період часу» дозволяє користувачам формувати звіт про видані інструменти в заданий період часу. Нижче у табл. 3.3 представлено тестові сценарії для цієї форми.

Таблиця 3.3 – Тестові сценарії для форми «Інструменти видані за період»

№	Сценарій	Вхідні дані	Очікуваний результат
1	Формування звіту з коректними датами	Початок періоду: 08.06.2024, Кінець періоду: 08.06.2024	Відображення звіту за вибраний період з коректними даними
2	Спроба формування звіту з початком періоду, що перевищує кінець періоду	Початок періоду: 09.06.2024, Кінець періоду: 08.06.2024	Поява повідомлення про помилку, звіт не формується
3	Формування звіту з відсутніми записами за вказаний період	Початок періоду: 01.06.2024, Кінець періоду: 01.06.2024	Відображення порожнього звіту, повідомлення про відсутність даних
4	Очищення полів форми після введення даних	Натискання кнопки "Очистити"	Поля форми очищуються
5	Вихід з форми	Натискання кнопки "Вихід"	Форма закривається

На рис. 3.18 представлено результати виконання тестових сценаріїв для цієї форми.

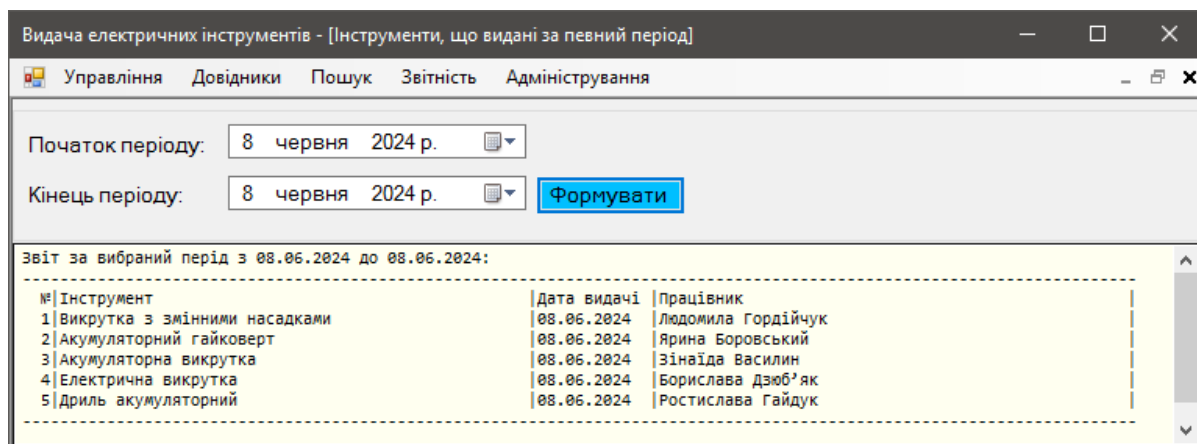


Рисунок 3.18 – Тестування форми «Інструменти, що видані за період часу»

Модульне тестування є важливим етапом розробки програмного забезпечення, яке забезпечує перевірку окремих модулів або компонентів системи на коректність їх роботи. На рис. 3.19 представлено результати проведеного модульного тестування розробленої системи (Unit Test).

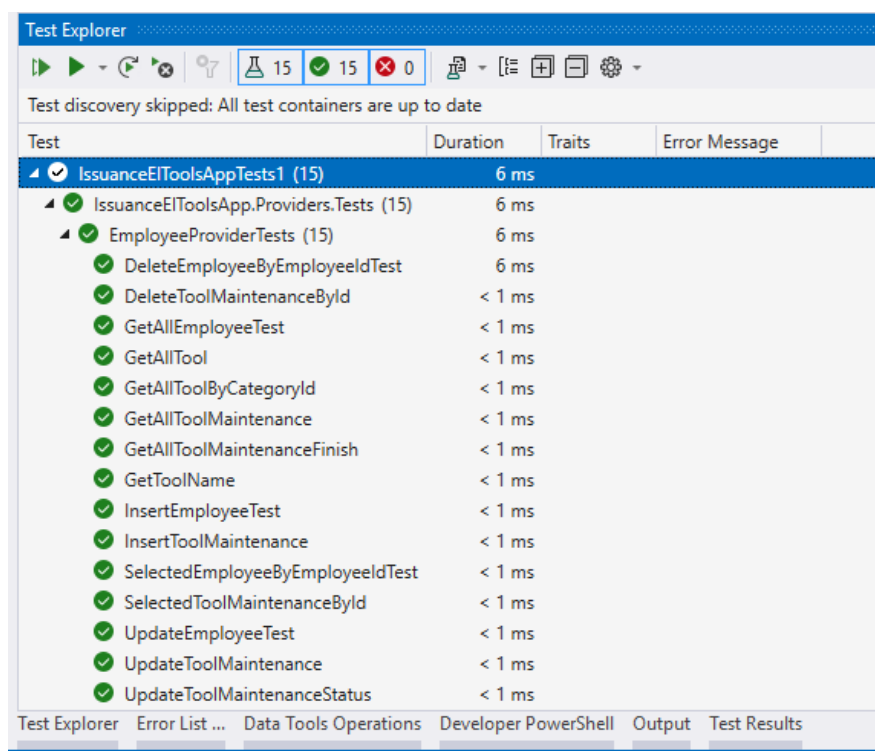
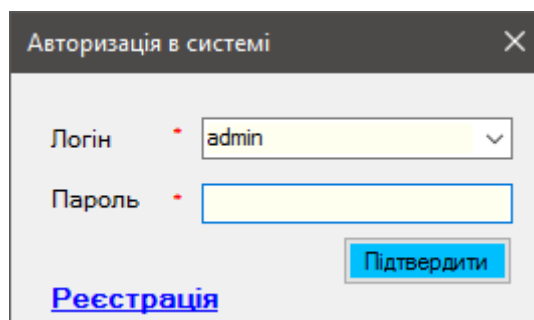


Рисунок 3.19 – Результати модульного тестування

За допомогою модульного тестування було виявлення помилок на ранніх стадіях розробки, що дозволило підвищити якість кінцевого продукту. Під час модульного тестування кожен модуль перевірявся незалежно від інших, що дозволяє ізолювати та ідентифікувати проблеми в конкретних частинах коду. Це тестування допомогло впевнитися в тому, що кожен модуль працює згідно зі своїми специфікаціями та виконує очікувані функції.

3.4 Інструкція користувача

Інструкція користувача містить детальні вказівки щодо використання різних функціональних можливостей системи, включаючи введення, редагування та видалення даних, а також формування звітів і взаємодію з базою даних. Першим кроком при роботі з системою є авторизація користувача, яка дозволяє забезпечити безпечний доступ до функціоналу системи. На рис. 3.20 представлено форму авторизації користувача, де необхідно ввести логін і пароль для доступу до системи.



The image shows a screenshot of a user authorization window titled "Авторизація в системі". The window has a dark header with a close button (X). Below the header, there are two input fields: "Логін" (Login) with a dropdown menu showing "admin" and "Пароль" (Password) with an empty text box. To the right of the password field is a blue button labeled "Підтвердити" (Confirm). At the bottom left, there is a blue link labeled "Реєстрація" (Registration).

Рисунок 3.20 – Вікно авторизації користувача

Для обліку видачі електричних інструментів необхідно заповнити систему інформацією про категорії інструментів, інструменти та працівників. Це забезпечить коректну роботу всіх функціональних можливостей системи та дозволить ефективно керувати процесом видачі та повернення інструментів. Одним з перших кроків є введення даних про категорії інструментів.

Форма для введення та перегляду категорій інструментів (рис. 3.21) дозволяє користувачам додавати нові категорії та переглядати наявні. Для додавання нової категорії необхідно заповнити поля "Назва" та "Опис" і натиснути кнопку "Додати". Усі категорії відображаються у списку праворуч, що дозволяє зручно переглядати наявні категорії.

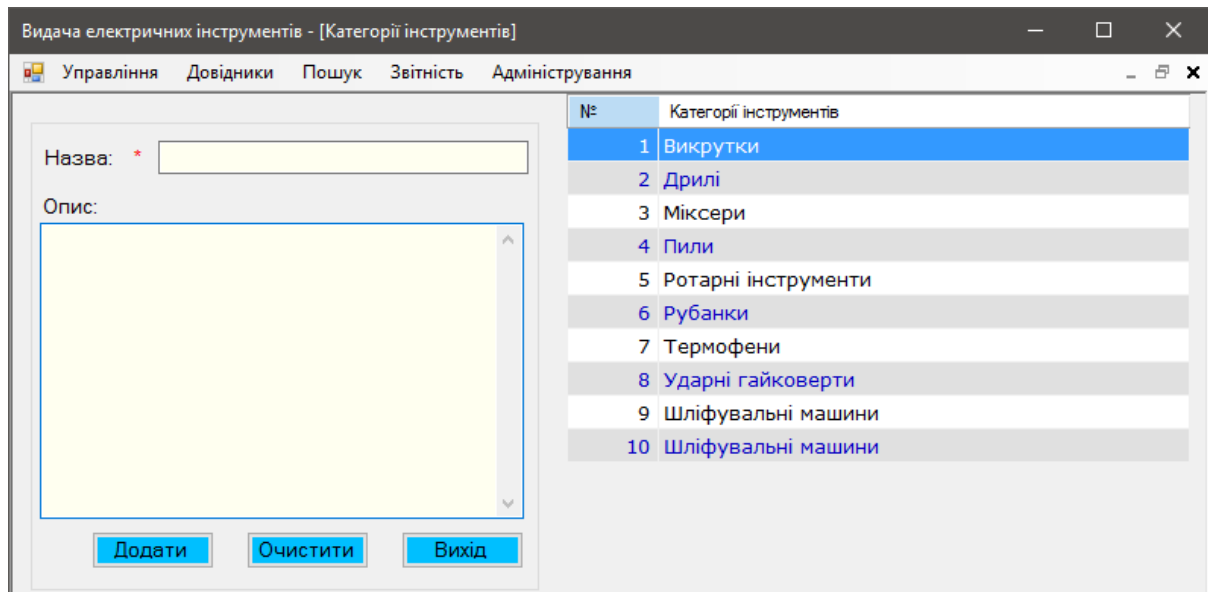


Рисунок 3.21 – Вікно для введення та перегляду категорій

Діалогове вікно для редагування категорії, представлене на рис. 3.22, містить поля "Назва" та "Опис", заповнені поточними значеннями вибраної категорії. Користувач може змінити ці значення та натиснути кнопку "Зберегти" для збереження змін. Крім того, у діалоговому вікні є кнопка "Видалити", яка дозволяє видалити категорію зі списку, та кнопка "Вихід" для закриття вікна без внесення змін. Це діалогове вікно забезпечує зручний інтерфейс для редагування та управління категоріями інструментів у системі.

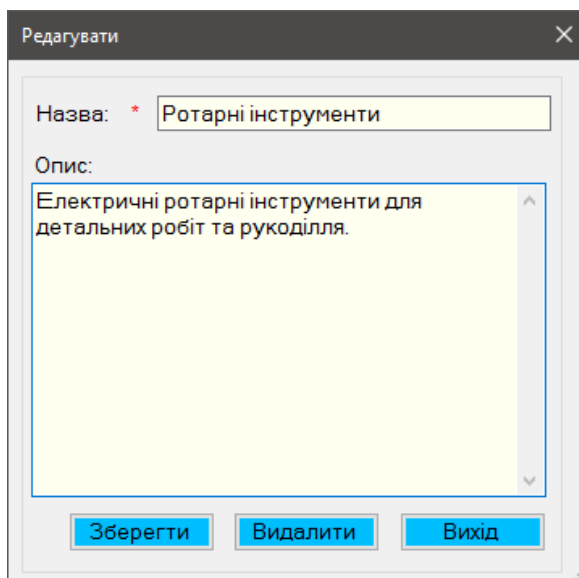
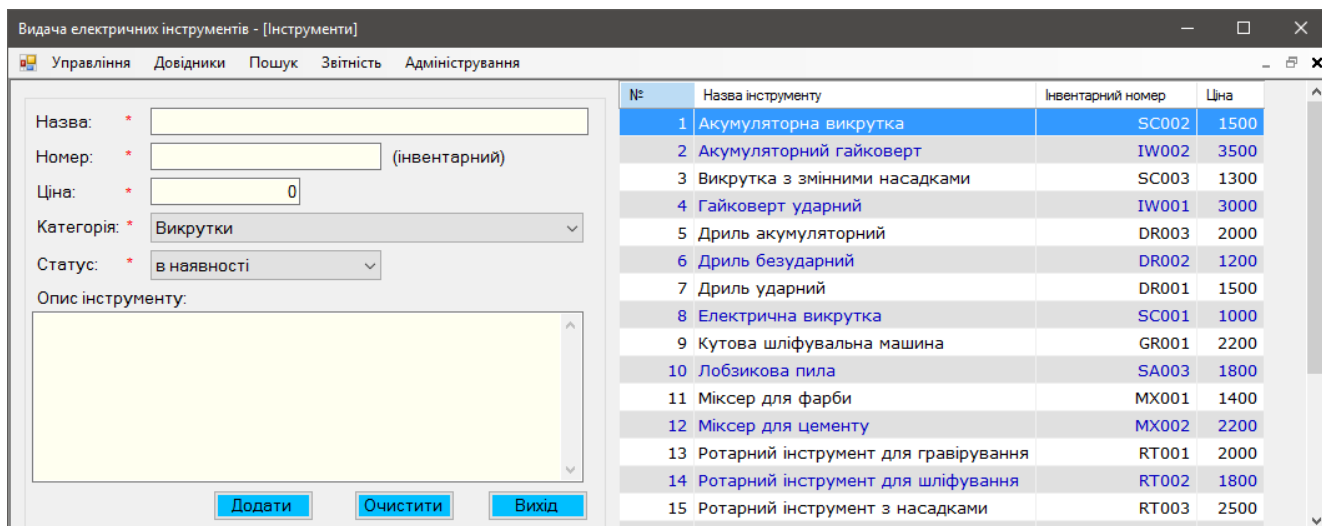


Рисунок 3.22 – Форма редагування категорій інструментів

Для управління інструментами система передбачає використання спеціальної форми, яка відкривається при переході по меню "Довідники" -> "Інструменти" (рис. 3.23). Ця форма дозволяє користувачам додавати нові інструменти, а також переглядати наявні в базі даних інструменти.



№	Назва інструменту	Інвентарний номер	Ціна
1	Акумуляторна викрутка	SC002	1500
2	Акумуляторний гайковерт	IW002	3500
3	Викрутка з змінними насадками	SC003	1300
4	Гайковерт ударний	IW001	3000
5	Дриль акумуляторний	DR003	2000
6	Дриль безударний	DR002	1200
7	Дриль ударний	DR001	1500
8	Електрична викрутка	SC001	1000
9	Кутова шліфувальна машина	GR001	2200
10	Лобзикова пила	SA003	1800
11	Міксер для фарби	MX001	1400
12	Міксер для цементу	MX002	2200
13	Ротарний інструмент для гравірування	RT001	2000
14	Ротарний інструмент для шліфування	RT002	1800
15	Ротарний інструмент з насадками	RT003	2500

Рисунок 3.23 – Вікно для додавання та перегляду інструментів

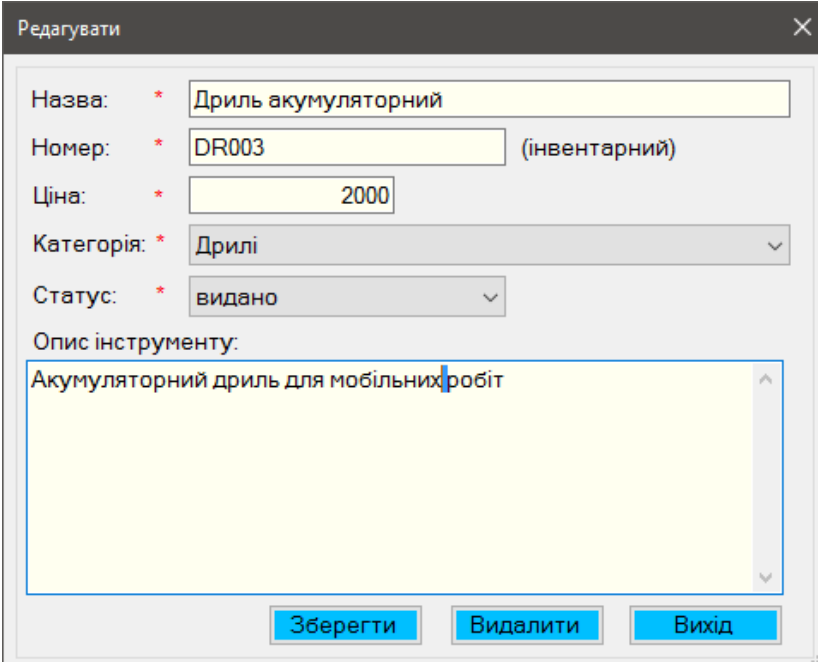
Форма для додавання та перегляду інструментів, зображена на рис. 3.20, містить кілька полів для введення інформації про інструмент, включаючи "Назва", "Номер" (інвентарний), "Ціна", "Категорія", "Статус" і "Опис інструменту". Для

додавання нового інструменту користувачу необхідно заповнити ці поля та натиснути кнопку "Додати". Всі додані інструменти відображаються у таблиці праворуч, де показані назва інструменту, інвентарний номер і ціна.

Користувач може також очистити всі поля введення за допомогою кнопки "Очистити" або закрити форму, натиснувши кнопку "Вихід". Ця форма забезпечує зручний інтерфейс для введення та управління інформацією про інструменти, що дозволяє ефективно контролювати процес їх видачі та повернення.

При натисканні на запис у таблиці інструментів відкривається діалогове вікно для редагування інформації про обраний інструмент. Це діалогове вікно дозволяє користувачу вносити зміни до існуючих даних про інструмент або видаляти його з бази даних.

Діалогове вікно для редагування інструменту, представлене на рис. 3.24, містить кілька полів для введення інформації, включаючи "Назва", "Номер" (інвентарний), "Ціна", "Категорія", "Статус" і "Опис інструменту". Поля заповнені поточними значеннями вибраного інструменту, що дозволяє зручно редагувати необхідні дані. Після внесення змін користувач може натиснути кнопку "Зберегти" для збереження оновленої інформації.



Редагувати

Назва: * Дриль акумуляторний

Номер: * DR003 (інвентарний)

Ціна: * 2000

Категорія: * Дрилі

Статус: * видано

Опис інструменту:

Акумуляторний дриль для мобільних робіт

Зберегти Видалити Вихід

Рисунок 3.24 – Форма редагування інструментів

Крім того, у діалоговому вікні є кнопка "Видалити", яка дозволяє видалити інструмент з бази даних, та кнопка "Вихід" для закриття вікна без внесення змін. Це діалогове вікно забезпечує зручний інтерфейс для управління інформацією про інструменти, дозволяючи редагувати та видаляти записи за потребою.

Для управління працівниками система передбачає використання спеціальної форми, яка відкривається при переході по меню "Довідники" -> "Працівники". Ця форма дозволяє користувачам додавати нових працівників, а також переглядати наявних у базі даних працівників.

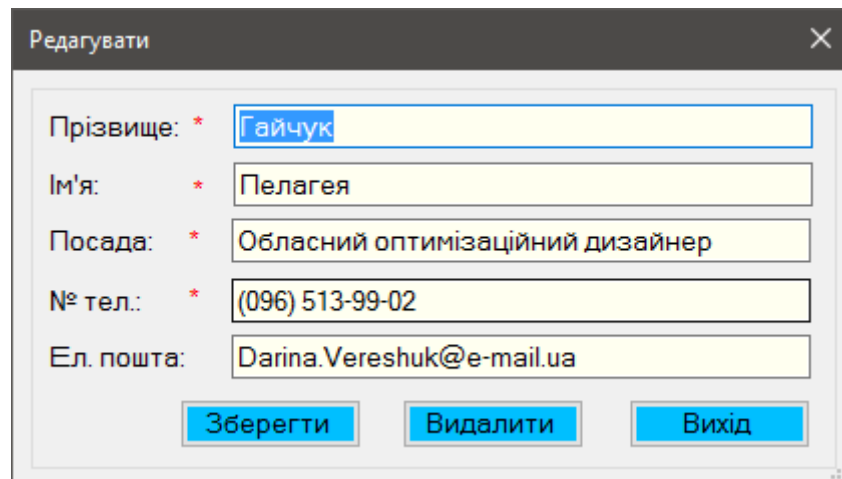
Форма для додавання та перегляду працівників, зображена на рис. 3.25, містить кілька полів для введення інформації про працівника, включаючи "Прізвище", "Ім'я", "Посада", "№ тел." і "Ел. пошта". Для додавання нового працівника користувачу необхідно заповнити ці поля та натиснути кнопку "Додати". Усі додані працівники відображаються у таблиці праворуч, де показані прізвище, ім'я та номер телефону.

№	Прізвище	Ім'я	№ телефону
1	Боровський	Ярина	(094) 528-78-65
2	Василин	Зінаїда	(044) 911-35-51
3	Гайдук	Ростислава	(091) 953-87-00
4	Гайчук	Пелагея	(096) 513-99-02
5	Галаєнко	Захар	(097) 608-67-96
6	Гнатишина	Геннадій	(096) 842-24-15
7	Горбач	Гордій	(092) 558-04-78
8	Гордійчук	Людмила	(097) 479-44-10
9	Гречко	Архип	(096) 957-19-95
10	Гриневський	Микола	(096) 826-57-69

Рисунок 3.25 – Вікно для додавання та перегляду працівників

Користувач може також очистити всі поля введення за допомогою кнопки "Очистити" або закрити форму, натиснувши кнопку "Вихід". Ця форма забезпечує зручний інтерфейс для введення та управління інформацією про працівників, що дозволяє ефективно контролювати дані про персонал, який користується інструментами в системі.

При натисканні на запис у таблиці працівників відкривається діалогове вікно для редагування інформації про обраного працівника (рис. 3.26). Це діалогове вікно дозволяє користувачу вносити зміни до існуючих даних про працівника або видаляти його з бази даних.



Прізвище: *	Гайчук
Ім'я: *	Пелагея
Посада: *	Обласний оптимізаційний дизайнер
№ тел.: *	(096) 513-99-02
Ел. пошта:	Darina.Vereshuk@e-mail.ua

Зберегти Видалити Вихід

Рисунок 3.26 – Форма редагування даних працівників

Для управління процесом видачі інструментів система передбачає використання спеціальної форми, яка відкривається при переході по меню "Управління" -> "Видача інструментів". Ця форма дозволяє користувачам реєструвати видачу інструментів працівникам, а також переглядати наявні записи про видані інструменти.

Форма для видачі інструментів, зображена на рис. 3.27, містить кілька полів для введення інформації про інструмент, включаючи "Категорія", "Інструмент" (та його інвентарний номер), "Працівник", "Дата видачі" та "Опис проведених робіт". Для реєстрації видачі нового інструменту користувачу необхідно заповнити ці поля та натиснути кнопку "Додати". Усі видані інструменти відображаються у таблиці праворуч, де показані інструмент, дата видачі та кнопки для повернення або видалення запису.

№	Інструмент	Дата видачі	Повернути	Видалити
1	Викрутка з змінними насадками	08.06.2024 16:26	Повернути	Видалити
2	Акумуляторний гайковерт	08.06.2024 16:26	Повернути	Видалити
3	Акумуляторна викрутка	08.06.2024 16:27	Повернути	Видалити
4	Електрична викрутка	08.06.2024 16:27	Повернути	Видалити
5	Дриль акумуляторний	08.06.2024 16:27	Повернути	Видалити

Рисунок 3.27 – Вікно для опрацювання даних видачі інструментів

Користувач може також очистити всі поля введення за допомогою кнопки "Очистити" або закрити форму, натиснувши кнопку "Вихід". Ця форма забезпечує зручний інтерфейс для реєстрації та управління процесом видачі інструментів, дозволяючи ефективно контролювати використання інструментів працівниками.

Для управління процесом обслуговування інструментів система передбачає використання спеціальної форми, яка відкривається при переході по меню "Управління" -> "Обслуговування". Ця форма дозволяє користувачам реєструвати обслуговування інструментів, а також переглядати наявні записи про проведене обслуговування.

Форма для обслуговування інструментів, зображена на рис. 3.28, містить кілька полів для введення інформації про обслуговування, включаючи "Категорія", "Інструмент" (та його інвентарний номер), "Дата обслуговування", "Ціна" та "Опис проведених робіт". Для реєстрації нового обслуговування користувачу необхідно заповнити ці поля та натиснути кнопку "Додати". Усі записи про обслуговування відображаються у таблиці праворуч, де показані інструмент, дата обслуговування, ціна та кнопки для позначення інструменту як відремонтованого або видалення запису.

№	Інструмент	Дата обслуговування	Ціна	Видалити
1	Кутова шліфувальна машина	08.06.2024 18:57	50	Відремонтовано Видалити

Рисунок 3.28 – Вікно для опрацювання даних обслуговування інструментів

Користувач може також очистити всі поля введення за допомогою кнопки "Очистити" або закрити форму, натиснувши кнопку "Вихід". Ця форма забезпечує зручний інтерфейс для реєстрації та управління процесом обслуговування інструментів, дозволяючи ефективно контролювати їх стан та своєчасне технічне обслуговування.

Для перегляду історії видачі інструментів система передбачає використання спеціальної форми, яка відкривається при переході по меню "Управління" -> "Історія видачі". Ця форма дозволяє користувачам переглядати всі записи про видачу та повернення інструментів, що були здійснені в системі.

Форма для історії видачі інструментів, зображена на рис. 3.29, містить таблицю, яка включає такі поля: "Інструмент", "Дата видачі", "Дата повернення" та "Примітки". Таблиця відображає повну інформацію про кожну транзакцію видачі та повернення інструментів, включаючи дати та примітки, що дозволяє користувачам легко відстежувати всі операції з інструментами.

№	Інструмент	Дата видачі	Дата повернення	Примітки
1	Електрична викрутка	02.06.2024 20:13	04.06.2024 20:13	
2	Кутова шліфувальна машина	02.06.2024 20:13	04.06.2024 20:13	
3	Викрутка з змінними насадками	02.06.2024 20:13	04.06.2024 20:13	
4	Ротарний інструмент для гравірування	02.06.2024 20:12	04.06.2024 20:12	
5	Акумуляторна викрутка	02.06.2024 20:12	04.06.2024 20:12	
6	Ротарний інструмент для гравірування	23.05.2024 17:24	01.06.2024 3:52	Deleniti ex et possimus velit et ut recusandae perspiciatis aut.
7	Міксер для фарби	09.05.2024 3:42	31.05.2024 2:22	Error inventore sint rem facilis eaque voluptates.
8	Ротарний інструмент для гравірування	22.05.2024 7:45	30.05.2024 15:34	Et quam alias magni aut tenetur molestiae occaecati.
9	Дриль безударний	30.06.2023 4:45	28.05.2024 17:32	Iste aut earum dolore accusantium eaque voluptas.
10	Шліфувальна машина по дереву	11.12.2023 7:16	27.05.2024 9:50	Nemo quis voluptates animi et laborum ad itaque.
11	Термофен промисловий	07.03.2024 15:23	27.05.2024 5:24	Eos earum omnis sequi ullam eius animi rerum sit.
12	Термофен будівельний	04.09.2023 2:20	24.05.2024 18:23	Error rem aperiam ipsam voluptates aut et.
13	Рубанок з регулюванням глибини	28.11.2023 1:20	24.05.2024 15:53	Dicta omnis ut et corporis vitae vitae.
14	Електрична викрутка	04.02.2024 22:11	23.05.2024 8:18	Pariatur ratione tempore voluptatum nam delectus vel.
15	Ротарний інструмент для гравірування	27.12.2023 19:16	22.05.2024 18:17	Nisi nemo reiciendis asperiores nam quasi ut in iste.

Рисунок 3.29 – Вікно для перегляду історії видачі інструментів

Для перегляду історії обслуговування інструментів система передбачає використання спеціальної форми, яка відкривається при переході по меню "Управління" -> "Історія обслуговування". Ця форма дозволяє користувачам переглядати всі записи про проведені обслуговування інструментів, що були здійснені в системі.

Форма для історії обслуговування інструментів, зображена на рис. 3.30, містить таблицю, яка включає такі поля: "Інструмент", "Дата обслуговування", "Ціна" та "Опис". Таблиця відображає повну інформацію про кожне обслуговування інструментів, включаючи дати, вартість та опис проведених робіт, що дозволяє користувачам легко відстежувати всі операції з обслуговування інструментів.

№	Інструмент	Дата обслуговування	Ціна	Опис
1	Ротарний інструмент для гравірування	17.06.2023 6:36	439,48	Qui accusantium repudiandae impedit autem.
2	Акумуляторний гайковерт	20.06.2023 12:17	156,87	Aut in ut maxime voluptates voluptas facere excepturi.
3	Термофен будівельний	26.06.2023 3:47	488,03	Mollitia recusandae quidem voluptatem accusantium.
4	Термофен промисловий	28.06.2023 14:18	811,76	Voluptate molestias velit et similique.
5	Рубанок з регулюванням глибини	16.07.2023 3:19	929,62	Consequatur nihil ut accusantium voluptate hic.
6	Ротарний інструмент для шліфування	19.07.2023 23:28	407,31	Laudantium placeat magnam neque.
7	Термофен промисловий	10.08.2023 1:10	702,08	Laborum rerum perspiciatis ut ipsa aspernatur asperiores quasi.
8	Акумуляторний гайковерт	20.08.2023 22:50	294,22	Quis voluptas officia.
9	Ротарний інструмент для гравірування	21.08.2023 17:39	330,63	Blanditiis et est quia quis quaerat quaerat nihil.
10	Термофен будівельний	23.08.2023 8:50	658,76	Voluptatem in eaque cum nemo adipisci exercitationem dignissimos.
11	Дриль безударний	24.08.2023 12:57	619,18	Nihil illo et facilis consequatur voluptate cum.
12	Лобзикова пила	26.08.2023 7:12	968,95	Tenetur rerum iure.
13	Шліфувальна машина по дереву	20.09.2023 17:23	285,87	Quo eaque illo quod eos expedita non ut perferendis cumque.
14	Рубанок електричний	04.10.2023 3:46	161,07	Aut ullam saepe quia vel tempora voluptatem.
15	Циркулярна пила	20.10.2023 17:23	465,48	Voluptate doloribus nisi deleniti illo dolor rem deleniti cupiditate incidunt.

Рисунок 3.30 – Вікно для перегляду історії обслуговування інструментів

Ця форма забезпечує зручний інтерфейс для перегляду історії обслуговування інструментів, дозволяючи користувачам отримувати детальну інформацію про технічне обслуговування та ремонти інструментів у минулому. Це сприяє покращенню контролю за станом інструментів та плануванню їх майбутнього обслуговування.

Для полегшення пошуку інструментів у системі передбачена спеціальна форма, яка відкривається при переході по меню "Пошук" -> "Інструментів". Ця форма дозволяє користувачам швидко знаходити необхідні інструменти за допомогою ключових слів або фраз.

Форма для пошуку інструментів, зображена на рис. 3.31, містить поле для введення пошукової фрази та кнопку "Знайти". Користувачу необхідно ввести ключове слово або частину назви інструменту в поле для введення та натиснути кнопку "Знайти". У таблиці нижче відобразяться результати пошуку, включаючи назву інструменту, інвентарний номер і ціну.

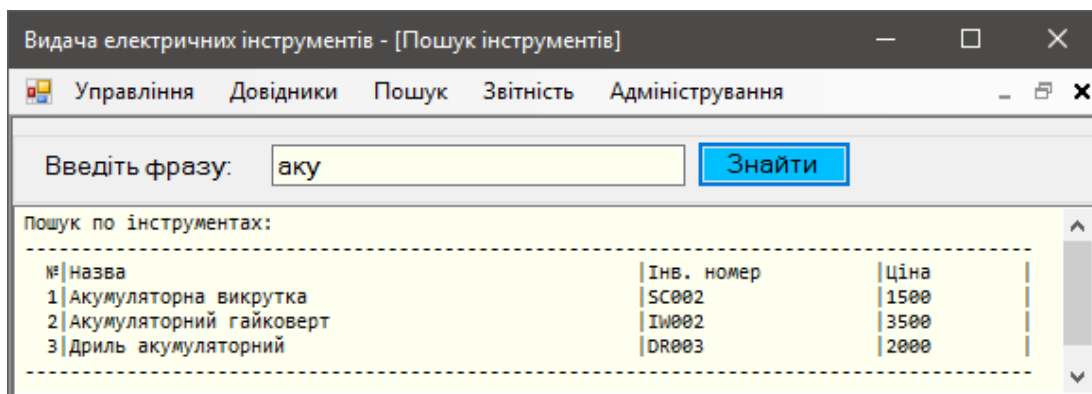


Рисунок 3.31 – Вікно для пошуку інструментів

Ця форма забезпечує зручний інтерфейс для швидкого пошуку інструментів у базі даних, що дозволяє користувачам легко знаходити потрібні інструменти та переглядати їх характеристики. Це сприяє ефективному управлінню інструментами та скороченню часу на їх пошук.

Форма для звітності, зображена на рис. 3.32, містить таблицю, яка включає такі поля: "Назва" інструменту та "Загальна сума" витрат на його обслуговування. Таблиця відображає повну інформацію про кожен інструмент, що дозволяє користувачам легко відстежувати загальні витрати на обслуговування та ремонти.

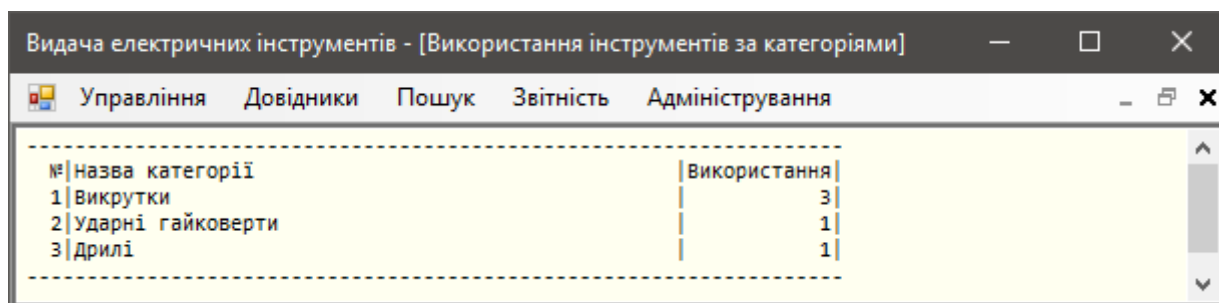
№	назва	Загальна сума
1	Гайковерт ударний	2513,31
2	Рубанок електричний	2379,06
3	Рубанок з регулюванням глибини	2243,61
4	Термофен промисловий	2236,45
5	Шліфувальна машина по дереву	2123,43
6	Термофен будівельний	2077,12
7	Лобзикова пила	2005,08
8	Викрутка з змінними насадками	2004,31
9	Міксер для фарби	1708,32
10	Ротарний інструмент для шліфування	1638,42
11	Ротарний інструмент для гравірування	1455,86
12	Міксер для цементу	1269,97
13	Дриль безударний	895,6
14	Дриль акумуляторний	797,95
15	Кутова шліфувальна машина	705,71
16	Циркулярна пила	465,48
17	Акумуляторний гайковерт	451,09
18	Ротарний інструмент з насадками	320,66
19	Шліфувальна машина для бетону	299,34
Загальна сума:		27590,77

Рисунок 3.32 – Формування звітності по загальній вартості обслуговування в гривнях

Ця форма забезпечує зручний інтерфейс для перегляду загальної вартості обслуговування інструментів, дозволяючи користувачам отримувати необхідні дані для аналізу витрат. Внизу таблиці відображається загальна сума витрат на обслуговування всіх інструментів, що сприяє кращому плануванню бюджету та ресурсів.

Для формування звітності по використанню інструментів за категоріями система передбачає використання спеціальної форми, яка відкривається при переході по меню "Звітність" -> "Використання інструментів за категоріями". Ця форма дозволяє користувачам отримати детальну інформацію про кількість використань інструментів, згрупованих за категоріями.

Форма для звітності, зображена на рис. 3.33, містить таблицю, яка включає такі поля: "Назва категорії" та "Використання". Таблиця відображає інформацію про кількість використань інструментів для кожної категорії, що дозволяє користувачам легко відстежувати, які категорії інструментів використовуються найчастіше.



№ назва категорії	Використання
1 Викрутки	3
2 ударні гайковерти	1
3 дрилі	1

Рисунок 3.33 – Використання інструментів за категоріями

Ця форма забезпечує зручний інтерфейс для перегляду використання інструментів за категоріями, дозволяючи користувачам отримувати необхідні дані для аналізу частоти використання різних типів інструментів. Це сприяє кращому плануванню закупівель та обслуговування інструментів на основі їх фактичного використання.

Для формування звітності по інструментах, що видані за певний період, система передбачає використання спеціальної форми, яка відкривається при

переході по меню "Звітність" -> "Інструменти, що видані за певний період". Ця форма дозволяє користувачам отримати інформацію про всі інструменти, що були видані в заданий часовий інтервал.

Форма для звітності, зображена на рис. 3.34, містить поля для вибору початкової та кінцевої дати періоду, а також кнопку "Формувати". Користувачу необхідно вказати потрібний період, після чого натиснути кнопку "Формувати" для генерації звіту. У таблиці нижче відобразяться результати, включаючи назву інструменту, дату видачі, працівника, який отримав інструмент, та інші деталі.

№ Інструмент	Дата видачі	Працівник
1 Викрутка з змінними насадками	08.06.2024	Людмила Гордійчук
2 Акумуляторний гайковерт	08.06.2024	Ярина Боровський
3 Акумуляторна викрутка	08.06.2024	Зінаїда Василен
4 Електрична викрутка	08.06.2024	Борислава Дзюб'як
5 Дриль акумуляторний	08.06.2024	Ростислава Гайдук

Рисунок 3.34 – Інструменти, що видані за певний період часу

Ця форма забезпечує зручний інтерфейс для перегляду інформації про інструменти, видані за конкретний період, що дозволяє користувачам ефективно відстежувати та аналізувати використання інструментів у різні періоди часу. Це сприяє покращенню управління інструментами та оптимізації їх використання.

3.5 Висновок

У рамках даного розділу було здійснено всебічну розробку та верифікацію програмного забезпечення для автоматизації процесу зберігання та видачі електричних інструментів. Спочатку були розроблені основні алгоритми, включаючи блок-схеми алгоритмів авторизації користувача, додавання,

редагування та видалення електричного інструменту. Ці алгоритми забезпечують надійну та безперебійну роботу системи, дозволяючи виконувати ключові операції з мінімальними витратами часу та ресурсів.

Було створено програмні модулі системи, що включають розробку користувацьких форм та опис основних методів. Зокрема, були розроблені форми для додавання, перегляду, редагування та видалення інструментів, форми для обслуговування інструментів, а також форми для ведення історії видачі та обслуговування. Для кожної форми було описано основні методи, що забезпечують їхню функціональність.

Експериментальна перевірка та тестування включала створення тестових сценаріїв для всіх форм, що забезпечують правильність роботи системи за різних умов. Було проведено модульне тестування, результати якого підтвердили коректність реалізації основних функціональних можливостей. Модульне тестування дозволило виявити та виправити помилки на ранніх стадіях, що підвищило надійність та стабільність роботи системи.

Крім того, була створена детальна інструкція користувача, яка включає опис усіх функціональних можливостей системи та надає покрокові вказівки щодо роботи з нею. Це дозволить користувачам швидко освоїти роботу з програмним забезпеченням та ефективно використовувати його для автоматизації процесу зберігання та видачі електричних інструментів.

4 АВТОМАТИЗАЦІЯ СИСТЕМИ

4.1 Технічні аспекти автоматизації

Автоматизація обліку електричних інструментів базується на використанні сучасних технологій для збору, обробки та аналізу даних. Один з ключових аспектів – це впровадження RFID (Radio Frequency Identification) або QR-кодів для кожного інструменту. Ці мітки дозволяють швидко ідентифікувати інструмент, забезпечуючи точність та швидкість введення даних в систему.

Отже перший етап автоматизації це впровадження RFID або QR-кодів. На кожен інструмент наклеюється унікальна мітка RFID або QR-код. Ці мітки скануються при видачі інструменту працівникам та при його поверненні. Це забезпечує автоматичне оновлення записів у базі даних та дозволяє відстежувати місцезнаходження інструменту в реальному часі. RFID мітки найчастіше мають невеликий розмір та ціну. У середньому ціна за 1 мітку складає близько 10 гривень.



Рисунок 4.1 – RFID мітка діаметром 25мм

Інтегрування таких технологій у програму дозволить налаштувати систему таким чином, щоб вона автоматично нагадувала необхідність проведення профілактичного обслуговування або заміни інструменту.

Сканування може бути здійснено за допомогою рамок кодів, що зчитують рфід. Такі рамки можуть бути встановлені на вході/виході зі складу, а також на вході/виході з території виробництва. Подібні системи реалізовані у магазинах, проте там вони виконують лише захисну функцію.

4.2 Переваги автоматизованого обліку

а) Підвищення точності та швидкості обліку

Завдяки скануванню RFID або QR-кодів облік інструментів стає майже миттєвим, що економить час і зусилля працівників.

б) Зниження витрат

Ефективне відстеження стану і місцезнаходження інструментів допомагає уникнути надмірних витрат на їхнє заміщення та ремонт. Своєчасне обслуговування інструментів на основі фактичних даних дозволяє продовжити їхній термін служби та зменшити кількість простоїв через несправності.

в) Підвищення продуктивності

Завдяки автоматизованому обліку працівники можуть більше часу приділяти безпосередньо своїм основним обов'язкам, а не обліку і пошуку інструментів. Крім того, системи автоматизації забезпечують швидкий доступ до інформації про стан інструментів, що дозволяє оперативно реагувати на будь-які проблеми.

г) Покращення управління ресурсами

Автоматизовані системи дозволяють більш ефективно управляти інструментами, надаючи керівникам повну картину їхнього використання. Це

допомагає оптимізувати процеси закупівлі, обслуговування та утилізації інструментів, що в кінцевому підсумку підвищує загальну продуктивність підприємства.

Автоматизація обліку електричних інструментів є важливим кроком на шляху до модернізації виробництва. Це не лише полегшує управління ресурсами, але й сприяє підвищенню ефективності та зниженню витрат, що робить підприємство більш конкурентоспроможним на ринку.

4.3 Інтеграція системи RFID у програму

Прив'язка системи RFID міток до програми на Windows, написаної на C#, включає декілька важливих кроків, від вибору обладнання до інтеграції програмного забезпечення. Нижче описано ці кроки докладніше.

а) Вибір RFID обладнання

Для інтеграції RFID системи з програмою на C# знадобиться:

- RFID мітки: електронні мітки, які можна прикріпити до інструментів;
- RFID рідери (зчитувачі): пристрої, які можуть зчитувати інформацію з RFID міток;
- антени: використовуються разом з RFID рідерами для покращення якості зчитування.

Вибір обладнання залежить від потреб та масштабів виробництва, зокрема відстані зчитування, кількості інструментів та середовища використання.

б) Підключення RFID рідера до комп'ютера

Більшість сучасних RFID рідерів підключаються до комп'ютера через USB або серійний порт (RS232). Деякі рідери також підтримують бездротове підключення через Wi-Fi або Bluetooth.

1. Підключення через USB:

- Підключити рідер до комп'ютера через USB порт.
- Встановити необхідні драйвери, які постачаються з рідером або доступні на веб-сайті виробника.

2. Підключення через серійний порт:

- Підключити рідер до серійного порту комп'ютера.
- Перевірити, який COM порт використовується для підключення, це знадобиться для налаштування програмного забезпечення.

в) Встановлення та налаштування SDK рідера

Багато виробників RFID рідерів надають SDK (Software Development Kit) для інтеграції свого обладнання з програмами. SDK містить необхідні бібліотеки та приклади коду, які полегшують роботу з рідером у програмі на C#.

- Завантаження та встановлення SDK з веб-сайту виробника.
- Перевірка документації SDK для отримання інформації про підключення та використання функцій зчитування.

5 ОХОРОНА ПРАЦІ

5.1 Загальні питання охорони праці

Основним законодавчим документом, що регулює організацію заходів з охорони праці в Україні, є Закон України «Про охорону праці» [31]. Відповідно до цього закону, всі компанії зобов'язані виконувати певні правила та стандарти для забезпечення безпеки та здоров'я своїх працівників.

Стаття 4 цього закону встановлює десять ключових принципів, що лежать в основі державної політики у сфері охорони праці. Закон підкреслює важливість пріоритету збереження життя та здоров'я працівників, вимагає від роботодавців забезпечення безпечних умов праці та надає велике значення промисловій безпеці.

Стаття 14 зобов'язує працівників дотримуватися вимог нормативно-правових актів з охорони праці. Працівники повинні ознайомитися з цими вимогами, дотримуватися їх, проходити медичні огляди та піклуватися про власну безпеку і здоров'я, а також про безпеку і здоров'я оточуючих.

5.2 Вплив автоматизації на безпеку праці

Автоматизація процесів має значний вплив на зниження ризику людських помилок, що можуть призвести до аварій чи інших небезпечних ситуацій. Збір інформації про несправності інструменту допомагають уникнути помилок, пов'язаних з людським фактором. Завдяки цьому знижується ймовірність виникнення інцидентів, які можуть мати серйозні наслідки для безпеки працівників.

Крім того, автоматизовані системи здатні постійно контролювати стан обладнання та навколишнього середовища. Це включає моніторинг якості повітря, температури, освітлення, що можуть впливати на безпеку праці. Вчасне виявлення небезпечних умов дозволяє оперативно вжити необхідних

заходів для запобігання нещасним випадкам, що суттєво підвищує загальний рівень безпеки на підприємстві.

Автоматичне реагування на аварійні ситуації є ще однією перевагою автоматизації. Системи можуть бути налаштовані на миттєве реагування у разі виникнення аварійних ситуацій, таких як витік небезпечних газів, перевищення допустимого рівня температури або повітряна тривога. Це забезпечує швидку реакцію та мінімізацію наслідків небезпечних подій, що сприяє зниженню ризику для життя і здоров'я працівників, а також зменшує можливі матеріальні збитки.

5.3 Стандарти безпеки при інсталяції та роботі автоматизованих систем

Автоматизація виробничих процесів вимагає суворої відповідності нормативно-правовим актам, що регулюють питання охорони праці. Встановлення та експлуатація автоматизованих систем повинні виконуватись у повній відповідності з цими вимогами, що включає забезпечення належного технічного стану обладнання. Регулярні перевірки та технічне обслуговування є невід'ємною частиною процесу, щоб гарантувати безперебійну та безпечну роботу обладнання. Це допомагає уникнути можливих поломок і аварійних ситуацій, що можуть загрожувати життю і здоров'ю працівників.

Також важливим аспектом є навчання та інструктаж персоналу. Працівники, які відповідають за експлуатацію автоматизованих систем, повинні проходити спеціальне навчання та регулярні інструктажі з охорони праці. Це допоможе їм набути необхідних знань та навичок для правильної експлуатації обладнання, а також підготовки до можливих аварійних ситуацій.

ВИСНОВКИ

Дана кваліфікаційна робота присвячена розробці системи автоматизації процесу зберігання та видачі електричних інструментів, яка забезпечує широкий функціонал для ефективного управління інструментами. Основні функціональні можливості системи включають реєстрацію нових інструментів з детальною інформацією про них, редагування та видалення наявних записів. Система також підтримує облік видачі та повернення інструментів працівниками, що дозволяє відстежувати історію використання кожного інструменту.

Окрім того, система містить модулі для управління обслуговуванням інструментів, де користувачі можуть реєструвати та переглядати записи про проведене технічне обслуговування та ремонти, з вказанням дати, вартості та опису виконаних робіт. Для покращення контролю за використанням інструментів, система забезпечує можливість формування звітів по виданих інструментах за певний період, а також звітів по загальній вартості обслуговування кожного інструменту.

Функціонал системи також включає можливість пошуку інструментів за ключовими словами, що значно спрощує процес знаходження необхідного обладнання. Для забезпечення безпеки доступу до системи передбачено механізм авторизації користувачів з різними рівнями доступу. Інтерфейс системи розроблений таким чином, щоб бути інтуїтивно зрозумілим та зручним для користувачів, що дозволяє швидко освоїти роботу з програмним забезпеченням та ефективно використовувати його у повсякденній діяльності.

У першому розділі була проведена детальна оцінка предметної області, включаючи аналіз основних аспектів зберігання інструментів, таких як безпека, організація простору, облік і контроль. Також розглянуто основні виклики, з якими стикаються підприємства при управлінні інструментами, та методи ведення обліку, серед яких виділяються штрихкодний облік, автоматизовані та хмарні системи. Аналіз сучасних систем автоматизованого обліку, таких як GigaTrak

Tool Tracking, Asset Panda та Cheqroom, дозволив виявити їхні переваги та недоліки, що стало основою для постановки задачі розробки власного рішення.

Другий розділ розкриває вибір технологій, проектування та особливості архітектури програмного забезпечення. Після аналізу різних мов програмування та систем управління базами даних були обрані C# та MS SQL Server як найоптимальніші технології для реалізації проекту. Вимоги до системи були визначені за допомогою діаграм варіантів використання, які описують функціональні та нефункціональні вимоги для системного адміністратора та користувача. Також було спроектовано базу даних, описано таблиці та побудовано фізичну модель. Архітектура проекту була розроблена на основі трьохрівневої моделі, яка включає детальний опис діаграм класів для кожного рівня.

Третій розділ присвячений розробці та верифікації програмного забезпечення. Було створено основні алгоритми для забезпечення роботи системи, включаючи авторизацію користувачів, додавання, редагування та видалення інструментів. Розробка програмних модулів включала створення користувацьких форм для введення, перегляду та управління даними, а також опис основних методів, що забезпечують функціональність цих форм. Експериментальна перевірка та тестування включала створення детальних тестових сценаріїв для всіх форм, а також проведення модульного тестування, яке підтвердило коректність реалізації функціональних можливостей системи. Інструкція користувача містить покрокові вказівки щодо використання системи, що допомагає користувачам швидко освоїти роботу з нею.

Проведена робота демонструє застосування сучасних підходів до автоматизації процесу зберігання та видачі електричних інструментів. Використання розробленого додатку дозволить значно спростити управління інструментами, забезпечити точний облік їх використання та обслуговування, підвищити ефективність роботи персоналу та оптимізувати витрати на технічне обслуговування інструментів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. ДСТУ 3008-15. Документація. Звіти у сфері науки та техніки. Структура та правила оформлення. Введ. 2015-06-22. К. Держстандарт України, 2017. 29 с.
2. Дипломне проектування для студентів усіх форм навчання спеціальностей 151 «Автоматизація та комп'ютерно-інтегровані технології»: навч. посібник / І.Ш. Невлюдов, А.О. Андрусевич, О.В. Токарева, Г.В. Пономарьова. Київ, 2018. 320 с.
3. Методичні вказівки з підготовки кваліфікаційної роботи бакалавра для студентів усіх форм навчання спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» освітньої програми «Автоматизація та комп'ютерно-інтегровані технології» / Упоряд.: І.Ш. Невлюдов, А.О. Андрусевич, О.В. Токарева, С.П. Новоселов, О.В. Сичова. Харків: ХНУРЕ, 2022. 45 с.
4. Поліщук О. М. Особливості застосування комп'ютерних технологій для автоматизації бухгалтерського обліку на підприємствах. Економічні науки. Серія : Облік і фінанси. – 2014. – 318 с.
5. Галусян А.В., Кононова О.Є. Організаційні аспекти та методика внутрішнього аудиту виробничих запасів підприємства. Східна Європа: економіка, бізнес та управління.. 2017. – 458 с.
6. Соколюк Юлія. Особливості обліку малоцінних та швидкозношуваних предметів. Східноєвропейський національний університет імені Лесі Українки. 2019.–31с.
7. GigaTrak: Tool Tracking System & Asset Management Software URL: <https://www.gigatrak.com/> (дата звернення 03.06.2024).
8. GigaTrak Tool Tracking System - Software Advice URL: <https://www.softwareadvice.ie/software/164311/tool-tracking-system> (дата звернення 03.06.2024).

9. Gigatrak Review: Pricing, Pros, Cons & Features URL: <https://comparecamp.com/gigatrak-review-pricing-pros-cons-features/> (дата звернення 03.06.2024).
10. Asset Panda: #1 Asset Management & Tracking Platform URL: <https://www.assetpanda.com/> (дата звернення 03.06.2024).
11. Asset Panda | Інтеграція та Автоматизація за 5 хвилин URL: <https://apix-drive.com/ua/asset-panda> (дата звернення 03.06.2024).
12. Pros and Cons of Asset Panda 2024 URL: <https://www.trustradius.com/products/asset-panda/reviews?qs=pros-and-cons> (дата звернення 03.06.2024).
13. Cheqroom: The #1 Equipment Management Software URL: <https://www.cheqroom.com/> (дата звернення 03.06.2024).
14. Cheqroom Alternative Using itemit's Equipment Tracking URL: <https://itemit.com/cheqroom-alternative-itemit-equipment-tracking-software/> (дата звернення 03.06.2024).
15. Cheqroom Reviews 2024. Verified Reviews, Pros & Cons URL: <https://www.capterra.com/p/140824/CHEQROOM/reviews/> (дата звернення 03.06.2024).
16. MATT WATSON. What is C# used for? / MATT WATSON // STACKIFY PRODUCT & COMPANY UPDATES. – 2020. – URL: <https://stackify.com/what-is-c-used-for/> (дата звернення 03.06.2024)..
17. NET: веб-сайт. URL:<https://training.epam.ua/ua/blog/301> (дата звернення 03.06.2024).
18. Java. URL:<https://dou.ua/lenta/articles/how-to-learn-java/> (дата звернення 03.06.2024).
19. C++ URL:http://www.znannya.org/?view=Cpp_basics (дата звернення 03.06.2024).
20. Visual Studio 2019 : веб-сайт. URL: <https://codeguida.com/post/1754> (дата звернення 03.06.2024).

21. Гулковський М. М., Бурак Н. Є. Сучасні системи управління базами даних СУБД. 2020. 43с.
22. Microsoft SQL Server URL: https://www.metabase.com/data_sources/microsoft-sql-server (дата звернення 03.06.2024).
23. PostgreSQL URL: <https://www.postgresql.org/> (дата звернення 03.06.2024).
24. Dubois Paul. MySQL. Addison-Wesley, 2013. – 96 р.
25. SQL Server Management Studio (SSMS) URL: <https://learn.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16> (дата звернення 03.06.2024).
26. Методичні вказівки до виконання розділу «Охорона праці» у випускних роботах ОКР «бакалавр» усіх форм навчання [Текст] / упоряд.: Б. В. Дзюнзюк, В. А. Айвазов, Т. Є. Стиценко. – Харків: ХНУРЕ, 2012. – 28 с.
27. <https://idcard.com.ua/ua/fudan1k-tag-pvc-label-round/> URL (дата звернення 26.06.2024).