

Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”

Завідувач кафедри кібербезпеки
та програмного забезпечення

д.т.н., професор

Олексій СМІРНОВ

“ ” 2021 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти

на тему

**“ Дослідження та програмна реалізація системи аудиту мережі
теплопостачання з використанням технології ІоТ”**

Виконав здобувач вищої освіти

II курсу, групи _____

ОПП «Комп’ютерна інженерія»

спеціальності 123 «Комп’ютерна інженерія»

Кіреєв С.М.

« » 2021 р.

Керівник проекту

доктор технічних наук, доцент

Єлизавета МЕЛЕШКО

« » 2021 р.

Рецензент _____

м. Кропивницький

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Освітній ступінь бакалавр
Галузь знань 12 "Інформаційні технології"
Спеціальність 123 "Комп'ютерна інженерія"
Освітньо-професійна (освітньо-наукова) програма "Комп'ютерна інженерія"

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.
Олексій СМІРНОВ
«__» _____ 2021 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ДРУГИМ (МАГІСТЕРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Кіреєву Сергію Михайловичу

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та програмна реалізація системи аудиту мережі теплопостачання з використанням технології IoT

2. Керівник роботи Мелешко Єлизавета Владиславівна, доктор техн. наук, доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу №42-13 від 02.08.2021 року

3. Строк подання роботи до захисту 10.12.2021 р.

4. Мета та завдання випускної кваліфікаційної роботи: Метою розробки є програмне забезпечення системи аудиту мережі теплопостачання з використанням технології IoT

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

- | | |
|--|---|
| <u>1. Призначення та область використання.</u> | <u>7. Економічна ефективність</u> |
| <u>2. Перегляд аналогічних існуючих систем.</u> | <u>розробленої програми.</u> |
| <u>3. Опис і обґрунтування проектних рішень.</u> | <u>8. Заходи з охорони праці та техніки</u> |
| <u>4. Етапи програмування системи.</u> | <u>безпеки.</u> |
| <u>5. Впровадження системи в промислову</u> | <u>9. Висновки.</u> |
| <u>експлуатацію.</u> | |

6. Наукова новизна

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

Наукова новизна 1 аркуш

Структурна схема системи 1 аркуш

Функціональна схема системи 1 аркуш

Блок-схема алгоритму роботи додатку 2 аркуша

Діаграма процесів 1 аркуш

Показники економічної ефективності 1 аркуш

6. Консультанти по роботі, із зазначенням розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Савеленко Г.В., к.т.н., доцент	25.10.2021	12.11.2021
Охорона праці	Оришака О.В., к.т.н., доцент	04.11.2021	20.11.2021

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.10.2021 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2021 р.	
3.	Розробка моделі компонента	20.10.2021 р.	
4.	Розробка структур даних	25.10.2021 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.10.2021 р.	
6.	Програмування алгоритмів	10.11.2021 р.	
7.	Розрахунок економічної ефективності	13.11.2021 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2021 р.	
9.	Оформлення ПЗ	17.11.2021 р.	
10.	Попередній захист роботи	28.11.2021 р.	

Дата видачі завдання

«__» _____ 20 р.

Підпис керівника

(прізвище та ініціали)

Завдання прийнято до виконання

«__» _____ 20 р.

Підпис здобувача

(прізвище та ініціали)

АНОТАЦІЯ

Кіреєв С.М. Дослідження та програмна реалізація системи аудиту мережі тепlopостачання з використанням технології IoT. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2021.

В даній випускній кваліфікаційній роботі розроблено програмне забезпечення, яке призначено для системи аудиту мережі тепlopостачання з використанням технології IoT.

Метою розробки є дослідження та програмна реалізація системи аудиту мережі тепlopостачання з використанням технології IoT.

Об'єктом дослідження є процес аудиту мережі тепlopостачання.

Предметом дослідження є методи аудиту мережі тепlopостачання з використанням технології IoT.

Методи дослідження базуються на методах об'єктно-орієнтованого програмування, теорії інформації та кодування, методах роботи з комп'ютерними системами та мережами, технології Інтернету речей.

Результат роботи – програмна реалізація системи аудиту мережі тепlopостачання з використанням технології IoT.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ архітектури IBM PC з ОС Windows 7/8/10.

Програму розроблено на мові програмування C# в середовищах розробки Microsoft Visual Studio та Unity.

Ключові слова: комп'ютерна інженерія, мережа тепlopостачання, технологія IoT

ABSTRACT

Kirieleiev S.M. Research and program realization of the heat supply network audit system using IoT technology. 123 Computer Engineering. Central Ukrainian National Technical University. Kropivnitsky 2021.

In this qualification final work the software which is intended for system of audit of a network of heat supply with use of IoT technology is developed.

The purpose of the development is the research and program implementation of the system of the heat supply network audit system using IoT technology.

The object of the study is the process of auditing the heating network.

The subject of the research is the methods of audit of the heat supply network using IoT technology.

Methods of research are based on the methods of object-oriented programming, information and coding theory, methods of working with computer systems and networks, Internet of Things technology.

The result of the work is the software implementation of the system of the heat supply network audit system using IoT technology.

In the process of working on a software model an analysis of existing hardware and software was performed. All components of the software developed are fully described.

User-friendly user interface is developed. These are instructions for working with software.

The program can be used on an IBM PC running Windows 7/8/10.

The program was developed in the C# programming language in Microsoft Visual Studio and Unity development environments.

Keywords: computer engineering, heat supply network, IoT technology

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	7
1.1 Призначення системи.....	7
1.2 Область застосування.....	8
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	9
2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми магістерської роботи	9
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування	19
2.3 Розгорнута постановка завдання	23
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	25
3.1 Опис функціонування системи.....	25
3.2 Розробка структурної схеми	29
3.3 Розробка функціональної схеми.....	31
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ.....	34
4.1 Блок-схеми та опис алгоритмів функціонування системи.....	34
4.2 Захист розробленого програмного забезпечення	46
5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	48
6 НАУКОВА НОВИЗНА	51
7 ДАНІ ПРО ЕКОНОМІЧНУ ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ....	52
7.1 Техніко-економічне обґрунтування теми магістерської роботи.....	52

						ВКРМ-123.21.0008.00.00.ПЗ		
Вим.	Арк.	№ докум.	Підп.	Дата				
<i>Розроб.</i>	<i>Кірсев С.М.</i>				<i>Дослідження та програмна реалізація системи аудиту мережі теплопостачання з використанням технології IoT</i>	Лім.	Аркуш	Аркушів
<i>Перев.</i>	<i>Мелешко Є.В.</i>					М	1	87
<i>Н.контр.</i>	<i>Гермак В.С.</i>					ЦНТУ КІ-20М		
<i>Затв.</i>	<i>Смірнов О.А.</i>							

7.2 Розрахунок трудомісткості розробки програмної продукції	54
7.3 Визначення чисельності виконавців і планового фонду зарплати	56
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника	60
7.5 Визначення собівартості розробки та ціни програмної продукції	65
7.6 Визначення об'єму капітальних вкладень у споживача програмної продукції	68
7.7 Визначення експлуатаційних витрат	68
7.8 Визначення економічної ефективності програмної продукції	70
7.9 Висновки	72
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	73
8.1 Аналіз умов праці на робочому місці програміста	73
8.2 Розробка заходів з умов поліпшення охорони праці	77
8.3 Висновки до розділу	81
9 ОСНОВНІ ВИСНОВКИ	82
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	84

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

- HTML – HyperText Markup Language;
IOT – Internet of Things;
АЗ – апаратний засіб;
ОС – операційна система;
ПЗ – постановка задач;
ТЗ – технічна задача.

Кафедра КБПЗ – 2021 рік

					ВКРМ-123.21.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

Актуальність теми. В наш час системи аудиту для різних галузей людської діяльності активно розвиваються, тому що для якісного управління будь чим необхідно проводити спостереження за об'єктом управління. Навіть при автоматичному управлінні системою, потрібні перевірки роботи та її результатів. Для цього існують програми аудиту. За допомогою них аудитор може перевірити ефективність роботи підприємства чи якої небудь системи. Аудит дозволяє своєчасно виявляти помилки в роботі системи та виправити їх.

Закономірно, що аудит також потрібно використовувати в системах, які регулюють подачу теплової енергії у будівлі. Такі програми здатні дати аудитору потрібну кількість інформації для оцінки ефективності роботи тої чи іншої системи тепlopостачання.

Системи аудиту тепlopостачання все частіше використовують IoT-технології, адже саме вони дозволяють поєднати фізичні пристрої за допомогою мережі та отримувати дані з різних об'єктів, на яких встановлені системи регулювання подачі теплової енергії.

Метою даної випускної кваліфікаційної роботи є створення програмного забезпечення системи аудиту мережі тепlopостачання з використанням технології IoT для того, щоб аудитор або власник цієї системи міг бачити якість роботи його системи на кожному з об'єктів де встановлено обладнання системи тепlopостачання.

Програмне забезпечення такого типу є актуальним, адже воно також може порівняти роботу системи мережі тепlopостачання як без нового обладнання, так із ним. Тим самим програмне забезпечення може показати різницю двох систем та зрозуміти доцільність встановлення нового обладнання.

					ВКРМ-123.21.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи аудиту мережі тепlopостачання з використанням технології IoT.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

- Огляд існуючих систем аудиту мережі тепlopостачання.
- Розробка системи аудиту мережі тепlopостачання з використанням технології IoT.
- Випробовування розробленої системи аудиту мережі тепlopостачання з використанням технології IoT.

Об'єктом дослідження є процес аудиту мережі тепlopостачання.

Предметом дослідження є методи аудиту мережі тепlopостачання з використанням технології IoT.

Методи дослідження базуються на методах об'єктно-орієнтованого програмування, теорії інформації та кодування, методах роботи з комп'ютерними системами та мережами, технології Інтернету речей.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

- Удосконалено метод аудиту мережі тепlopостачання з використанням технології IoT.
- Розроблено вітчизняний продукт аудиту мережі тепlopостачання з використанням технології IoT, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі аудиту мережі тепlopостачання з використанням технології IoT.

Достовірність наукових результатів підтверджена теоретичними викладеннями, даними комп'ютерного моделювання, коректними дослідженнями

					ВКРМ-123.21.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

параметрів на функціонуючій обчислювальній мережі, а також відповідністю отриманих результатів окремим результатам, наведеним у науковій літературі.

Тобто, виходячи з вище наведеного, дослідження та програмна реалізація системи аудиту мережі тепlopостачання з використанням технології IoT, є актуальною науково-практичною задачею, яка потребує вирішення у цій випускній кваліфікаційній роботі.

Кафедра _ КБПЗ _ 2021 рік

					VKPM-123.21.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Розроблена система призначена для використання в організаціях та на підприємствах, які займаються установкою систем опалення. Вона використовує технологію IoT.

IoT – концепція мережі фізичних пристроїв, які мають вбудовані датчики, а також ПЗ, що дозволяє здійснювати передачу і обмін даними між зовнішнім середовищем і комп'ютерними системами за допомогою використання стандартних протоколів зв'язку.

Основною концепцією IoT є можливість підключення всіляких пристроїв, які людина може використовувати в повсякденному житті, наприклад, таких як холодильник, кондиціонер, пральна машинка, автомобіль, велосипед і навіть кросівки. Всі ці об'єкти (речі) повинні бути оснащені вбудованими сенсорами, обмінюватися інформацією між собою і виконувати різні дії в залежності від отриманої інформації. Прикладом впровадження такої технології є системи типу «Розумний будинок». Така система аналізує дані навколишнього середовища і в залежності від показників регулює температуру в приміщенні. Наприклад, зимою регулюється інтенсивність опалення, а літом використовуються механізми відкривання і закривання вікон, завдяки чому провітрюються кімнати, і все це відбувається автоматично без втручання людини.

Програми аудиту різних систем з використанням системи IoT переважно потрібні на підприємствах та в організаціях, де одній людині треба обслуговувати десятки, або навіть сотні пристроїв, що розташовані в різних кабінетах і на різних поверхах будівлі.

					ВКРМ-123.21.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

1.2 Область застосування

Розроблене в ході роботи програмне забезпечення можна використовувати при налаштуванні програмного середовища будь яких ПК під управлінням ОС Windows. Така система може бути корисною в багатьох сферах використання сучасних комп'ютерних систем.

Особливо доцільним є використання запропонованого програмного забезпечення та положень описаних в пояснювальній записці для підприємств що займаються встановленням систем теплопостачання. Воно надасть можливість отримувати інформацію про температуру води, що протікає по будівлі. Крім того надасть можливість слідкувати за роботою системи теплопостачання та вчасно виявляти несправності.

Вказані положення підтверджують актуальність поставленої мети досліджень та задачі випускної кваліфікаційної роботи і їх виконання має підтвердити рівень кваліфікації автора.

Розроблена програма дозволяє швидко та ефективно провести аудит системи теплопостачання та порівняти їх з попередніми результатами.

Використані в ході роботи над проектом алгоритми та методи програмування, а також програмний код з коментарями і пояснювальна записка будуть корисні починаючим програмістам та проектувальникам комп'ютерних систем на базі технології IoT.

Також потрібно зазначити, що програмне забезпечення, розроблене в рамках даної роботи не має великих вимог до робочих станцій – мінімальні системні вимоги такі ж як і для встановлення ОС Windows.

Програма може працювати на усіх версіях операційної системи Windows починаючи з Windows 7 і до останніх версій.

					ВКРМ-123.21.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

Перед тим, як перейти безпосередньо до роботи над розробкою будь-якої програмної системи необхідно провести дослідження вже існуючих аналогів і літературних джерел. Аналіз їх переваг та недоліків дозволить дізнатися які уже наробки у даному напрямку існують, використати деякі готові підходи та рішення, уникнути багатьох помилок, обрати оптимальні методи розробки, вдосконалити та розробити свої методи на базі існуючих.

До інформаційних технологій аудиту комп'ютеризованих систем можна віднести ПЗ, яке може бути використане, як для перевірки так і для автоматизації систем управління. Комп'ютерно-інформаційні технології є основним інструментом на етапі переходу від традиційних технологій до систем комплексної автоматизації.

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми магістерської роботи

Програмне забезпечення систем аудиту можна поділити на категорії:

1) Нормативно-довідкові та інформаційно-пошукові системи. Призначені для забезпечення правової підтримки рішень, що приймаються і задоволення основних інформаційних потреб у період проведення перевірки. Серед нормативно-довідкових та інформаційно-пошукових систем можна виділити наступні групи:

- системи нормативно-правової підтримки, які забезпечують нормативно-правову підтримку аудитора своєчасною та актуальною інформацією. Приклади таких систем – «Ліга: Закон», «Консультант Плюс», «Грант», «Юрисконсульт», тощо;

- довідкові системи, що забезпечують необхідною додатковою

					ВКРМ-123.21.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

інформацією для підготовки та планування аудиту. Сюди, можна віднести:

- інформаційно-довідкові системи для забезпечення інформацією про суб'єкта господарювання на етапі прийняття рішення про проведення аудиту.

- інформаційно-пошукові системи, розміщені в мережі Інтернет, які надають загальну інформацію аудитору та забезпечують широкі можливості для пошуку інформації у відкритих джерелах;

- бази даних про підприємства та організації, які можуть бути корисними в процесі аудиту для дослідження та порівняння показників об'єкту аудиту з подібними об'єктами галузі.

2) Інформаційні технології загального призначення, до яких належать текстові й табличні редактори, засоби електронного документообігу та інші технології, основним призначенням яких є надання аудиторам різноманітних засобів підготовки документів, автоматизації розрахунків, введення баз даних клієнтів.

3) Спеціалізовані інформаційно-аналітичні та статистичні системи, які забезпечують аудитора засобами для проведення комплексного фінансового аналізу та організації статистичних досліджень.

4) Інформаційні системи обліку. Підготовлені та налаштовані аудитором інформаційні системи обліку, дають змогу здійснювати процедуру тестування алгоритмів інформаційних систем обліку. Досягається це за допомогою автоматизованого чи ручного перенесення даних з інформаційної системи клієнта до інформаційної системи, налаштованої користувачем, з їх подальшою обробкою. Порівнявши результати обробки із результатами, сформованими системою клієнта, аудитор має можливість зробити висновки стосовно вірності функціонування алгоритмів інформаційної системи клієнта.

Енергоаудит – це перевірка та аналіз енергоспоживання будівлі чи об'єкта для скорочення енерговитрат підприємства.

Головне завдання енергоаудиту – виявити нераціональне використання енергоресурсів та запропонувати клієнту заходи щодо енергозбереження, які

					ВКРМ-123.21.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

дозволять скоротити витрати на енергоспоживання.

Енергетичний аудит потрібен для того, щоб скоротити витрати на опалення, електроенергію, газ, воду. Знайти втрати та виток тепла, води, електроенергії та усунути їх. Зробити енергопаспорт та програму енергозбереження. А також перед ремонтом, наприклад, щоб зрозуміти, які вікна необхідно міняти, який утеплювач ставити, яка система освітлення підійде краще і т.д.

Енергоаудит проводиться так:

- збір вихідних даних;
- виїзд на об'єкт, що обстежується;
- аналіз існуючих договірних умов з енергопостачальними організаціями;
- аналіз витрат на електрику, тепло, воду, газ;
- інструментальне обстеження будівель, споруд, основного технологічного обладнання, котельні;
- аналіз отриманих даних, розрахунки, визначення місць втрат та витоків;
- розробка звіту з енергетичного аудиту, енергопаспорт, технічний висновок;
- погодження заходів щодо енергозбереження із замовником;
- впровадження заходів щодо енергозбереження, скорочення витрат.

На першому етапі енергоаудиту підприємства здійснюється:

- збір даних щодо обсягів споживання тепла, води, газу, електричної енергії за останні рік-два;
- аналіз проектної документації, архітектурно-планувальних даних щодо будови та дані Технічного паспорту БТІ;
- збір даних щодо кількості людей, персоналу та відвідувачів будівлі чи організації;
- збір даних про навантаження та технічні характеристики основного обладнання;
- збір фінансових даних щодо оплати, рахунків, комуналки та інших

					ВКРМ-123.21.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

послуг.

Після аналізу вихідних даних та документів відбувається інструментальний енергоаудит, який включає наступні етапи (вони більш детально розглянуті нижче):

- Обстеження лічильників та системи обліку тепла, електроенергії, води.
- Обстеження тепlopостачання.
- Обстеження будівель та огорожувальних конструкцій.
- Обстеження системи електропостачання.
- Обстеження водопостачання та водовідведення.
- Обстеження котельні.
- Тривалість енергетичного обстеження залежить від складності об'єкта, кількості вимірів, глибини та деталізації інформації, що збирається.

Розглянемо приклади програмного забезпечення для енергоаудиту.

Програмні продукти ENSI для енергоаудиту

Понад 20 років норвезька консалтингова компанія ENSI (Energy Saving International) надає технічне сприяння та послуги з розвитку кадрового потенціалу у сфері “Енергетичний Аудит Будинків” для багатьох країн. Багато енергоаудиторів використовують у своїй роботі методичні матеріали та програмні продукти ENSI.

Для того, щоб допомогти енергоаудиторам у проведенні швидких розрахунків енергетичних характеристик будівель, компанія ENSI розробила зручний для користувача програмний продукт ENSI EAB (Енергоаудит Будинків). Програмний продукт підходить для роботи з будівлями, що реконструюються, і для оцінки заходів щодо підвищення енергоефективності в існуючих будівлях.

Програма ENSI EAB включає за промовчанням нормативні вступні дані для всіх параметрів (стандартні величини). Тому, енергетичні розрахунки можна проводити дуже швидко, лише змінюючи параметри, які від нормативних величин.

					ВКРМ-123.21.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

Програмне забезпечення дозволяє швидко змінювати введені параметри та/або типові будівлі, включені до довідкових даних, легко моделювати та швидко отримувати результати.

У програмне забезпечення включено сім типових будівель:

- Житловий будинок.
- Лікарня.
- Дитячий садок.
- Офіс.
- Школа.
- Університет.
- Склад.

Підтипи та інші типи будівлі можуть бути легко додані.

Ряд параметрів, що вводяться у програму:

- Теплові характеристики будівлі;
- Конструкція будівлі, положення та орієнтація;
- Інфільтрація;
- Мікроклімат у приміщеннях;
- внутрішні теплові навантаження;
- Висвітлення;
- Системи опалення, вентиляції та кондиціонування;
- Автоматичне регулювання;
- Кліматичні умови, включаючи вплив сонячної радіації.

У розрахунках використовують усі основні параметри, необхідні енергетичних розрахунків, такі як коефіцієнти теплопередачі, інфільтрація, температури, графіки роботи, продуктивність вентиляційних систем, періоди роботи устаткування, освітлення, ККД систем енергозабезпечення тощо.

Для кожного пункту енергетичного бюджету на екрані програмного забезпечення можна бачити параметри, що впливають використання енергії, і стандартні показники, засновані на сучасних російських будівельних стандартах.

					ВКРМ-123.21.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

Зовнішній вигляд програми ENSIEAB наведено на рис. 2.1.

Параметры	Стандартн.	По факту	Базовая линия	Чувствительн. kWh/m ² a	Меры	Эконом.
1. Отопление		56,0 kWh/m²a				
U – стен	0,30 W/m ² K	0,45	0,45	+ 0,1 W/m ² K = 5,48	0,30	-7,72
U – окон	2,40 W/m ² K	3,00	3,00	+ 0,1 W/m ² K = 3,03	1,30	-46,60
U – крыши	0,20 W/m ² K	0,20	0,20	+ 0,1 W/m ² K = 1,39	0,20	
U – пола	0,30 W/m ² K	0,30	0,30	+ 0,1 W/m ² K = 1,39	0,30	
Козф. компактности	0,37 -	0,37	0,37		0,37	
Козф. остекления	27,1 %	27,1	27,1		27,1	
Солнечн. поступления	0,55 -	0,55	0,55		0,55	
Инфильтрация	0,25 1/h	0,40	0,40	+ 0,1 1/h = 10,28	0,25	-14,39
Внутр. температура	21,0 °C	19,0	21,0	+ 1 °C = 7,57	21,0	
Сниженная температура	18,0 °C	18,0	18,0	+ 1 °C = 5,79	18,0	
Вклад от						
Вентиляция (отопление)	kWh/m ² a	7,43	0,00		0,00	
Освещение	kWh/m ² a	17,60	17,96		10,19	
Прочее	kWh/m ² a	10,59	10,81		9,81	
Потребность в энергии	kWh/m²a	70,7	82,5		28,1	
КПД теплоотдачи	93,0 %	93,0	93,0		93,0	
КПД распредел. системы	97,0 %	97,0	97,0		97,0	
Автом. управление	98,0 %	98,0	98,0		98,0	
Э и О / ЭМ	98,0 %	95,0	95,0		98,0	-3,01
Сумма	kWh/m²a	84,2	98,3		32,5	
КПД генерации	100,0 %	100,0	100,0		100,0	
Потребление энергии	kWh/m²a	84,2	98,3		32,5	

Рисунок 2.1 – Програма для енергоаудиту ENSIEAB

Результаты расчетов программы ENSIEAB могут использоваться для документирования соответствия строительным нормам как выходные данные для энергетической сертификации та для энергоаудиту існуючих будівель. Программу можна також використовувати для документування відповідності будівель будівельним нормам.

Результаты расчетов подаются как данные про потребление энергии. Програма розраховує бюджети споживання енергії та навантажень.

Економия энергии рассчитывается для каждого энергосберегающего заходу (кВтч/м², кВтг/рік). За потреби виконуються розрахунки Базового енергоспоживання.

Наприклад, якщо під час обстеження з'ясувалося, що температура будівлі нижче нормативних вимог, необхідно розрахувати, скільки енергії споживало б цей будинок за умови забезпечення нормативних температур. Це важливо з метою оцінки реальної ефективності енергозберігаючих заходів.

Програма виконує розрахунок ЕТ-криву (залежність споживання енергії від температури зовнішнього повітря), яка потрібна на ведення системи Енергомоніторингу.

САПР РАСХОД-РУ

ПЗ, що застосовується енергоаудиту, зокрема, для розрахунку та проектування витратомірів змінного перепаду тиску зі стандартними звужувальними пристроями відповідно до комплексу міждержавних стандартів.

САПР "РАСХОД-РУ" дозволяє автоматизувати процес розрахунку та проектування витратомірів змінного перепаду тиску, а разом виконати всі необхідні перевірки умов застосування (обмежень) цього методу.

САПР "РАСХОД-РУ" призначена для проектування витратомірів 48 різних середовищ, у тому числі, таких, як природний газ, вологий нафтовий газ, перегріта пара, суха насичена пара, вода, повітря, азот, аргон, кисень та ін. Для природного газу в САПР "РАСХОД-РУ" реалізовані всі діючі методики розрахунку коефіцієнта стисливості газу (КХ19 мод.; УС GERG-91 мод.; УС AGA8-92DC; УС ВНИЦ СМВ; ЦД 7-2005).

САПР "РАСХОД-РУ" у загальному випадку забезпечує можливість реалізації чотирьох різних варіантів розрахунку параметрів витратоміра:

- За заданим значенням верхньої межі вимірювань перепаду тиску;
- За заданим значенням допускається втрати тиску на пристрої, що звужує;
- Забезпечують мінімальну невизначеність результату вимірювання витрати середовища;
- За заданими характеристиками звужувального пристрою та вимірювального трубопроводу (так званий зворотний розрахунок витратоміра).

					ВКРМ-123.21.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

Перші три варіанти розрахунку застосовують при початковому проектуванні витратоміра змінного перепаду тиску: перший - для заданого вимірювального перетворювача перепаду тиску; другий і третій - для вирішення оптимізаційної задачі за критерієм, відповідно, втрат тиску на пристрої, що звужує, і точності вимірювання. Четвертий варіант розрахунку застосовують при уточненні параметрів заданого витратоміру та умов його застосування під вимоги ГОСТ 8.586.1,2,3,4,5—2005 (тут же виконується перевірка виконання вимог цих стандартів щодо прямолінійних ділянок вимірювального трубопроводу і витратоміру в цілому).

Крім того, при проектуванні витратоміра САПР "РАСХОД-РУ" передбачається можливість вибору одного з трьох варіантів конфігурації витратоміра: з роздільним виміром параметрів потоку; з обчислювачем витрати та кількості середовища; із вимірювальним комплексом.

При цьому в кожному вимірювальному каналі будь-якого параметра потоку (перепаду тиску на пристрої, що звужує, температури середовища, тиску середовища) може бути встановлено (поєднано) до чотирьох вимірювальних перетворювачів. У свою чергу, для розширення діапазону вимірювання витрати середовища у вимірювальних каналах перепаду тиску на звужувальному пристрої і тиску середовища передбачена можливість застосування додаткових (паралельно працюючих) вимірювальних перетворювачів.

Після вибору типу розв'язуваної задачі (варіанту розрахунку параметрів витратоміра) вводять вихідні дані для розрахунку. Інтерфейс для введення параметрів та характеристик витратоміра складається з шести розділів: параметри та характеристики середовища; параметри та характеристики звужуючого пристрою; параметри та характеристики трубопроводу; параметри та характеристики прямолінійних ділянок та оснащення вимірювального трубопроводу; параметри та характеристики засобів вимірювань та обчислень; параметри та характеристики витратомірного вузла.

Зовнішній вигляд програми ENSIEAB наведено на рис. 2.2.

					ВКРМ-123.21.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

Для роботи із Сервісом енергоаудиторам та СРО надається програмне забезпечення АРМ "Е-Pass". Використання АРМ "Е-Pass" дозволяє підвищити ефективність роботи енергоаудиторів щодо оформлення результатів енергетичних обстежень. Програмне забезпечення розроблено відповідно до нормативно-правових актів, що регламентують процес проведення енергетичних обстежень.

Зовнішній вигляд даного програмного забезпечення наведено на рис. 2.3.

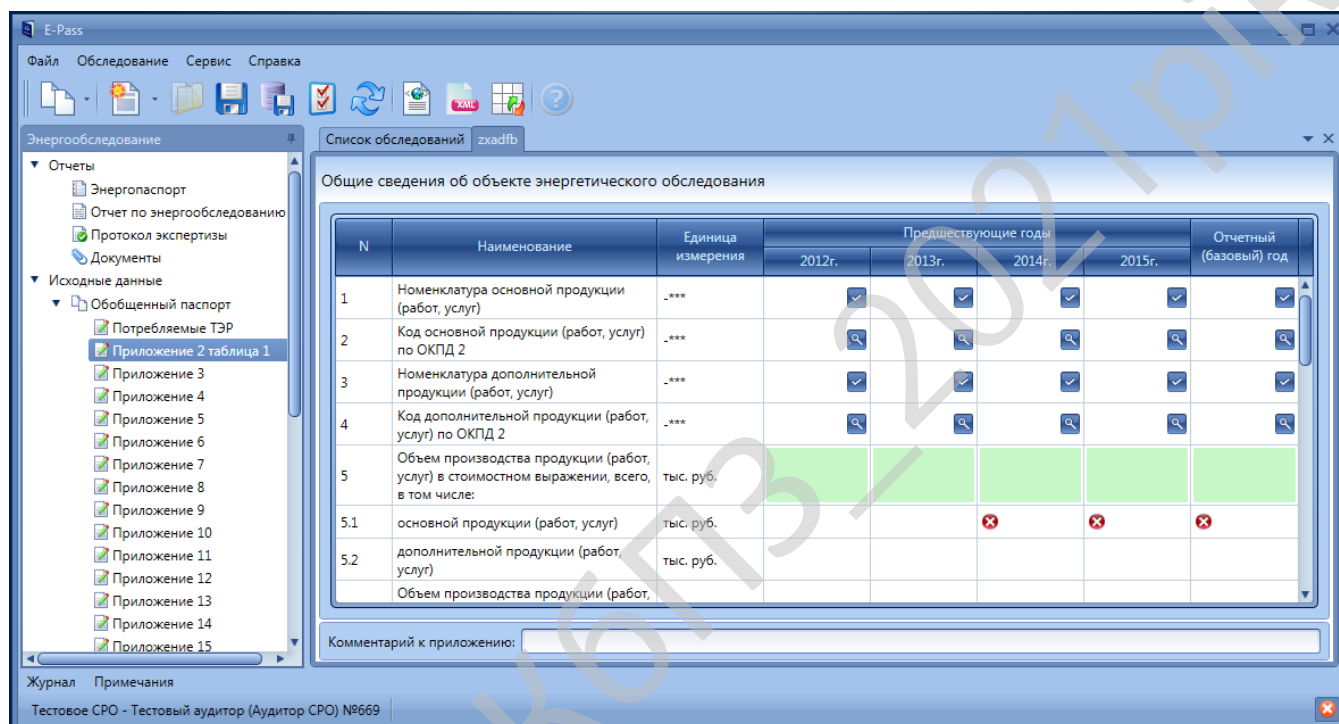


Рисунок 2.3 – Програма для енергоаудиту АРМ "Е-Pass"

Можливості програмного забезпечення:

- Зручне введення даних результатів проведення енергетичного обстеження.
- Автоматизоване ведення реєстру енергоаудиторів та енергообстежень (зі зручним пошуком та угрупованням).
- Підтримка здійснення процедури експертизи результатів енергообстеження.
- Відстеження процесу обміну та узгодження результатів обстеження між енергоаудитором та СРО.
- Зберігання результатів енергообстеження на центральному сервері, з можливістю роботи з локальною копією даних (мобільний режим).

- Можливість роботи в мобільному режимі безпосередньо на об'єкті, без постійного підключення до сервера.
- Автоматичне заповнення розрахункових полів.
- Наочний розрахунок теплових характеристик будівлі.
- Вбудований калькулятор вартості енергетичного обстеження.
- Інтерактивна нормативно-довідкова система.
- Автоматичне оновлення програмного забезпечення та нормативно-довідкової інформації.
- Автоматизоване формування документації для друку: звіт з енергетичного обстеження; енергетичний паспорт організації.

Використання АРМ "Е-Pass" дозволяє зробити процес оформлення результатів енергетичного обстеження та формування енергетичного паспорта більш простим та зручним, що дозволяє знизити витрати та підвищити ефективність компаній, що працюють у сфері проведення енергетичних обстежень.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Для розробки програмного забезпечення системи аудиту мережі теплопостачання з використанням технології IoT була вибрана мова програмування C# і середовище розробки Visual Studio.

C# – високорівнева, об'єктно-орієнтована, універсальна мова програмування. Синтаксис C# близький до C++ і Java.

Мова має строгу статичну типізацію, підтримує поліморфізм, переваження операторів, вказівники на функції-члени класів, атрибути, події, властивості, винятки, коментарі у форматі XML.

Присутність або відсутність тих або інших виразних особливостей мови диктується тим, чи може конкретна мовна особливість бути трансльована у відповідні конструкції CLR. CLR надає C#, як і всім іншим .NET -орієнтованим

					ВКРМ-123.21.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

2) Проект SharpDevelop від компанії icsharpcode, який використовується як альтернатива Visual Studio. Забезпечує повну реалізацію Common Language Infrastructure. Остання стабільна версія IDE 4.4 (28 серпня 2013), тестова версія 5.0 (13 лютого 2014). Зовнішній вигляд IDE дуже нагадує Microsoft Visual C#, що робить комфортним перехід від одного середовища до іншого.

3) Проект Mono, початий компанією Xamarin і продовжений її покупцем і наступником Novell, забезпечує відкритий компілятор C#, повну відкриту реалізацію Common Language Infrastructure, включаючи потрібні бібліотеки фреймворка відповідно до специфікації ECMA, і близьку до повної реалізацію власницьких бібліотек класів Microsoft .NET до .NET 2.0, але не специфічних бібліотек .NET 3.0 і .NET 3.5, як для Mono 2.0.

4) Проект DotGNU також надає відкритий компілятор C#, близьку до повної реалізацію Common Language Infrastructure, включаючи потрібні бібліотеки фреймворка відповідно до специфікації ECMA, і підмножину деяких залишених власницьких бібліотек класів Microsoft .NET до .NET 2.0 (які не документовані або не включені у специфікації ECMA, але включені у стандартне визначення Microsoft .NET Framework).

5) DotNetAnywhere Micro Framework Common Language Runtime націлений на вбудовані системи, і підтримує майже всі специфікації C# 2.0.

6) Мова C# була обрана тому що вона є об'єктно-орієнтованою і дозволяє реалізовувати задачі які я перед собою ставив

7) Інтерфейс програми був розроблений за допомогою Visual Studio.

Також для кращої візуалізації даних у цій роботі було використано рушій Unity.

Unity це – багатоплатформовий інструмент для розробки двох- та тривимірних додатків та ігор, що працює на операційних системах Windows і OS X. Створені за допомогою Unity застосунки працюють під системами Windows, OS X, Android, Apple iOS, Linux, а також на гральних консолях Wii, PlayStation 3 і Xbox 360.

					ВКРМ-123.21.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

Є можливість створювати Інтернет-додатки за допомогою спеціального під'єднуваного модуля для браузера Unity, а також за допомогою експериментальної реалізації в межах модуля Adobe Flash Player. Застосунки, створені за допомогою Unity, підтримують DirectX та OpenGL.

Редактор Unity має простий Drag & Drop інтерфейс, який легко налаштовувати, що складається з різних вікон, завдяки чому можна проводити налагодження гри прямо в редакторі. Русій підтримує три сценарних мови: C#, JavaScript (модифікація).

Проект в Unity ділиться на сцени (рівні) – окремі файли, що містять свої ігрові світи зі своїм набором об'єктів, сценаріїв, і налаштувань. Сцени можуть містити в собі як, об'єкти (моделі), так і порожні ігрові об'єкти — тобто ті, які не мають моделі. Об'єкти, в свою чергу містять набори компонентів, з якими і взаємодіють скрипти. Також у них є назва (в Unity допускається наявність двох і більше об'єктів з однаковими назвами), може бути тег (мітка) і шар, на якому він повинен відображатися. Так, у будь-якого предмета на сцені обов'язково присутній компонент Transform— він зберігає в собі координати місця розташування, повороту і розмірів по всіх трьох осях. У об'єктів з видимою геометрією також за умовчанням присутній компонент Mesh Renderer, що робить модель видимою.

Також Unity підтримує фізику твердих тіл і тканини, фізику типу Ragdoll (ганчіркова лялька). У редакторі є система успадкування об'єктів; дочірні об'єкти будуть повторювати всі зміни позиції, повороту і масштабу батьківського об'єкта. Скрипти в редакторі прикріплюються до об'єктів у вигляді окремих компонентів.

При імпорті текстури в русій можна згенерувати alpha-канал, mip-рівні, normal-map, light-map, карту відображень, проте безпосередньо на модель текстуру прикріпити не можна — буде створено матеріал, з яким буде призначений шейдер, і потім матеріал прикріпиться до моделі. Редактор Unity підтримує написання і редагування шейдерів. Крім того він містить компонент

					ВКРМ-123.21.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

для створення анімації, яку також можна створити попередньо в 3D-редакторі та імпортувати разом з моделлю, а потім розбити на файли.

Графічний рушій використовує DirectX (Windows), OpenGL (Mac, Windows, Linux), OpenGL ES (Android, iOS), та спеціальне власне API для Wii. Також підтримуються bump mapping, reflection mapping, parallax mapping, screen space ambient occlusion (SSAO), динамічні тіні з використанням shadow maps, render-to-texture та повноекранні ефекти post-processing. Unity може підтримувати файли 3ds Max, Maya, Softimage, Blender, modo, ZBrush, Cinema4D, Cheetah3D, Adobe Photoshop, Adobe Fireworks та Allegorithmic Substance. В ігровий проект Unity можна імпортувати об'єкти цих програм та робити налаштування за допомогою графічного інтерфейсу.

2.3 Розгорнута постановка завдання

Незважаючи на те, що всі програми, представлені в огляді, виконують одну і ту саму функцію, навряд чи можна сказати однозначно, яка краще, а яка – гірше. Дані програмні продукти досить потужні, але всі в них мають високі вимоги до апаратної частини системи, і на жаль мають закритий програмний код та досить високу ціну.

Аналіз існуючих програмних продуктів дозволив скласти основні вимоги до програмного забезпечення, яке буде розроблено в моєму кваліфікаційній бакалаврській роботі:

1. Система має працювати на персональних ПК невеликої потужності;
2. Повинна підтримуватись ОС Windows останніх версій;
3. Програма не повинна перевантажувати систему користувача;
4. Система повинна мати зручний та легкозрозумілий інтерфейс.

Розроблена в цій роботі система дозволяє без зайвих зусиль та вивчення складного інтерфейсу просто зрозуміти користувачу як з нею працювати.

Для реалізації поставленої мети в роботі над проектом необхідно виконати

					ВКРМ-123.21.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

наступні завдання:

- 1) необхідно провести аналіз літературних та Інтернет-джерел з даної тематики;
- 2) дослідити існуючі програмні аналоги, визначити їх переваги та недоліки;
- 3) проаналізувати засоби розробки програм і обрати найбільш відповідний даному проекту;
- 4) розробити структурну схему системи;
- 5) розробити функціональну схему системи та діаграму процесів;
- 6) розробити алгоритми роботи програмного забезпечення;
- 7) розробити програмне забезпечення для вирішення поставленої науково-практичної задачі.

					ВКРМ-123.21.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Першим кроком до розробки програмного забезпечення є необхідність провести дослідження програмного оточення задачі, визначити, які системні компоненти потрібно використовувати в ході роботи, щоб найефективніше вирішити поставлену задачу, який вони мають інтерфейс, і які функції можуть надавати розробнику.

Для створення програмного забезпечення у цій роботі було обрано середовище розробки Visual Studio.

Visual Studio – інтегроване середовище розробки програмного забезпечення від фірми Microsoft. Дане середовище дозволяє створювати різноманітні програмні продукти: консольні програми, програми з графічним інтерфейсом, наприклад віконні додатки Windows Forms, а також Web-додатки тощо.

Середовище Visual Studio дозволяє розробляти Додатки, використовуючі Різні мови програмування: Visual C #, Visual Basic, Visual F #, Visual C ++, Python и т.д. Також існує можливість розробляти Додатки не тільки під Windows, а і під інші популярні платформи: Android, iOS

Версія Visual Studio Community є абсолютно безкоштовна для учнів, студентів та розробників програм з відкритим програмним кодом.

Visual Studio побудована в архітектурі, що підтримує можливість використання доповнень (Add-Ins), — плагінів від сторонніх розробників, що дозволяє розширювати можливості середовища розробки.

Середовище Visual Studio має багато важливих інструментів, які як:

1) Оглядач рішень дозволяє переглядати файли коду, пересуватися по ньому і управляти ними. Оглядач рішень дозволяє впорядкувати код шляхом

					ВКРМ-123.21.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

об'єднання файлів в рішення і проекти. У вікні редактора де ви, швидше за все, будете проводити більшу частину часу, відображається вміст файлу. Тут ви можете редагувати код або розробляти призначений для користувача інтерфейс, наприклад вікно з кнопками або текстові поля.

2) Team Explorer дозволяє відслідковувати робочі елементи і використовувати код спільно з іншими користувачами за допомогою технологій управління версіями, таких як Git і система управління версіями Team Foundation (TFVC).

3) Середовище Visual Studio доступна для Windows і Mac. Функції Visual Studio для Mac багато в чому аналогічні можливостям Visual Studio 2019 і оптимізовані для розробки кроссплатформених і мобільних додатків. Ця стаття присвячена версії Visual Studio 2019 для Windows.

Нижче перераховані деякі популярні можливості Visual Studio, які допоможуть підвищити продуктивність розробки програмного забезпечення:

1) Хвилясті лінії і швидкі дії. Хвилясті лінії позначають помилки або потенційні проблеми коду прямо під час введення. Ці візуальні підказки дозволяють усувати проблеми негайно і не чекати, поки помилка буде виявлена під час збирання або запуску програми. Якщо навести курсор миші на хвилясту лінію, на екран будуть виведені додаткові відомості про помилку. Крім того, в поле зліва може з'явитися значок лампочки зі швидкими діями щодо усунення помилки.

2) Очистка коду. Можна одним натисканням кнопки відформатувати код і застосувати до нього виправлення, запропоновані параметрами стилю коду, угодами в файлі EditorConfig і (або) аналізаторами Roslyn. Очищення коду допомагає усунути багато проблем в коді ще до перевірки коду

3) Рефакторинг. Включає в себе такі операції, як інтелектуальне перейменування змінних, витягування однієї або декількох рядків коду в новий метод, зміна порядку параметрів методів і багато іншого.

					ВКРМ-123.21.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

успішного підключення програма може дати можливість побачити такі показники:

- 1) Температура вхідного потоку води. Температура води, що надходить з теплової мережі у будівлю;
 - 2) Температура вихідного потоку води. Температура води що виходить з будівлі;
 - 3) Об'єм води що пройшов по будівлі за певний період часу;
 - 4) Температура приміщень у будівлі;
 - 5) Температура зовнішнього повітря;
- б) Збереження даних.

В режимі збереження даних програма зберігає показники що вона отримує в режимі перегляду. Для цього присутня спеціальна кнопка у вікні інтерфейсу;

в) Перегляд архівів. В режимі перегляду архівів програма дає можливість передивитись збережені дані та прослідкувати за роботою системи тепlopостачання.

г) Порівняння даних. В режимі порівняння даних, програма має змогу, маючи дані за різні проміжки часу, провести їх порівняння. Це робиться для того щоб можна було побачити зміни в роботі системи тепlopостачання. Як позитивні так і негативні.

Також для зберігання даних використовується база даних, що знаходиться на сервері. Тобто таким чином ми можемо бути впевнені, дані які ми зібрали не пропадуть в т ому разі, якщо з комп'ютером щось станеться. Це також сприяє надійності зберігання даних.

Для взаємодій обладнання встановленого на об'єкті з програмним забезпеченням використовується сгенерована контролером HTML сторінка.

HTML – це мова тегів, якою пишуться гіпертекстові документи для мережі Інтернет. Веб-браузери отримують HTML-документи з веб-сервера або з локальної пам'яті і передають документи в мультимедійні веб-сторінки. HTML

					ВКРМ-123.21.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

описує структуру веб-сторінки семантично і спочатку включені сигнали для зовнішнього вигляду документа.

HTML впроваджує засоби для:

- 1) створення структурованого документа шляхом позначення структурного - складу тексту: заголовки, абзаци, списки, таблиці, цитати та інше;
- 2) отримання інформації із Всесвітньої мережі через гіперпосилання;
- 3) створення інтерактивних форм;
- 4) включення зображень, звуку, відео, та інших об'єктів до тексту.

3.2 Розробка структурної схеми

При розробці структурної схеми (рис. 3.1) програмного забезпечення були розглянуті основні модулі програми.

Структурна схема розділена на три блоки, а саме:

- а) обладнання на об'єкті;
- б) головний сервер;
- в) програмне забезпечення.

У першому блоці показано функціонування обладнання на об'єкті.

Розглянемо функції та елементи обладнання, які потрібні для отримання показників системі теплопостачання.

а) Обладнання для отримання показників представлене датчиками які прикріплюються хомутами до труб теплопостачання будівлі. Також для зменшення похибки датчиків, вони ізолюються за допомогою скловолокна.

Також використовується датчик об'єму води, задача якого отримувати показник кількості води, яка пройшла через будівлю за певний період часу.

б) Обладнанням для збору показників з датчиків є універсальний контролер який збирає інформацію з вс іх датчиків встановлених на об'єкті. Для кожного контролера задається своя окрема IP адреса. Його наступною дією є генерування HTML сторінки до якої має програмне забезпечення може отримати доступ.

					ВКРМ-123.21.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

в) Генерація HTML сторінки для можливості перегляду даних через мережу. Також кожному показнику надається свій тег для того щоб його можна було відрізнити.

У другому блоці структурної схеми знаходиться сервер, який виконує роль шлюза через який ми можемо отримати доступ до показників що збираються в контролері.

У третьому блоці структурної схеми описується програмне забезпечення та його взаємодія з обладнанням, що знаходиться на об'єкті.

а) Підключення до HTML сторінки через головний сервер.

Для цього використовується IP адреса сторінки а саме внутрішня адреса що записана на сервері.

б) Пошук даних по тегам.

До кожного елемента в програмі, що показує значення показника температури або об'єму води в мережі теплопостачання будівлі, прив'язаний тег що відповідає за конкретний показник. Це дозволяє отримати потрібне значення в програмі та чітко бачити до чого воно відноситься

					ВКРМ-123.21.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

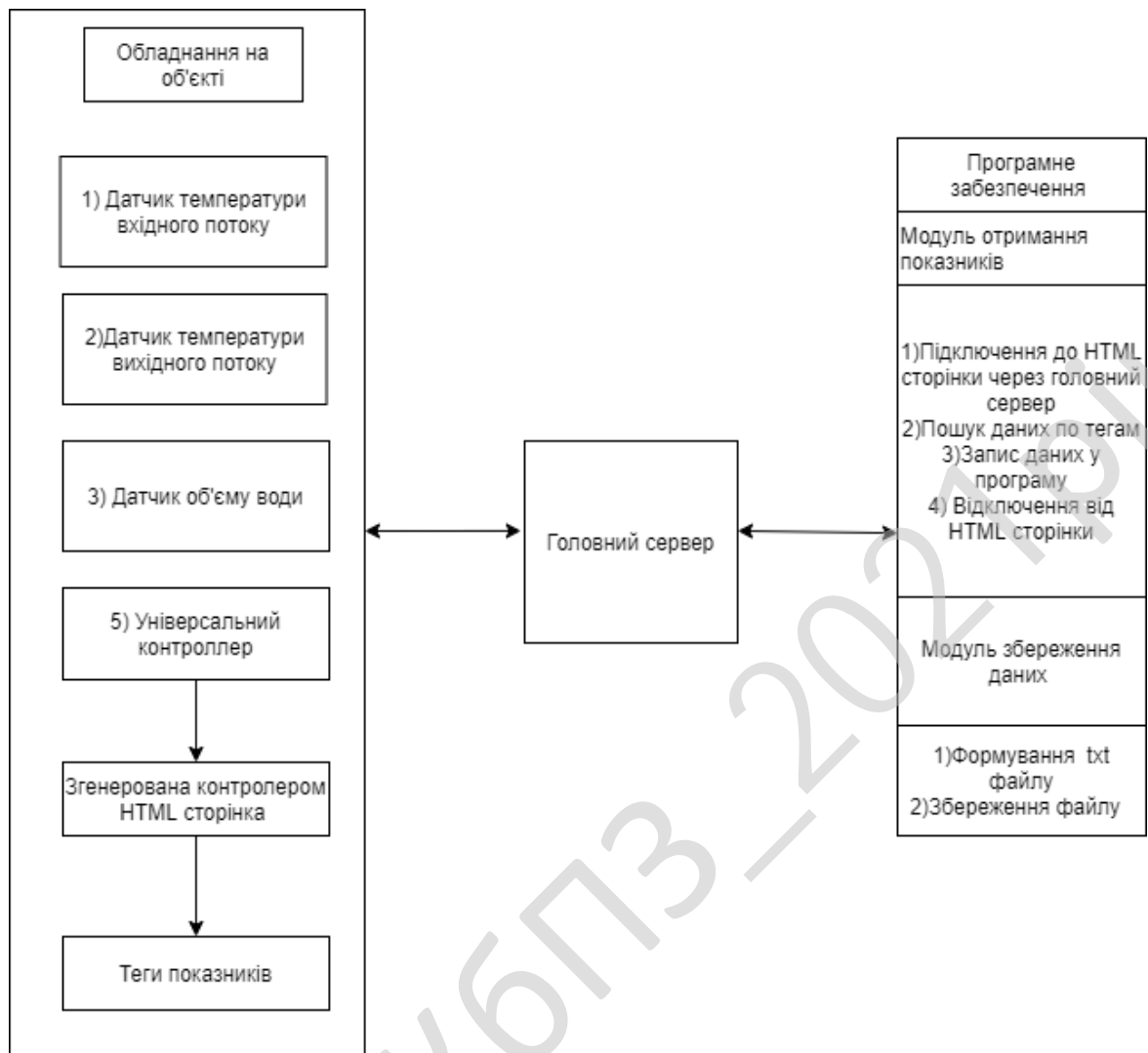


Рисунок 3.1 – Структурна схема

3.3 Розробка функціональної схеми

Функціональна схема (рисунок 3.2) розбита на три блоки перерахуємо їх з низу вгору:

а) Нижній блок функціональної схеми це блок сгенерованої контролером HTML-сторінки .

б) Середній блок функціональної схеми описує взаємодію основного модуля програми з функціональними модулями і взаємодією з в верхнім і нижнім функціональним блоком схеми.

в) Верхній блок відповідає за перевірку існування шляхів до файлу

програмного забезпечення і запуск програмного забезпечення.

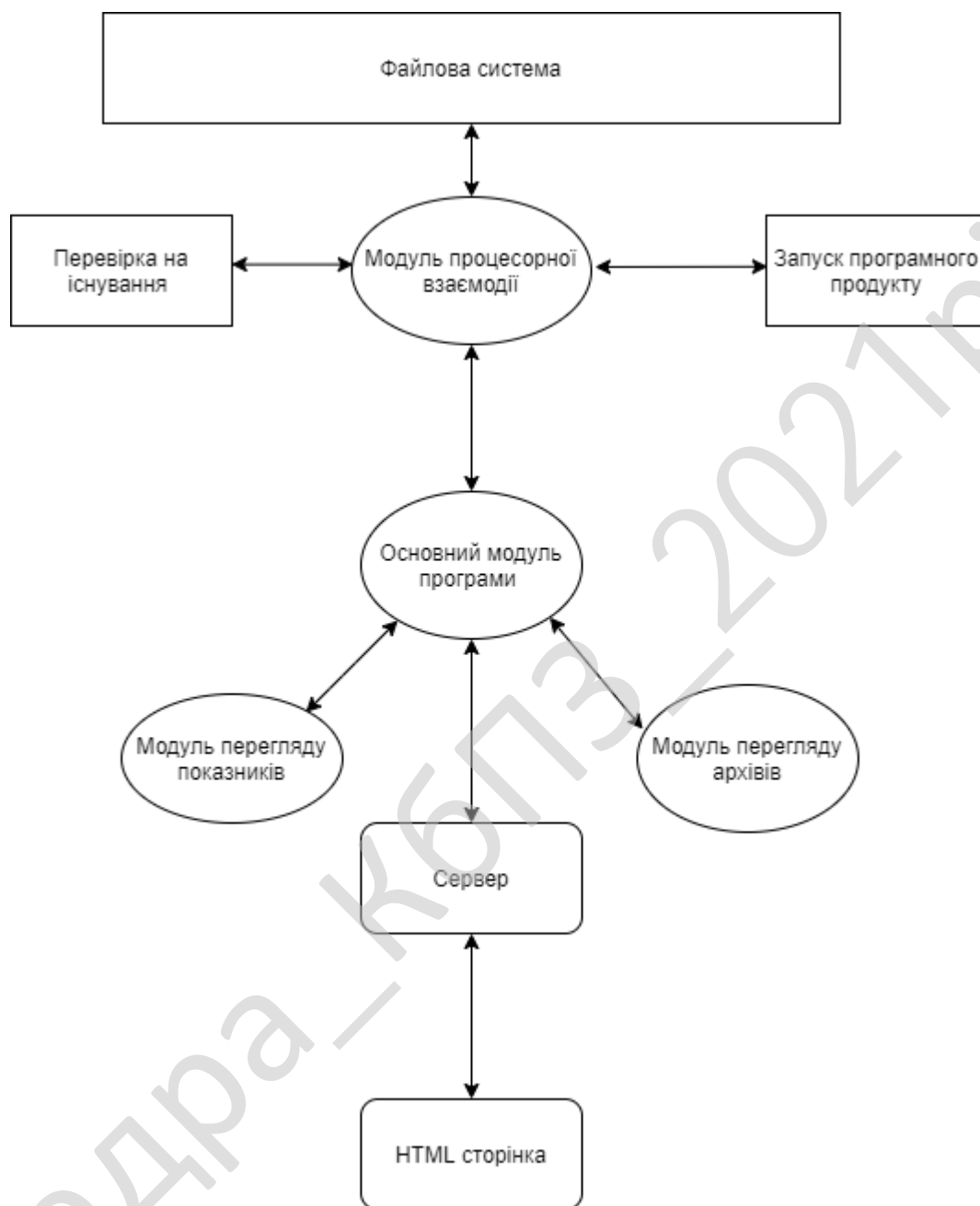


Рисунок 3.2 – Функціональна схема системи

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-123.21.0008.00.00.ПЗ

Арк.

32

3.4 Розробка діаграми процесів

На діаграмі процесів системи (рис. 3.3) відображено взаємодію процесів розробленої системи та порядок дій, що відбуваються під час її роботи.

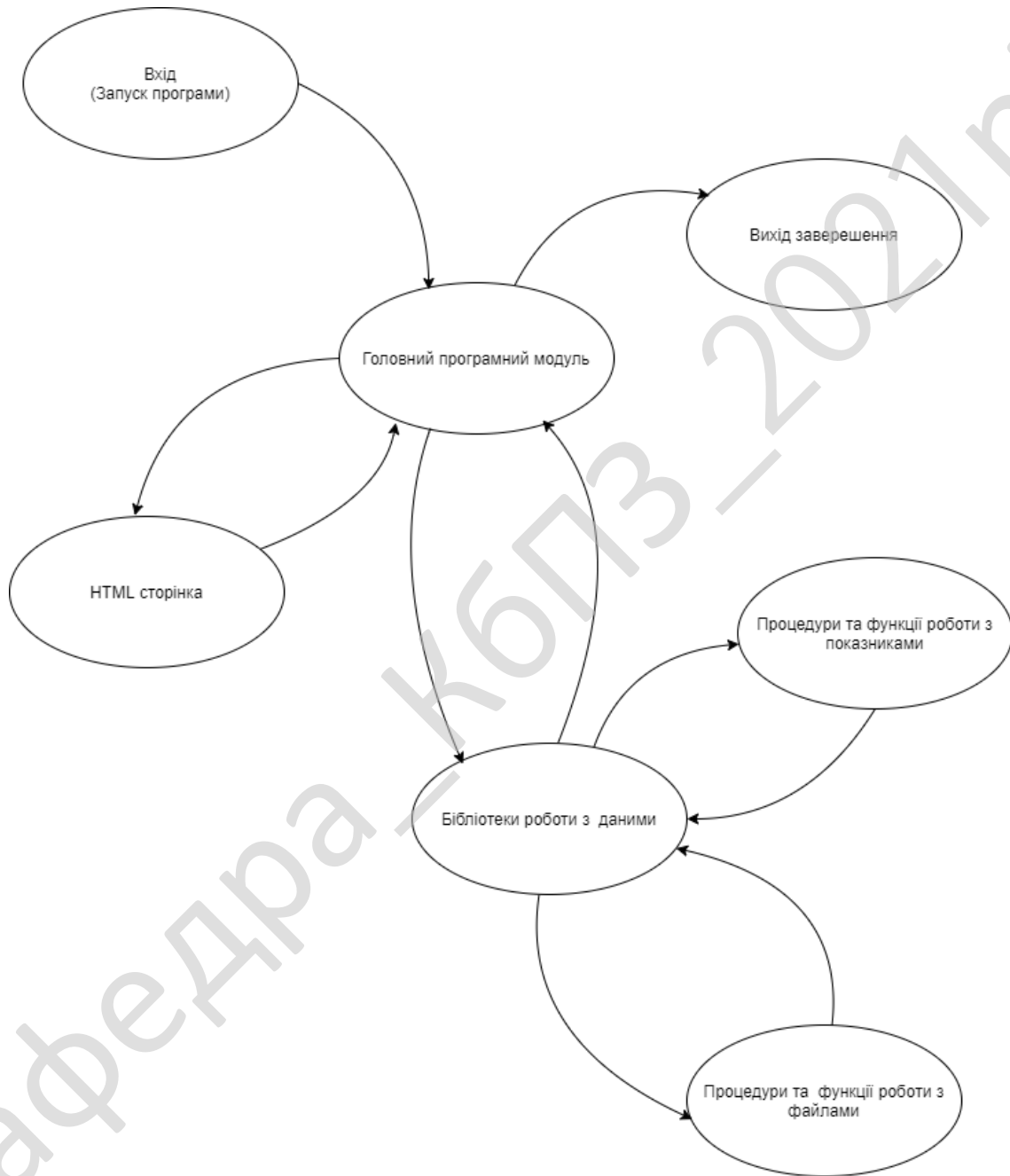


Рисунок 3.3 – Діаграма процесів системи

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-123.21.0008.00.00.ПЗ

Арк.

33

4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ

Після вибору засобів програмування та створення функціональної, структурної схем та діаграми процесів, перейдемо до розробки алгоритмів системи та програмного забезпечення для рішення поставленої задачі.

4.1 Блок-схеми та опис алгоритмів функціонування системи

Першим кроком перед розробкою блок-схеми програми, стояв вибір мови програмування, виходячи з вибору мови програмування C# і інших технічних засобів, зокрема середовищ розробки (Microsoft Visual Studio та Unity), алгоритм програми відповідно буде об'єктно-орієнтованим. Виходячи з цього при складанні блок-схеми основна увага надавалася блоку обробки запитів користувача і підпрограмі запитів користувача.

При розробці блок-схеми основної програми, були виділені такі стадії її роботи:

1) Ініціалізація. Первинна стадія роботи програми, на даному етапі відбувається ініціалізація призначених для користувача типів даних і відповідно і ініціалізація змінних і масивів.

2) Блок перевірки програмного забезпечення. На стадії перевірки програми, відбувається перевірка на цілісність і наявність файлів, фізичного файлу збереженої інформації.

3) Підключення до сторінки контролера. Стадія підключення до контролера(пристрою) через сервер який об'єднує всі об'єкти.

					ВКРМ-123.21.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

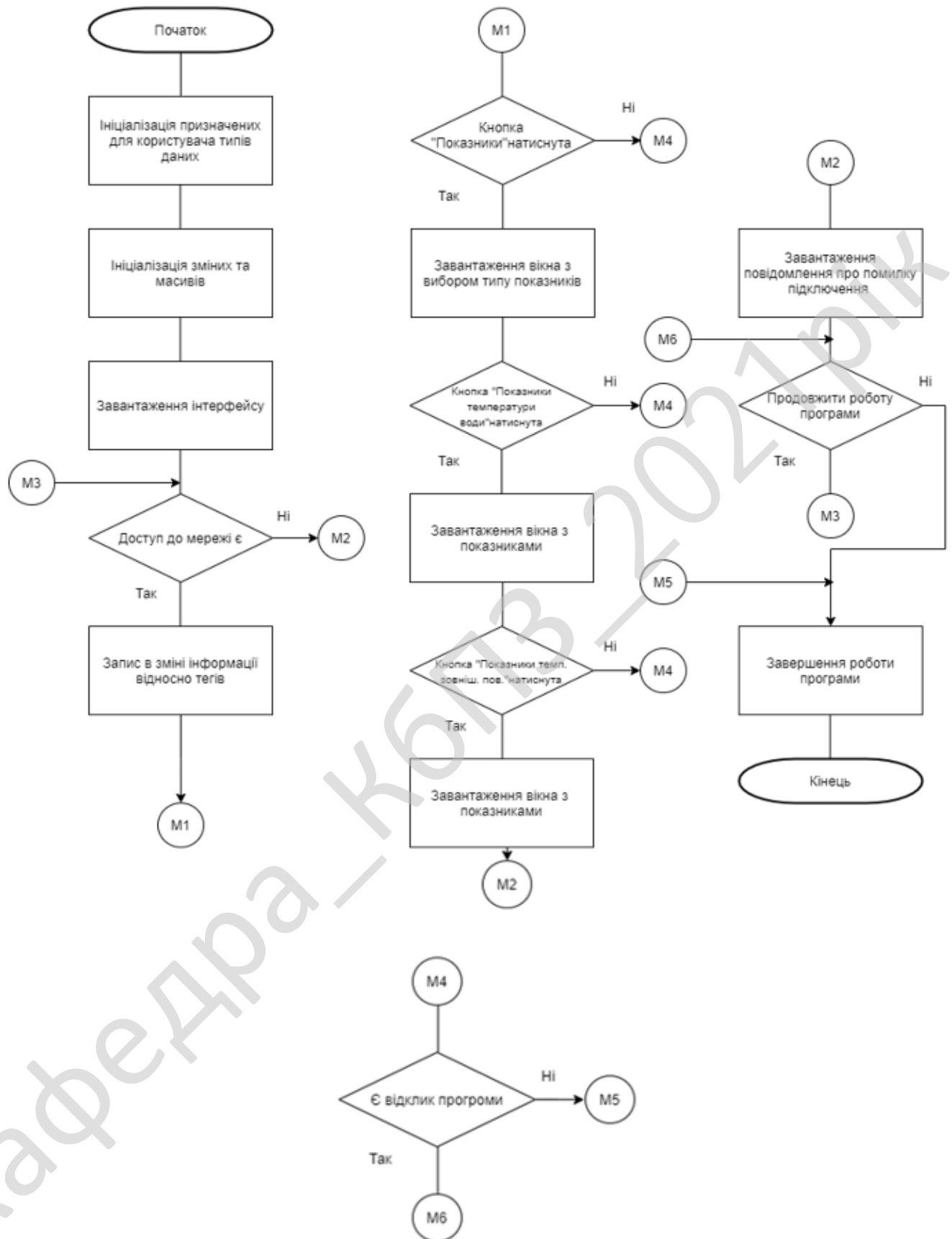


Рисунок 4.1 - Блок-схема головного модуля

4) Блок роботи програмного забезпечення. Основний блок програми після перевірки цілісності програмного забезпечення перевірки бази даних і підключення її, виконується підпрограма запиту користувача де описаний основний алгоритм обробки запитів користувача. Виходячи з функціональних можливостей програми описаних у розділі 4.1 всі запити користувача (адміністрування, запит даних, видалення даних) об'єднані в одну підпрограму.

5) Завершення роботи програми. Блок завершення програми винесений також в окремий модуль, в якому перед завершення роботи програми відбувається відключення від сторінки контролера з перевіркою на активні запити, і звільненням виділених ресурсів програми.

Для реалізації політики управління правами доступу та надання користувачам системи можливості реєстрації під різними ролями з різними правами доступу було розроблено наступний модуль:

```
using System;
using System.Collections.Generic;
using System.Web.Security;

namespace Auditor
{
    public static class RoleManagement
    {
        public static void RoleRegister(string roleName, string description)
        {
            if (!Roles.RoleExists(roleName))
            {
                Roles.CreateRole(roleName);
                RoleUpdate(roleName, description);
            }
            else
            {
                throw new Exception("Role already exists!");
            }
        }

        public static void RoleUpdate(string roleName, string description)
        {

```

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-123.21.0008.00.00.ПЗ

Арк.

36

```

        if (Roles.RoleExists(roleName))
        {
            string query = @"UPDATE [aspnet_Roles]
                               SET [description] = @description
                               WHERE [loweredrolename] = @rolename;";
            var parameters = new Dictionary<string, object>() { {
                "description", description },
                {
                "rolename", roleName } };
            DatabaseUtils.ExecuteNonQuery(query, parameters,
            DatabaseUtils.ConnectionTarget.Services);
        }
        else
        {
            throw new Exception("Role does not exist!");
        }
    }

    public static void RoleDelete(string roleName)
    {
        if (Roles.RoleExists(roleName))
        {
            string[] usersInRole = Roles.GetUsersInRole(roleName);
            if (usersInRole.Length > 0)
            {
                Roles.RemoveUsersFromRole(usersInRole, roleName);
            }
            Roles.DeleteRole(roleName);
        }
        else
        {
            throw new Exception("Role does not exist!");
        }
    }
}

```

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-123.21.0008.00.00.ПЗ

Арк.

37

Розроблене програмне забезпечення служить універсальним інструментом в руках програміста або системного адміністратора. При повному або частковому встановленні ліцензійного програмного забезпечення на одній або декількох клієнтських машин, що обслуговуються, при певних ситуаціях, таких як: збій операційної системи; крах апаратної частини комп'ютера; оновлення версії ліцензійного програмного забезпечення або необдумані дії користувача. В такому разі дані що біли отримані не будуть втрачені адже вони будуть продубльовані в пам'яті самого контролера.

Програма призначена для аудиту так контролю систем мережі теплопостачання що встановлюються в житлових будівлях. Таким чином, здійснюється швидкий доступ до будь якого з об'єктів де встановлене обладнання. У розробленому програмному забезпеченні можна відразу побачити на якому з об'єктів сталася поломка адже навіть те що один з датчиків перестане працювати або буде показувати неправдиву інформацію може означати що робота на об'єкті порушена.

Програма дозволяє своєчасно виявляти несправності на об'єктах. ПЗ полегшить роботу всім аудиторам різних організацій, в управлінні яких є велика кількість об'єктів, а також може допомогти аудиторам що мають перевіряти якість роботи системи.

Програмне забезпечення працює майже безвідмовної більш зручне у використанні, в порівнянні з іншими програмами використовуючи технологію ІОТ без високих вимог до конфігурації комп'ютера. Її надійність полягає у тому, що файл з даними про показники може зберігатись і переглядатись у будь який час. Також до особливостей ПЗ відноситься можливість перегляду кількості води що пройшла за певний проміжок часу.

Блок-схеми розроблених процедур для одержання отриманих під час проведення аудиту значень параметрів системи наведені на рис. 4.2.

					ВКРМ-123.21.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

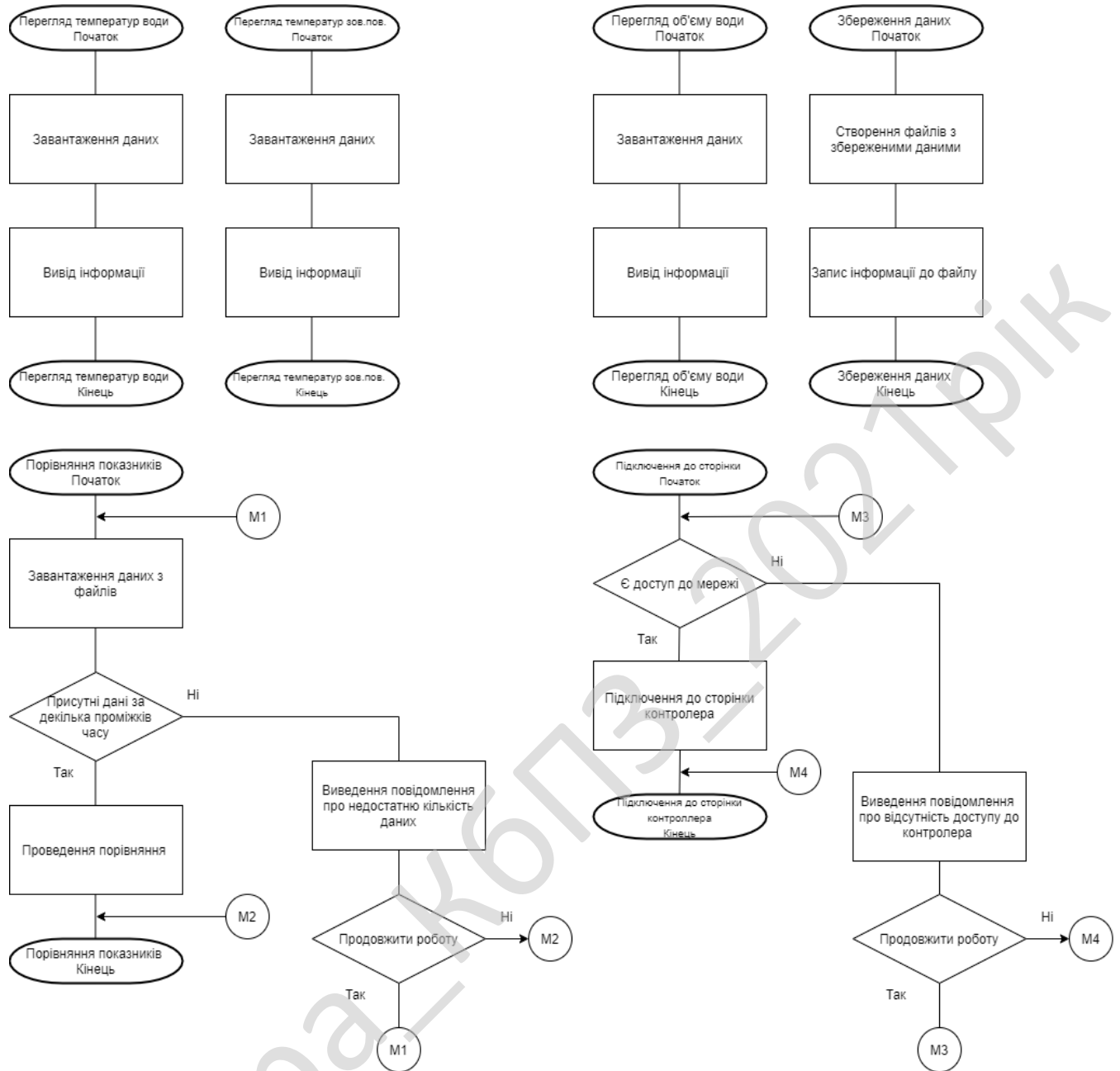


Рисунок 4.2 - Блок-схеми процедур системи

Збір інформації про систему повністю автоматизований. Користувачу залишиться тільки виконувати необхідні дії в заданій послідовності.

Перерахуємо можливості розробленого програмного забезпечення:

1. Забезпечує швидкий доступ до різних об'єктів;
2. Використовується для створення записів про зміну температур та інших показників;
3. Можливість швидкого запуску тому що не потребує інсталяції ;

4. Не вимоглива до конфігурації комп'ютера;;
5. Простота в роботі (обігу);
6. Високий рівень захисту від перехоплень пакетів з даними;
7. Автоматична функція перевірки підключення до сервера;
8. Стабільна робота у всіх версіях MS Windows;
9. Простота і наочність роботи в програмі.

Для роботи з базою даних системи розроблено наступний модуль:

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Web.Configuration;

namespace Auditor
{
    public static class DatabaseUtils
    {
        public enum ConnectionTarget
        {
            Services,
            App
        }

        public static string ConnectionString(ConnectionTarget target)
        {
            string connectionString;
            switch (target)
            {
                case ConnectionTarget.Services:
                    connectionString =
                        WebConfigurationManager.ConnectionStrings["csServices"].ConnectionString;
                    break;

                case ConnectionTarget.App:
                    connectionString =
                        WebConfigurationManager.ConnectionStrings["csApp"].ConnectionString;
                    break;
            }
        }
    }
}
```

					ВКРМ-123.21.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

```

        default:
            connectionString = null;
            break;
    }
    return connectionString;
}

/// <summary>
/// Execute SQL query. Returns value from SCOPE_IDENTITY()
/// </summary>
public static int? ExecuteIdentityQuery(string query) =>
ExecuteIdentityQuery(query, null, ConnectionTarget.App);

public static int? ExecuteIdentityQuery(string query,
Dictionary<string, object> parameters) => ExecuteIdentityQuery(query,
parameters, ConnectionTarget.App);

public static int? ExecuteIdentityQuery(string query, ConnectionTarget
target) => ExecuteIdentityQuery(query, null, target);

public static int? ExecuteIdentityQuery(string query,
Dictionary<string, object> parameters, ConnectionTarget target)
{
    int? scopeIdentity = null;
    QueryValidation(query);
    var parametersList = CreateParametersList(parameters);
    ParametersValidation(parametersList);
    query = query + Environment.NewLine + "SELECT SCOPE_IDENTITY()";
    try
    {
        using (SqlConnection connection = new
SqlConnection(connectionString(target)))
            using (SqlCommand command = new SqlCommand(query, connection))
            {
                if (parametersList != null)
                {
                    command.Parameters.AddRange(parametersList.ToArray());
                }
                connection.Open();
                scopeIdentity =
Utils.ConvertToNullableInt(command.ExecuteScalar());
                command.Parameters.Clear();
            }
    }
}

```

Вим.	Арк.	№ докум.	Підпис	Дата

БКРМ-123.21.0008.00.00.ПЗ

Арк.

41

```

        connection.Close();
    }
}
catch (Exception ex)
{
    throw;
}
return scopeIdentity;
}

/// <summary>
/// Execute SQL query. Returns value from ROWS_AFFECTED
/// </summary>

public static int ExecuteNonQuery(string query) =>
ExecuteNonQuery(query, null, ConnectionTarget.App);

public static int ExecuteNonQuery(string query, Dictionary<string,
object> parameters) => ExecuteNonQuery(query, parameters,
ConnectionTarget.App);

public static int ExecuteNonQuery(string query, ConnectionTarget
target) => ExecuteNonQuery(query, null, target);

public static int ExecuteNonQuery(string query, Dictionary<string,
object> parameters, ConnectionTarget target)
{
    int rowsAffected;
    QueryValidation(query);
    var parametersList = CreateParametersList(parameters);
    ParametersValidation(parametersList);
    try
    {
        using (SqlConnection connection = new
SqlConnection(ConnectionString(target)))
        using (SqlCommand command = new SqlCommand(query, connection))
        {
            if (parametersList != null)
            {
                command.Parameters.AddRange(parametersList.ToArray());
            }
            connection.Open();

```

Вим.	Арк.	№ докум.	Підпис	Дата

БКРМ-123.21.0008.00.00.ПЗ

Арк.

42

```

        rowsAffected = command.ExecuteNonQuery();
        command.Parameters.Clear();
        connection.Close();
    }
}
catch (Exception ex)
{
    throw;
}
return rowsAffected;
}

/// <summary>
/// Execute SELECT SQL query. Returns DataTable
/// </summary>

public static DataTable ExecuteSelectQuery(string query) =>
ExecuteSelectQuery(query, null, ConnectionTarget.App);

public static DataTable ExecuteSelectQuery(string query,
Dictionary<string, object> parameters) => ExecuteSelectQuery(query,
parameters, ConnectionTarget.App);

public static DataTable ExecuteSelectQuery(string query,
ConnectionTarget target) => ExecuteSelectQuery(query, null, target);

public static DataTable ExecuteSelectQuery(string query,
Dictionary<string, object> parameters, ConnectionTarget target)
{
    DataTable result = new DataTable();
    QueryValidation(query);
    var parametersList = CreateParametersList(parameters);
    ParametersValidation(parametersList);
    try
    {
        using (SqlConnection connection = new
SqlConnection(ConnectionString(target)))
            using (SqlCommand command = new SqlCommand(query, connection))
                using (SqlDataAdapter adapter = new SqlDataAdapter(command))
                    {
                        if (parametersList != null)
                            {

```

Вим.	Арк.	№ докум.	Підпис	Дата

БКРМ-123.21.0008.00.00.ПЗ

Арк.

43

```

        command.Parameters.AddRange(parametersList.ToArray());
    }
    connection.Open();
    adapter.Fill(result);
    command.Parameters.Clear();
    connection.Close();
}
}
catch (Exception ex)
{
    throw;
}
return result;
}

/// <summary>
/// Execute SELECT SQL query. Returns object
/// </summary>
public static object ExecuteScalarSelectQuery(string query) =>
ExecuteScalarSelectQuery(query, null, ConnectionTarget.App);

public static object ExecuteScalarSelectQuery(string query,
Dictionary<string, object> parameters) => ExecuteScalarSelectQuery(query,
parameters, ConnectionTarget.App);

public static object ExecuteScalarSelectQuery(string query,
ConnectionTarget target) => ExecuteScalarSelectQuery(query, null, target);

public static object ExecuteScalarSelectQuery(string query,
Dictionary<string, object> parameters, ConnectionTarget target)
{
    object result;
    QueryValidation(query);
    var parametersList = CreateParametersList(parameters);
    ParametersValidation(parametersList);
    try
    {
        using (SqlConnection connection = new
SqlConnection(ConnectionString(target)))
            using (SqlCommand command = new SqlCommand(query, connection))
            {
                if (parametersList != null)

```

						БКРМ-123.21.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			44

```

        {
            command.Parameters.AddRange(parametersList.ToArray());
        }
        connection.Open();
        result = command.ExecuteScalar();
        command.Parameters.Clear();
        connection.Close();
    }
}
catch (Exception ex)
{
    throw;
}
return result;
}

private static void QueryValidation(string query)
{
    if (string.IsNullOrWhiteSpace(query))
    {
        throw new Exception("Empty SQL query!");
    }
}

private static List<SqlParameter>
CreateParametersList(Dictionary<string, object> parameters)
{
    var parametersList = new List<SqlParameter>();
    if (parameters != null)
    {
        foreach (var parameter in parameters)
        {
            parametersList.Add(new SqlParameter(parameter.Key,
parameter.Value ?? DBNull.Value));
        }
    }
    return parametersList.Count > 0 ? parametersList : null;
}

private static void ParametersValidation(List<SqlParameter>
parameters)
{

```

Вим.	Арк.	№ докум.	Підпис	Дата

БКРМ-123.21.0008.00.00.ПЗ

Арк.

45

```

        if (parameters != null)
        {
            int distinctParameterNames = (from x in parameters
                select
                x.ParameterName).Distinct().Count();
            if (distinctParameterNames != parameters.Count())
            {
                throw new Exception("Parameter names are not unique!");
            }
        }
    }
}

```

4.2 Захист розробленого програмного забезпечення

При розробці методу захисту розробленого ПЗ було створено два рівні захисту програми, верхній рівень – вікно введення пароля на вхід в програмне забезпечення, другий рівень – обов’язкова наявність сертифікатів та приватних ключів для доступу на об’єкти.

Перший рівень захисту створений лише для того щоб, у разі зникнення комп’ютера або ноутбука, на якому встановлено програму, ніхто не міг отримати доступ до інформації про об’єкти аудиту. Також присутні жорсткі вимоги до пароля:

1. Він має містити не менше 10 символів.
2. У складі пароля мають бути присутні як цифри, так і букви.
3. Букви мають бути використані як нижнього, так і верхнього регістра.
4. Пароль не має містити логічних зв’язків. Тобто це має бути випадковий набір букв та цифр.
5. Пароль потрібно змінювати кожні 7 днів.

Другий рівень захисту це наявність спеціальних сертифікатів і ключів, що генеруються на основі даних сервера. Так відбувається захист від можливого проникнення ззовні та втрати важливих даних.

					ВКРМ-123.21.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

Таким чином програмне забезпечення в повній мірі можна використовувати лише в тому випадку, якщо всі умови доступу виконані. Для такого типу програм, дворівневого захисту буде цілком достатньо. Такий захист робиться лише для того, щоб запобігти неприємним випадкам з неавторизованим доступом до мережі пристроїв.

Кафедра КБПЗ – 2021 рік

					ВКРМ-123.21.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Програмне забезпечення призначене для аудиту та моніторингу роботи системи теплопостачання. При запуску головного файлу проекту, подвійним натисненням клавіші миші або натисненням клавіші "Введення" на клавіатурі користувач побачить спочатку вікно авторизації (рис. 5.1), а після успішної авторизації – головне вікно програми (рис. 5.2).



Рисунок 5.1 – Вікно авторизації

Для початку роботи процесу установки необхідно підключитися до сервера. Для цього необхідно натиснути на кнопку «Початок» у верхньому меню основного вікна програми (рис. 5.2). При вдалому підключенні на екрані з'явиться вікно з повідомленням про вдале підключення до обладнання через сервер.

					ВКРМ-123.21.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48



Рисунок 5.2 – Головне вікно програми

Перерахуємо функціональні кнопки програмного забезпечення, головного вікна програми.

1) Початок роботи. Запуск робочих вікон та підключення до обладнання через сервер (рис. 5.3).

2) Інформація про програму. При натисненні на цю клавішу програма видає інформацію про розробника (рис. 5.4).

3) При натисненні на кнопку «Вихід» користувач має можливість завершити роботу з програмним забезпеченням.

Після переходу на початок роботи програми відкривається вікно з вибором режиму роботи (рис. 5.3), що містить наступні кнопки:

1) Показники. Кнопка «Показники» дозволяє поповнити переглянути показники температур та води в реальному часі.

2) Архіви. Кнопка «Архіви» дозволяє переглянути збережені дані. Це дозволяє завжди проконтролювати зміни в роботі системи теплопостачання. Також ці дані зберігаються у пам'яті контролера. Таким чином можливість втрати даних мінімізується.

3) Порівняти. Дозволяє порівняти роботу системи мережі тепlopостачання як без нового обладнання, так із ним.

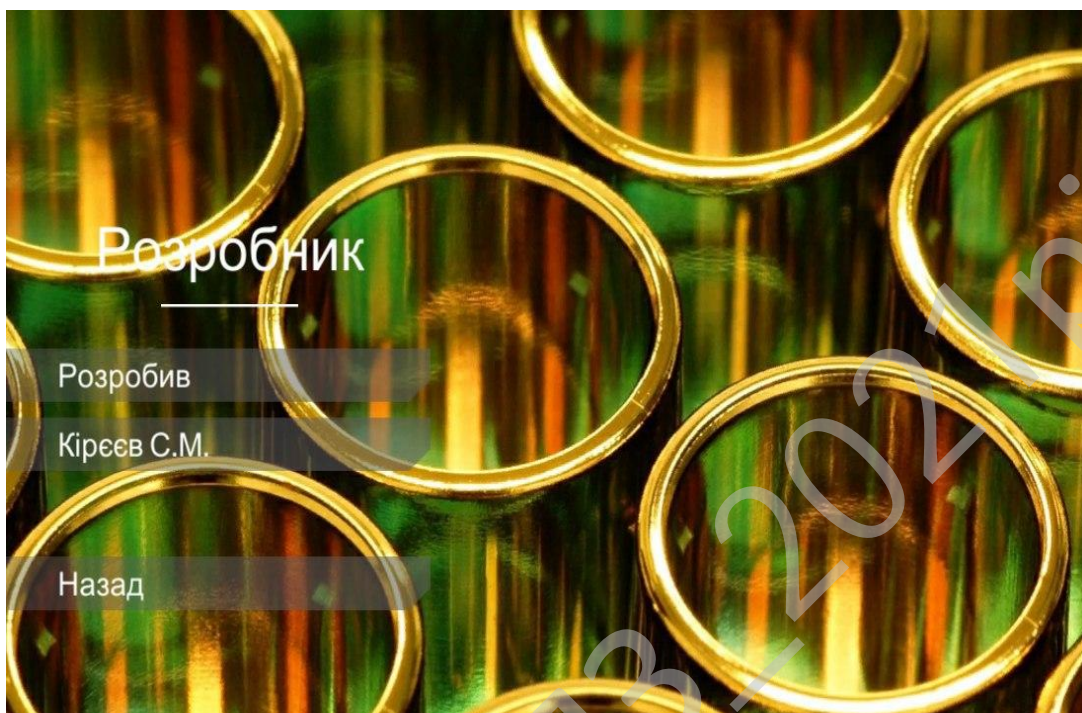


Рисунок 5.3 Інформація про розробника

4) Повернення назад. Ця кнопка є завжди необхідною. Натиснувши її користувач може повернутися попереднє вікно та виконати іншу дію.



Рисунок 5.4 – Вікно з вибором режиму роботи

6 НАУКОВА НОВИЗНА

Метою роботи є дослідження та програмна реалізація системи аудиту мережі тепlopостачання з використанням технології IoT.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

- Огляд існуючих систем аудиту мережі тепlopостачання.
- Розробка системи аудиту мережі тепlopостачання з використанням технології IoT.
- Випробовування розробленої системи аудиту мережі тепlopостачання з використанням технології IoT.

Об'єктом дослідження є процес аудиту мережі тепlopостачання.

Предметом дослідження є методи аудиту мережі тепlopостачання з використанням технології IoT.

Методи дослідження базуються на методах об'єктно-орієнтованого програмування, теорії інформації та кодування, методах роботи з комп'ютерними системами та мережами, технології Інтернету речей.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

- Удосконалено метод аудиту мережі тепlopостачання з використанням технології IoT.
- Розроблено вітчизняний продукт аудиту мережі тепlopостачання з використанням технології IoT, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі аудиту мережі тепlopостачання з використанням технології IoT.

					ВКРМ-123.21.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

7 ДАНІ ПРО ЕКОНОМІЧНУ ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ

7.1 Техніко-економічне обґрунтування теми магістерської роботи

Після ознайомлення з підприємством та засобами розробки програмної продукції був розроблений план розробки програми. Був підрахований необхідний час для розробки та впровадження програми. Цей час склав 60 днів (три місяці).

В магістерській роботі було проведено дослідження та виконана програмна реалізація системи аудиту мережі теплопостачання з використанням технології IoT.

Розроблене програмне забезпечення має достатню надійність і задовольняє усім поставленим умовам, а саме:

- а) невеликий розмір;
- б) невеликі системні потреби;
- в) незалежність від встановлених на комп'ютері баз даних;
- г) зручність у користуванні та надійність.

Таблиця 7.1 – Початкові дані

Показники	Позначення	Характеристика або величина
1	2	3
1. Кількість розроблених програм період, шт.	N	1
2. Кількість екземплярів програм, шт.	Ne	80 (ост. цифра № зал *10 ¹)
3. Запланований термін розробки, днів	Fpq	60 (3 місяці)
4. Група задачі підсистеми управління (1-6)	–	1
5. Ступінь новизни задачі (А, Б, В, Г)	–	Б
6. Складність алгоритму (1, 2, 3)	–	2

Продовження таблиці 7.1

1	2	3
7. Кількість макетів вхідної інформації	–	3
8. Кількість форм вихідної інформації.	–	4
9. Мова програмування (1-6)	–	3
10. Попередній досвід (1-6)	–	3
11. Гнучкість проекту ПП (1-6)	–	3
12. Детальність проекту ПП (1-6)	–	2
13. Рівень спрацьованості колективу (1-6)	–	2
14. Ступінь вимірності процесів (1-6)	–	3
15. Необхідна надійність програмного забезпечення (1-6)	–	2
16. Розмір бази даних (порівняно з розміром програми) (1-6)	–	2
17. Складність кінцевого програмного продукту (1-6)	–	2
18. Необхідний рівень забезпечення повторного використання (1-6)	–	2
19. Документованість відповідно до планованого життєвого циклу (1-6)	–	2
20. Вимоги до швидкодії ПП (1-6)	–	2
21. Обмеження на розміри основного сховища даних (1-6)	–	2
22. Різноманітність використовуваних обчислювальних платформ (1-6)	–	2
23. Професійний рівень аналітиків (1-6)	–	2
24. Професійний рівень програмістів (1-6)	–	2
25. Постійність складу команди розробників (1-6)	–	2
26. Досвід розробки додатків (1-6)	–	2
27. Досвід роботи з обчислювальною платформою (1-6)	–	2

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-123.21.0008.00.00.ПЗ

Арк.

53

Продовження таблиці 7.1

1	2	3
28. Досвід роботи з мовою і інструментами середовища розробки (1-6)	–	2
29. Досвід роботи з програмними інструментами розробки (1-6)	–	3
30. Розробка ПЗ для декількох серверів одночасно (1-6)	–	2
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн.	–	80000 (2 ост. цифри № зал*10 ⁴)
33. Норматив додаткової зарплати, % :	Нд	10
34. Норматив відрахувань у соціальні фонди, %	Нс	22
35. Норматив загальногосподарських витрат, %	Нг	15
36. Норматив витрат на освоєння нових мов програмування, %	Нп	15
37. Рівень рентабельності програмної продукції, %	Ре	50
38. Ставка податку на додану вартість, %	Ндв	20

7.2 Розрахунок трудомісткості розробки програмної продукції

Значення трудомісткості розробки програмного забезпечення для стадій ТЗ, ЕК, ТП та ВП визначаємо по типовим нормам часу приведеним в додатках МВ. Стадія РП є найбільш тривалою і трудомісткою, що робить значний вплив на інші стадії проекту.

Визначимо трудомісткість розробки ПЗ для стадії РП.

Обчислюємо номінальні трудовитрати, люд-міс.:

$$T_{ном} = A \text{ Size}^B, \quad (7.1)$$

де: A – коефіцієнт Боема, $A = 2,45$;

Size – загальний об'єм відлагодженого програмного коду, тис. рядків;

B – показник ступеня, що визначається співвідношенням:

$$B = 1,01 + 0,001 \sum W_i, \quad (7.2)$$

де: W_i – сумарне значення п'яти показників (МВ, додаток 2), що відображають особливості розробки проекту програмного продукту (ПП) і колективу розробників.

$$B = 1,01 + 0,001(2,43 + 3,64 + 3,38 + 3,95 + 2,73) = 1,027.$$

$$T_{ном} = 2,45 \cdot 2,7^{1,026} = 6,78 \text{ люд-міс.}$$

Визначаємо уточнені (з урахуванням приведених в МВ додатку 3 сімнадцяти додаткових коефіцієнтів) трудовитрати, люд-міс.:

$$T_{уточн} = T_{ном} \prod V_j, \quad (7.3)$$

де: $\prod V_j$ – добуток сімнадцяти додаткових коефіцієнтів, приведених в МВ додатку 3.

$$T_{уточн} = 6,78 \cdot (0,88 \cdot 0,93 \cdot 0,88 \cdot 0,91 \cdot 0,95 \cdot 1 \cdot 1 \cdot 0,87 \cdot 1,22 \cdot 1,16 \cdot 1,1 \cdot 1,1 \cdot 1,12 \cdot 1,1 \cdot 1,1 \cdot 1,1) = 9,37 \text{ люд-міс.}$$

Ці коефіцієнти дозволяють диференційовано оцінювати результати роботи програмістів, беручи до уваги швидкодію програми, використання різноманітних обчислювальних платформ і інструментів розробки, взаємодію декількох серверів, вимоги до об'ємів баз даних і ін.

Визначаємо підсумкові трудовитрати по стадії робочий проект, люд-дні:

$$T_{РП} = 0,3 C T_{уточн}^{0,33 + 0,2(B-1,01)} S, \quad (7.4)$$

де: C – визначений емпірично коефіцієнт, запропонований авторами методики, (МВ, додаток 4);

S – коефіцієнт стиснення (або подовження) графіка робіт %, що дозволяє коректувати терміни розробки ПЗ згідно встановленим вимогам. Вибираємо в межах (25...350)%.

$$T_{РП} = 0,3 \cdot 2,85 \cdot 9,37^{0,33 + 0,2(1,026 - 1,01)} \cdot 87 = 168 \text{ люд/день.}$$

					ВКРМ-123.21.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

Для зручності визначення загальної трудомісткості на розробку ПЗ результати розрахунків по стадіям зводимо до таблиці 7.2.

Таблиця 7.2 – Визначення трудомісткості розробки програмного забезпечення

Стадії розробки	Трудомісткість за типовими нормами та розрахунками	
	Величина, люд/дні	Підстава
Технічне завдання	9	Д5
Ескізний проект	10	Д6
Технічний проект	9	Д7
Робочий проект	168	Ф 7.1-7.4
Впровадження	13	Д13
Всього	209	–

7.3 Визначення чисельності виконавців і планового фонду зарплати

Чисельність ставок інженерів-програмістів для розробки ПЗ визначається за формулою:

$$Ч = \frac{T_{nz} N}{F_{pq} - H_{ев}}, \quad (7.5)$$

де: F_{pq} – плановий фонд робочого часу одного спеціаліста, днів;

T_{nz} – трудомісткість розробки програмного забезпечення люд-дні.

$$Ч = \frac{209 \cdot 1}{60 - 5} = 3,8 \text{ ставки.}$$

Чисельність інженерів-електронщиків для проведення технічного обслуговування та ремонту комп'ютерних мереж визначається в залежності від наявності технічних засобів і норм витрат часу на виконання профілактичних робіт на протязі року.

Визначаємо затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за період розробки. Результати розрахунку зводимо

до табл. 7.3.

Таблиця 7.3 – Затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за розрахунковий період

Найменування обладнання	Профілактичне обслуговування			
	Кіл-ть хв. на од. обл.	Кіл-ть обладнання	Затрати часу в хв.	Затрати часу в год.
Системний блок ПК	385	12	4620	77
Монітор	160	12	1920	32
Клавіатура	140	12	1680	28
Маніпулятор «мишка»	30	12	360	6
Принтер матричний	185	1	185	3
Принтер лазерний	355	2	710	12
Принтер струминний	300	1	300	5
Сканер	155	2	310	5
Концентратор – маршрутизатор	155	2	310	5
Кабельні господарства ЛВС на 1 м.п.	2,5	100	250	4
Кабельне господарство електромережі	48	50	2400	40
Копіювальний апарат	285	2	570	10
Усього за рік:			3 _ч	227

Час на профілактику обладнання в загальному балансі робочого часу інженерів-електронщиків не повинен складати більше 10%.

Виходячи з цього фонд робочого часу інженерів-електронщиків складає:

$$\Phi_{ор}^c = \frac{3_{ч} \cdot n_{міс}}{1,2}, \quad (7.6)$$

$$\Phi_{ор}^c = \frac{227 \cdot 3}{1,2} = 567,5 \text{ год.}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

$$Ч_{ел} = \frac{\Phi_{др}^c}{F_{др} \cdot T_{зм}}, \quad (7.7)$$

$$Ч_{ел} = 567,5 / (60 \cdot 8) = 1,2 \text{ ставки.}$$

Для забезпечення нормального технічного обслуговування засобів ТО та мереж, необхідно прийняти найбільше ціле значення розрахункової чисельності інженерів-електронщиків.

Чисельність інженерів-системотехніків, адміністраторів мережі, дизайнерів WEB вузлів, системних програмістів, бухгалтерів-економістів визначається за потребою в залежності від функціональних обов'язків. Після визначення чисельності персоналу складається штатний розклад.

Таблиця 7.4 – Розрахунок чисельності штатного персоналу сектору системного та адміністративного обслуговування засобів ОТ та комп'ютерних мереж

Посада	Вид роботи	Час	К-ть штатних одиниць
Адміністратор загальної мережі, аналітик	Адміністрування локальної мережі, поштового та серверу DNS (OC FreeBSD), маршрутизатора Cisco, доменного контролеру Windows Server 2016, серверу доступу ADSL (OC Linux), налаштування ADSL, VPN, PPPoE, Frame Relay, Wi-Fi	2	0,5
	Налаштування і конфігурування базової станції безпроводного зв'язку (CMTS)	0,5	
	Розробка та впровадження проектів з організації зв'язку між віддаленими об'єктами, ЛОМ	0,5	
	Забезпечення цілодобової роботи зв'язку клієнтів до мережі Інтернет	1	
Всього		4	

Вим.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------

ВКРМ-123.21.0008.00.00.ПЗ

Арк.

58

Продовження таблиці 7.4

Посада	Вид роботи	Час	К-ть штатних одиниць
Продакт-менеджер	Презентації нової продукції, пошук каналів збуту	1	0,25
	Підтримка постійних клієнтів	0,5	
	Оформлення договорів, ведення тендерів	0,25	
	Контроль взаєморозрахунків з постачальниками	0,25	
Всього		2	
Дизайнер WEB	Розробка концепції оформлення та інтерфейсу сайту, оптимізація дизайну існуючих, проектує їх структуру та навігацію	1	0,25
	Створення графічних і стилістичних елементів сайту	0,5	
	Оформлення банерів і промо-сторінок	0,25	
	Розміщення графіки і контенту на Інтернет сторінках	0,25	
Всього		2	
Інженер верстальник	Розробка та верстка макетів рекламної продукції та технічної документації	1	0,25
	Верстка друкованих видань	0,5	
	Додрукова підготовка макетів	0,25	
	Розміщення графіки і контенту на Інтернет сторінках	0,25	
Всього		2	

Вим.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------

ВКРМ-123.21.0008.00.00.ПЗ

Арк.

59

Складемо штатний розклад виконавців.

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньомісячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	1	8000	24000
Продакт-менеджер	0,25	7000	5250
Інженер-програміст	3,8	7000	79800
Інженер-електронщик	1,2	7000	25200
Інженер-системотехнік	0,25	7000	5250
Адміністратор мережі	0,5	7000	10500
Системний програміст	0,25	7000	5250
Дизайнер WEB	0,25	7000	5250
Інженер-верстальник	0,25	7000	5250
Бухгалтер-економіст	0,5	7000	10500
Всього за період розробки	$R_{cn} = 8,25$	-	$\Phi_{роб} = 176250$

Розрахуємо середньоденну зарплату одного виконавця:

$$z_{cd} = \frac{\Phi_{роб}}{R_{cn} F_{pq}}, \quad (7.8)$$

де: $\Phi_{роб}$ – загальна сума зарплати за плановий період, грн.

$$z_{cd} = \frac{176250}{8,25 \cdot 60} = 356 \text{ грн.}$$

7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника

Балансова вартість будівель визначається з урахуванням кількості робочих місць виконавців, питомої площі на одне робоче місце, та вартості одного квадратного метра виробничої площі:

					ВКРМ-123.21.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

$$B_{y\partial} = R_{cn}^1 S_y C_{nl}, \quad (7.9)$$

де: R_{cn}^1 – кількість робочих місць виконавців, шт. Приймаємо 8 робочих місць;

S_y – питома площа на одне робоче місце, m^2 ;

C_{nl} – вартість одного квадратного метра площі, грн.

Згідно даних ТОВ науково-дослідницького консалтингового підприємства «Пектораль» (м. Кіровоград, вул. Глинки 16) ціна одного квадратного метра площі новобудови, вік якої не перевищує 25 років, по місту складає 800...1600 у.о./ m^2 . Враховуючи, що курс складає 1 у.о. = 25 грн. приймаємо для розрахунку вартість одного метра квадратного рівною 20000 грн./ m^2 . На кожне робоче місце у середньому потрібно 8 m^2 . З урахуванням цього:

$$B_{y\partial} = 8 \cdot 8 \cdot 20000 = 1280000 \text{ грн.}$$

Вартість передавальних пристроїв складає 10% від вартості будівель, і у даному випадку вона складе: 128000 грн.

Балансова вартість інвентарю розраховується за нормою 3500 грн. на одне робоче місце. Тобто:

$$I_{nb} = R_{cn}^1 \cdot C_m, \quad (7.10)$$

де: C_m – ціна меблів для одного робочого місця, грн.

$$I_{nb} = 8 \cdot 3500 = 28000 \text{ грн.}$$

Балансова вартість обчислювальної техніки визначається по оптовим цінам постачальника з врахуванням витрат на транспортування.

Специфікація на обчислювальну техніку наведена в таблиці 7.7.

Дані по оптовій ціні на обладнання та комплектуючі вибирались за прайсом Інтернет магазину Компбест за 17.11.20 – джерело <https://compbest.com.ua>.

					ВКРМ-123.21.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

Таблиця 7.6 – Специфікація

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Персональний комп'ютер		10947
Системний блок		Acer X2632G DT
Процесор	Intel Core i3-4130 (2(4) ядра по 3.40 GHz)	-
Системна плата	X4660G VX4660G-I5840H1 4x USB 2.0, 2x USB 3.0, 5x Audio Ports, Com Port, 2x PS/2, LAN (RJ-45); DVI, HDMI	-
Відеокарта	AMD Radeon RX 550 4GB GDDR5 Red Dragon PowerColor (AXRX 550 4GBD5-DH)	-
Жорсткий диск	HDD Seagate Barracuda 750 Gb 7200 32Mb SATAII ST3750528AS (ST3750528AS) + 120 GB SSD	-
Оперативна пам'ять	DIMM 4096Mb DDR3 PC3-10600 CL9 Transcend JetRam, non-Reg., no-ECC , CL 9 (2 модулі)	-
DVD-привод	DVD -RW/+RW , LG SATA SuperMulti Bulk 22x, SecurDisc, black	-
Корпус	GRESSO GE-7525, 500W (120mm big fan), 2xIDE, full-ATX,БЖ 2xSATA, 1xFDD, Air Duct, 2xUSB 2.0, Mic+Audio, silver/black	-

Продовження таблиці 7.6

Найменування комплектуючої або обладнання	Тип	Оптова ціна
інше	Клавіатура, мишка	Подарунок
Монітор	22" TFT, ASUS VW223D (5ms, 300/3000:1, 170/160, D-SUB, Wide)	3600
Принтер лазерний	Canon i-SENSYS LBP6030W	2700
Принтер струминний	Epson Stylus Photo P50 (C11CA45341) + USB cable	5500
Копіювальний апарат	Canon i-SENSYS MF217W with Wi-Fi	5965

Витрати на транспорт, монтаж та випробування можуть бути прийняті в межах до 10% від оптової ціни.

Для визначення необхідної кількості капітальних вкладень складемо таблицю 7.8.

Таблиця 7.7 – Балансова вартість обчислювальної техніки

Найменування обчислювальної техніки	Кількість, шт.	Ціна за одиницю, грн.	Витрати на транспортування, монтаж та випробування.	Загальна вартість, грн.
Персональні комп'ютери	15	10947	16420,5	180625,5
Принтер лаз.	2	2700	540	5940
Принтер струм.	1	5500	550	6050
Сканери	1	968	96,8	1064,8
Копіюв. апарат	1	5965	596,5	6561,5
Всього	—	—	—	200241,8

Таблиця 7.8 – Вартість основних фондів та амортизаційні відрахування розробника

Групи та види основних фондів	Балансова вартість, грн.	Амортизація	
		Норма, %	Відрахування, грн.
1	2	3	4
Група 3			
1. Будівлі	1280000	-	-
2. Передавальні пристрої	128000	-	-
Всього по групі	1408000	5	70400
Група 4			
3. Обчислювальна техніка	200242	-	-
Всього по групі	200242	50	100121
Нематеріальні активи			
4. Нематеріальні активи	80000	10	8000
Група 5, 6			
5. Вимірювальні пристрої	9031	25	2257,75
6. Транспортні засоби	122500	20	24500
7. Господарський інвентар	28000	25	7000
Всього по групі	159531	-	20000
Разом	$K_p = 1847773$		$A_p = 198521$

Примітка: вартість автомобіля (Suzuki GSX-R 600, 2012г) взята по даним з автосалону автотрейдинг, вкладки автобазар, джерело <http://www.auto-trading.com.ua/sale/lot69649.html>, складає 4900 USD, що враховуючи курс 25 складає 122500 грн.

7.5 Визначення собівартості розробки та ціни програмної продукції

Визначимо основну зарплату виконавців:

$$Z_o = \frac{Z_{cd} \cdot T_{nz}}{N_e}, \quad (7.11)$$

де: N_e – кількість екземплярів програм, шт.

$$Z_o = 356 \cdot 209 / 80 = 930 \text{ грн.}$$

Визначимо додаткову зарплату (оплата відпусток, виконання державних та суспільних обов'язків) на рівні 10%:

$$Z_d = Z_o \cdot H_q \cdot 0,01, \quad (7.12)$$

де: H_q – норматив додаткової зарплати, %.

$$Z_d = 930 \cdot 10 \cdot 0,01 = 93 \text{ грн.}$$

Відрахування на соціальні потреби за нормативом $H_c = 22\%$ від суми основної та додаткової зарплати:

$$C_{oc} = 0,01 \cdot H_c (Z_o + Z_d), \quad (7.13)$$

де: H_c – відрахування на соціальні потреби, %.

$$C_{oc} = 0,01 \cdot 22(930+93) = 225 \text{ грн.}$$

Визначимо загальногосподарські витрати (електроенергію, ремонт і утримання приміщень і т.д) за нормативом $H_z = 15\%$ від основної зарплати:

$$G_{ocn} = Z_o \cdot H_z \cdot 0,01, \quad (7.14)$$

де: H_z – загальногосподарські витрати, %.

$$G_{ocn} = 930 \cdot 15 \cdot 0,01 = 140 \text{ грн.}$$

Визначимо витрати на матеріали для розробки програмної продукції за нормами споживання та діючими цінами за одиницю виміру:

$$Z_M = (Z_{M1} + Z_{M2} + Z_{M3}) / N_e, \quad (7.15)$$

де: Z_{M1} – вартість паперу, грн.;

Z_{M2} – вартість запам'ятовуючих пристроїв, грн.;

Z_{M3} – вартість фарби, картриджей, тонеру, грн.;

					ВКРМ-123.21.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

N_e – кількість екземплярів програм, шт.

Згідно виданих викладачем норм приймаємо одну пачку паперу на квартал. Тоді, враховуючи, що вартість пачки паперу складає $C_n = 105$ грн., визначаємо вартість паперу за період розробки $N_m = 3$ міс:

$$Z_{M1} = C_n \cdot N_m. \quad (7.16)$$

$$Z_{M1} = 105 \cdot 1 = 105 \text{ грн.}$$

Згідно виданих викладачем норм до вартості запам'ятовуючих пристроїв входить вартість CD дисків в кількості, що дорівнює кількості екземплярів програм та одного DVD диска для збереження резервної копії програми:

$$Z_{M2} = \sum C_d, \quad (7.17)$$

де: C_d – вартість дисків CD/DVD: CDR TDK 700Mb, 80Min, 52x Cake box – 2 грн./шт., DVD-R LG 4,7Gb, 16x speed Cake box – 2 грн./шт.

$$Z_{M2} = 80 \cdot 12 = 960 \text{ грн.}$$

Згідно виданих викладачем норм одноразовій заправці підлягають усі друкуючі пристрої і становить:

$$Z_{M3} = \sum C_z, \quad (7.18)$$

де: C_z – вартість розхідних матеріалів друкуючих пристроїв: відновлення та заправка картриджу для Canon i-SENSYS LBP6030W – 574 грн.; картридж для Epson Stylus Photo P50 – 558 грн.; відновлення картриджу для MF217W – 570 грн.

$$Z_{M3} = 574 + 558 + 570 = 1702 \text{ грн.}$$

$$Z_M = (105 + 960 + 1702) / 80 = 34 \text{ грн.}$$

Визначимо витрати на освоєння нових мов програмування або операційних систем за нормативом ($H_n = 15\%$) від основної зарплати виконавців:

$$O_n = Z_o \cdot H_n \cdot 0,01, \quad (7.19)$$

де: H_n – норматив витрат на освоєння нових мов програмування, %.

$$O_n = 930 \cdot 15 \cdot 0,01 = 140 \text{ грн.}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ($N_e = 80$ прим.):

					ВКРМ-123.21.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

$$A_m = \frac{A_p \cdot N_{mic}}{N_e \cdot 12}, \quad (7.20)$$

де: A_p – загальна річна сума амортизаційних відрахувань, грн.

$$A_m = 198521 \cdot 3 / (80 \cdot 12) = 620 \text{ грн.}$$

Величини ціна підприємства, податок на додану вартість, відпускна ціна програмної продукції визначаються за формулами, приведеними в таблиці 7.9

Таблиця 7.9 – Нормативна калькуляція собівартості розробки програмного забезпечення задачі

Найменування статей витрат	Позначення	Величина, грн.
1. Основна зарплата виконавців	Z_o	930
2. Додаткова зарплата виконавців	Z_d	93
3. Відрахування на соціальні потреби	C_{oc}	225
4. Загальногосподарські витрати	G_{ocn}	140
5. Витрати на матеріали	Z_M	34
6. Освоєння нових операційних систем, мов програмування	O_n	140
7. Амортизація основних фондів	A_m	620
8. Повна собівартість програмного забезпечення	C_n	2182
9. Плановий прибуток	P_p	1091
10. Ціна підприємства $C_n = C_n + P_p$	C_n	3273
11. Податок на додану вартість $ПДВ = 0.01 \cdot H_{об} \cdot C_n$	$ПДВ$	654,6
12. Відпускна ціна програмної продукції $C = C_n + ПДВ$	C	3927,6

Повна собівартість ПЗ визначається як сума витрат за попередніми статтями калькуляції:

$$C_n = Z_o + Z_d + C_{oc} + \Gamma_{ocn} + Z_m + O_n + A_m. \quad (7.21)$$

$$C_n = 930 + 93 + 225 + 140 + 34 + 140 + 630 = 2182 \text{ грн.}$$

Визначимо плановий прибуток за рівнем рентабельності (P_n) програмної продукції, яка залежить від складності програми та ступеня новизни задачі.

Для даного програмного забезпечення рівень рентабельності складає 50%.

$$P_p = 0,01 \cdot P_n \cdot C_n, \quad (7.22)$$

де: P_n – рівень рентабельності, %.

$$P_p = 0,01 \cdot 50 \cdot 2182 = 1091 \text{ грн.}$$

7.6 Визначення об'єму капітальних вкладень у споживача програмної продукції

Об'єм капітальних вкладень у споживача програмної продукції визначаємо на основі балансової вартості основних фондів, яка враховує ціну, транспортно-заготівельні витрати, вартість будівель, монтажних та пусконаладжувальних робіт, а також витрати на випробування у виробничих умовах. Результати розрахунків зводимо у таблицю 7.9.

Таблиця 7.10 – Розрахунок об'єму капітальних вкладень у споживача програмної продукції

Найменування капітальних вкладень	Сума за варіантами, грн.	
	Базовий	Новий
Вартість програмної продукції	–	3928
Всього капітальних витрат	–	3928

7.7 Визначення експлуатаційних витрат

Експлуатаційні витрати у споживача програмної продукції визначаємо при умові роботи підсистеми на протязі року. Результати зводимо до таблиці 7.11.

Таблиця 7.11 – Розрахунок експлуатаційних витрат у споживача програмної продукції

Найменування статей витрат	Позначення	Сума витрат за варіантами, грн.	
		Базовий	Новий
1. Витрати на відновлення працездатності системи після збоїв та профілактичні роботи	Z_p	21901	4380
2. Витрати на амортизацію	$Z_{ам}$	0	982
Всього витрат за рік	I	21901	5362

Витрати на відновлення працездатності системи після збоїв та профілактичні роботи:

$$Z_p = T_p \cdot Z_2 \cdot (1 + 0,01 \cdot H_q) \cdot (1 + 0,01 \cdot H_c), \quad (7.23)$$

де: T_p – кількість годин обслуговування сегменту мережі за рік, год.;

Z_2 – заробітна плата обслуговуючого персоналу, грн/год.

Після купівлі нового програмного забезпечення кількість профілактичних годин робіт зменшилася з 240 годин на рік до 48 годин на рік, тому витрати на технічне обслуговування зменшилися з:

$$Z_{p \text{ баз}} = 240 \cdot 68 \cdot 1,1 \cdot 1,22 = 21901 \text{ грн},$$

до:

$$Z_{p \text{ нов}} = 48 \cdot 68 \cdot 1,1 \cdot 1,22 = 4380 \text{ грн}.$$

Витрати по амортизації визначаються на основі норм амортизаційних відрахувань, вартості програмної продукції і основних фондів. Для розрахунку складаємо таблицю 7.12.

Визначимо величину економічного ефекту у користувача програмної продукції за формулою:

$$E_{cn} = (I_{\delta} - I_n) - E_n(K_n - K_{\delta}), \quad (7.26)$$

де: I_{δ} , I_n – величина експлуатаційних витрат за базовим и новим варіантом відповідно;

K_{δ} , K_n – об'єм капітальних вкладень за варіантами, що порівнюються.

$$E_{cn} = (21901 - 5362) - 0,25 \cdot 3928 = 15557 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у споживача програмної продукції за рахунок зниження експлуатаційних витрат:

$$T_{cn} = \frac{K_n - K_{\delta}}{I_{\delta} - I_n}, \quad (7.27)$$

$$T_{cn} = \frac{3928}{21901 - 5362} = 0,2 \text{ року.}$$

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

Таблиця 7.13 – Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величина
1	2	3
1. Кількість екземплярів програми	Прим.	80
2. Повна собівартість розробленої програми	Грн.	2182
3. Ціна розробленої програми	Грн.	3273
4. Плановий прибуток від реалізації розробленої програми	Грн.	1091
5. Рентабельність програмної продукції	%	50
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	1847773
7. Загальний прибуток від реалізації програмної продукції	Грн.	87280

Продовження таблиці 7.13

1	2	3
8. Величина економічного ефекту при виготовлені програмної продукції	Грн.	44222
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Років	1,3
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	3928
11. Величина економічного ефекту у користувача програмної продукції	Грн.	15557
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Років	0,2

7.9 Висновки

Розроблена програма економічно вигідна. За рахунок впровадження програмного забезпечення досягається скорочення часу обробки інформації, підвищується культура праці, підвищення якості приймаючих управлінських рішень.

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

Загальна комп'ютеризація суспільства призвела до того, що використання комп'ютерів стало повсюдним у всіх сферах економіки та народного господарства. Застосування ЕОМ дозволило значно підвищити продуктивність праці, змінити характер і зміст праці. Комп'ютеризація, поряд з незаперечними перевагами, тягне за собою і багато проблем. Для того, щоб активне застосування комп'ютерних технологій не стало додатковим чинником погіршення здоров'я, вкрай необхідно щоб робоче місце відповідало гігієнічним вимогам. Темою дипломного проекту є розробка та дослідження та реалізація програмного продукту, тому актуально буде розгляд умов праці програміста.

8.1 Аналіз умов праці на робочому місці програміста

Робота програміста пов'язана з постійною роботою на ЕОМ, яка відбувається у кімнаті розмірами 4,6м×7,1м×2,9 м. Одна з її більших стін має шість двостулкових вікон, розмірами 2,2м×1,8 м, які виходять на північний схід. Вікна розташовані рівномірно по всій довжині стіни. Підлога в кімнаті покрита леноліумом, всі стіни пофарбовані світло оранжевого кольору до висоти 2,8м, а далі підвісна стеля. Уздовж стін розташовані комп'ютерні столи. На них розташовуються 2 персональні комп'ютери й інша оргтехніка (сканер принтери, телефони й ксерокс). Столи мають пластикове покриття. Габарити їхньої робочої поверхні 1350 мм×900 мм. Висота столів 740 мм. Висота стільців від рівня підлоги становить 440 мм.

Згідно НПАОП 0.00 – 1.28 – 10 «Правила охорони праці під час електронно-обчислювальних машин» площа повинна задовольняти умові - не менш 6 м² на одне робоче місце. А згідно ДСН 3.3.6.042-99 “Санітарні норми мікроклімату виробничих приміщень” цей показник може бути скорегованим.

					ВКРМ-123.21.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

Кратність повітрообміну в приміщенні вузла також регламентується ДСанПіН 3.3.2.007 — 98, вона повинна становити 20 м³/годину на одне місце. Виконання даних вимог забезпечить підтримку в приміщенні вузла оптимального значення вологості й складу повітря.

Відповідно ДБН В.2.5 – 28 – 2006 роботу програміста можна віднести до роботи з малою точністю (найменший розмір об'єкта розрізнення від 1 до 5 мм) V-го розряду зорової роботи, з великою контрастністю об'єкта розрізнення (символів на екрані дисплея), з темним тлом (під розряд зорової роботи В). Приміщення вузла можна віднести до 1-ої групи приміщень, у яких проводиться розрізнення об'єктів зорової роботи при фіксованому напрямку лінії зору того, що працює на робочу поверхню. Для такого типу приміщень і розряду зорової роботи нормоване значення коефіцієнта природної освітленості (КПО) робочої поверхні (при сполученому висвітленні), повинен становити 0,5%, освітленість при штучному висвітленні повинна становити 300 лк.

За результатами виміру освітленості відділом охорони праці величина освітленості від системи загального штучного висвітлення лежить у межах 200-250 лк, що не відповідає вимогам, які пред'являються до приміщення.

Відповідно ДСанПіН 3.3.2.007 - 98 рівні звукового тиску в робочому приміщенні не повинні перевищувати в октавних смугах із середньо геометричними частотами наступних значень, наведених у таблиці 8.1.

У приміщенні перебувають наступні джерела шуму: електродвигуни внутрішнього вентилятора ЕОМ; працюючі принтери; працюючі дисководи. Шум, вироблений вентилятором можна класифікувати як постійний, всі інші джерела шуму, як імпульсні. Відповідно паспорта на приміщення рівень звуку, Дб(А), обмірюваний за шкалою (А) шумоміра досяг величини 28,3 Дб(А) при роботі всього встаткування вузла, включаючи й ксерокс. Це дозволяє зробити висновок про відповідність рівня звуку в приміщенні вимогам нормативних актів.

					ВКРМ-123.21.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

Таблиця 8.1 – Допустимі спектри рівнів звукового тиску

Робоче місце	Рівень звукового тиску, дБ, в октавних смугах із середньгеометричними частотами, Гц								Рівень звуку і еквівалентний рівень звуку, дБА
	63	125	250	500	1000	2000	4000	8000	
Приміщення конструкторських бюро, програмістів обчислювальних машин, лабораторій для теоретичних робіт і опрацювання експериментальних даних, прийому хворих в медпунктах	71	61	54	49	45	42	40	38	50

Ергономічні вимоги до робочого місця працюючого з ВДТ ЕОМ і ПЕОМ нормують Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин: ДСанПІН 3.3.2-007-98.

Оптимальне положення тіла того, що працює забезпечується відповідною конструкцією робочого місця, а також регуляцією висоти робочої поверхні, сидіння, простори й підставки для ніг. Даного місця програміста не мають регульованих параметрів. Відмінності реальних параметрів робочого місця від параметрів відповідні вимоги нормативного акту дані в таблиці 8.2.

Таблиця 8.2 – Відмінності реальних параметрів робочого місця від параметрів відповідні вимоги нормативного акту

Ріст людини, см	Висота робочої поверхні мм,	Висота простору для ніг, мм	Висота робочого сидіння, мм
175	765(740)	655(600)	450(440)

У дужках зазначені реальні значення параметрів робочого місця; всі вони не відповідають параметрам, зазначеним у стандарті.

Параметри мікроклімату можуть мінятися в широких межах, тоді як необхідною умовою життєдіяльності людини є підтримка сталості температури тіла завдяки властивості терморегуляції, тобто здатності організму регулювати віддачу тепла в навколишнє середовище.

У приміщеннях, де встановлені комп'ютери, повинні дотримуватися певні параметри мікроклімату. У санітарних нормах ДСН 3.3.6.042 - 99 встановлені величини параметрів мікроклімату, що створюють комфортні умови. Ці норми встановлюються в залежності від пори року, характеру трудового процесу і характеру виробничого приміщення (див. табл. 8.3).

Таблиця 8.3 – Параметри мікроклімату для приміщень, де встановлені комп'ютери

Період року	Параметр мікроклімату	Величина
Холодний	Температура повітря в приміщенні	22 - 24°C
	Відносна вологість	40 - 60%
	Швидкість руху повітря	до 0,1 м/с
Теплий	Температура повітря в приміщенні	23 - 25°C
	Відносна вологість	40 ... 60%
	Швидкість руху повітря	0,1 ... 0,2 м / с

8.2 Розробка заходів з умов поліпшення охорони праці

Перерахуємо проведені заходи щодо забезпечення умов праці на робочому місці програміста.

З точки зору забезпечення електробезпеки до цих заходів можна віднести: устаткування розподільних щитів спеціальними розетками з заземлюючими контактами; організація заземлення всіх приладів і пристроїв; періодична перевірка всіх приладів і пристроїв; щорічна здача іспитів з охорони праці.

З точки зору забезпечення оптимальних умов мікроклімату, рівня звуку і освітленості до цих заходів можна віднести: організацію природної вентиляції, за допомогою дефлектора, для забезпечення необхідного повітрообміну в приміщенні вузла; організацію системи центрального опалювання, для підтримки оптимальної температури в холодний період року; організацію штучного загального освітлення, для забезпечення необхідних умов зорової роботи, що відповідають, оформлення паспорта на приміщення вузла, з занесенням в нього вимірювань освітленості і рівня звуку, проведених відділом охорони праці.

З точки зору забезпечення пожежної безпеки до цих заходів можна віднести наявність схеми евакуації з приміщення вузла, у випадку пожежі, повішену на вхідні двері.

Аналіз умов праці на робочому місці інженера-програміста показав, що на робочому місці не виконуються вимоги ергономіки. Для виконання їх можна запропонувати заміну не регульованого сидіння на крісло з регульованими ергономічними параметрами, а також заміну використовуваного столу на робоче місце оператора ЕОМ.

Різними фірмами в сукупності розроблено понад 11 схем регулювань параметрів робочого крісла, які забезпечують: плавне переміщення сидіння по висоті за допомогою газової пружини; плавна зміна нахилу спинки і сидіння; регулювання пружинного протivotиску спинки крісла на спину оператора; перестановку спинки по висоті; зміна глибини сидіння шляхом зміни вигину краю сидіння; синхронне повторення рухів оператора сидінням і спинкою в

					ВКРМ-123.21.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

правильному кутовому співвідношенні; синхронне повторення спинкою крісла рухів верхньої частини тулуба того, що сидить; амортизацію сидіння.

Найбільш популярними моделями комп'ютерних крісел, які володіють чотирма основними регулюваннями (висоти сидіння, висоти, глибини і нахилу спинки), є італійські, фінські моделі «Toronto», «Bini», «Metro», «NewStar», «Capris», «Fenix», «Xenus», «Quintus», «Senior», «Volos», «Ergo» і т.д. Подібні крісла, відрегульовані відповідно до зростання і ваги оператора, а також характеру виконуваної роботи, дозволяють понизити навантаження на опорно-руховий апарат людини, що працює за комп'ютером.

8.3 Дослідження інформаційного навантаження на програміста

Програміст, у залежності від підготовки і досвіду, вирішує задачі різної складності, але в загальному випадку робота програміста будується по наступному алгоритму:

Таблиця 8.4 – Алгоритм роботи програміста

Етап	Зміст	Витрата часу, %
III	Постановка задачі Вивчення матеріалу за поставленою задачею	6.25
III	Визначення методу рішення задачі	6.25
IV	Складання алгоритму рішення задачі	12.5
V	Програмування	25
VI	Налагодження програми, складання звіту	50

Даний алгоритм відображає загальні дії програміста при рішенні поставленої задачі незалежно від її складності.

Таблиця 8.5 – Загальні дії програміста при рішенні поставленої задачі незалежно від її складності

Етап	Член алгоритм	Зміст роботи	Літерне позначенн
1	2	3	4
I	1	Одержання першого варіанта технічного	A
	7	Складання і уточнення технічного завдання	Ri
		Одержання остаточного варіанта технічного завдання	Сіл
	4	Складання переліку матеріалів, що існують	НП2
	5	Вивчення матеріалів по тематиці задачі	Ag
	6	Вибір методу рішення	C2J3
	7	Уточнення й узгодження обраного методу	B2
	8	Остаточний вибір методу рішення	СчТ4
	9	Аналіз вхідної і вихідної інформації	B2
	10	Вибір мови програмування	C4TS
	11	Визначення структури програми	H3C5q
	12	Складання блок-схеми програми	C6q2
	13	Логічний аналіз програми і коректування її	F1H4W2
	14	Компіляція програми	F2
	15	Виправлення помилок	D1W2
	16	Редагування програми	F2H5B3W
	17	Виконання програми	F3
	18	Аналіз результатів виконання	H6W5
	19	Тестування	C7W6
	20	Підготовка звіту про роботу	F4

Підрахуємо кількість членів алгоритму і їхню частоту (імовірність) щодо загального числа, прийнятого за одиницю. Імовірність повторення і-ої ситуації визначається по формулі:

$$P_i = k/p, \quad (8.1)$$

де k - кількість повторень кожного елемента одного типу, p - сумарна кількість

повторень від джерела інформації, одного типу.

Результати розрахунку зведемо в таблицю:

Таблиця 8.6 – Ймовірність повторення і-тої ситуації

Джерело	Члени алгоритму	Символ	Кількість	Частота повторень
1	Аферентні - усього		6	1,00
	Вивчення технічної	А	2	0,33
	Спостереження	Р	4	0,67
2	Еферентні - усього		18	1,00
	Уточнення	В	3	0,17
	Вибір найкращого	С	8	0,44
	Виправлення	0	1	0,06
	Аналіз отриманих	н	6	0,33
	Виконання	к	0	0
3	Логічні умови		13	1,00
	Прийняття рішень на основі вивчення	І	5	0,39
	Грабічні матеріали	Ч	2	0,15
	Отриманого тексту	У	6	0,46
	Усього:		37	

Кількісні характеристики алгоритму (Табл.8.6) дозволяють розрахувати інформаційне навантаження програміста. Ентропія інформації елементів кожного джерела інформації розраховується по формулі, біт/сигн:

$$H_j = \sum_{i=1}^m p_i \log_2 p_i \quad (8.2)$$

де m - число однотипних членів алгоритму розглянутого джерела інформації.

$$H_1 = 2 \times 2 + 2 \times 4 = 12$$

$$H_2 = 3 \times 1,585 + 8 \times 3 + 0 + 6 \times 2,585 = 44,265$$

$$H_3 = 5 \times 2,323 + 2 \times 1 + 6 + 2,585 = 29,125$$

Потім визначається загальна ентропія інформації, біт/сигн:

$$H_s = H_1 + H_2 + H_3, \quad (8.3)$$

де H_1 , H_2 , H_3 - ентропія аферентних, еферентних елементів і логічних умов відповідно.

$H_s = 10 + 44,265 + 29,125 = 83,39$. Далі визначається потік інформаційного навантаження біт/хв,

$$F = \frac{H_{\Sigma} \cdot N}{t}, \quad (8.4)$$

де N - сумарне число всіх членів алгоритму; t - тривалість виконання всієї роботи, хв.

Від кожного джерела в інформації (члена алгоритму) у середньому надходить 3 інформаційних сигнали в годину, час роботи - 225 годин,

$$\Phi = \frac{83,39 \cdot 3 \cdot 225}{13500} = 2,6 \text{ біт/с}$$

Розраховане інформаційне навантаження порівнюється з припустимою. При необхідності приймається рішення про зміни в трудовому процесі.

Умови нормальної роботи виконуються при дотриманні співвідношення:

$$\Phi_{\text{доп.мін}} < \Phi_{\text{расч}} < \Phi_{\text{доп.макс}}, \quad (8.5)$$

де $\Phi_{\text{доп.мін}}$, і $\Phi_{\text{доп.макс}}$ мінімальний і максимальний припустимі рівні інформаційних навантажень (0,8 і 3,2 біт/з відповідно);

$\Phi_{\text{расч}}$ - розрахункове інформаційне навантаження $0,8 < 2,6 < 3,2$.

8.3 Висновки до розділу

У цій частині дипломної проекту були розглянуті вплив факторів трудового і виробничого середовища програмістів, дослідження інформаційного навантаження на програміста. Дотримання умов, що визначають оптимальну організацію робочого місця програміста і навантаження, отримані ним в процесі роботи, дозволить зберегти гарну працездатність протягом усього робочого дня, підвищить, як у кількісному, так і в якісному відносінах продуктивність праці програміста.

					ВКРМ-123.21.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

9 ОСНОВНІ ВИСНОВКИ

Розроблене програмного забезпечення системи аудиту мережі теплопостачання з використанням технології ІОТ призначене для перевірки систем мереж теплопостачання.

Воно має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань з його сторони.

Система легко інтегрується в сучасні платформи і взаємодіє з усіма необхідними компонентами. Підтримується на платформі Windows останньої версії.

Актуальність таких програм є дуже високою адже контроль за системами теплопостачання є надзвичайно важливим. Він дозволяє ефективно налаштувати роботу обладнання.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

У випускній кваліфікаційній роботі наведені теоретичне узагальнення й рішення наукового завдання дослідження методів аудиту мережі теплопостачання з використанням технології ІоТ.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем аудиту мереж теплопостачання.
- Розроблена системи аудиту мережі теплопостачання з використанням технології ІоТ.
- Проведене випробовування розробленої системи аудиту мережі теплопостачання з використанням технології ІоТ.

При створенні програмного забезпечення було використано об'єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційного програмного забезпечення.

					ВКРМ-123.21.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

Програма реалізована на мові високого рівня С# з використанням середовищ розробки Microsoft Visual Studio та Unity. Програма призначена для виконання під управлінням багатозадачної операційної системи Windows 7/8/10.

Для підвищення рівня безпеки розробленого програмного забезпечення запропоновано застосовувати два рівні захисту програми, верхній рівень – вікно введення пароля на вхід в програмне забезпечення, другий рівень – обов'язкова наявність сертифікатів та приватних ключів для доступу на об'єкти.

Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення та масштабування.

Розроблене програмне забезпечення має реальний економічний ефект від його впровадження у виробництво у сумі 15557 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 0,2 роки.

					ВКРМ-123.21.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		83

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Хокинг Дж. Unity в действии. Мультиплатформенная разработка на C#. - М.: Питер, 2018. - 608 с.
2. Торн А. Основы анимации в Unity // издательство ДМК-Пресс, 2016. – 176 с.
3. Агуров, Павел С#. Сборник рецептов / Павел Агуров. - М.: "БХВ-Петербург", 2012. - 432 с.
4. Албахари, Джозеф С# 3.0. Справочник / Джозеф Албахари, Бен Албахари. - М.: БХВ-Петербург, 2012. - 944 с.
5. Албахари, Джозеф С# 3.0. Справочник / Джозеф Албахари, Бен Албахари. - М.: БХВ-Петербург, 2013. - 944 с.
6. Альфред, В. Ахо Компиляторы. Принципы, технологии и инструментарий / Альфред В. Ахо и др. - М.: Вильямс, 2015. - 266 с.
7. Бишоп, Дж. С# в кратком изложении / Дж. Бишоп, Н. Хорспул. - М.: Бином. Лаборатория знаний, 2013. - 472 с.
8. Вагнер, Билл С# Эффективное программирование / Билл Вагнер. - М.: ЛОРИ, 2013. - 320 с.
9. Зиборов, В.В. Visual C# 2012 на примерах / В.В. Зиборов. - М.: БХВ-Петербург, 2013. - 480 с.
10. Зиборов, Виктор Visual C# 2010 на примерах / Виктор Зиборов. - М.: "БХВ-Петербург", 2011. - 432 с.
11. Ишкова, Э. А. Самоучитель С#. Начала программирования / Э.А. Ишкова. - М.: Наука и техника, 2013. - 496 с.
12. Касаткин, А. И. Профессиональное программирование на языке си. Управление ресурсами / А.И. Касаткин. - М.: Высшая школа, 2012. - 432 с.
13. Лотка, Рокфорд С# и CSLA .NET Framework. Разработка бизнес-объектов / Рокфорд Лотка. - М.: Вильямс, 2010. - 816 с.

					ВКРМ-123.21.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		84

14. Мак-Дональд, Мэтью Silverlight 5 с примерами на C# для профессионалов / Мэтью Мак-Дональд. - М.: Вильямс, 2013. - 848 с.
15. Марченко, А. Л. Основы программирования на C# 2.0 / А.Л. Марченко. - М.: Интернет-университет информационных технологий, Бином. Лаборатория знаний, 2011. - 552 с.
16. Подбельский, В. В. Язык C#. Базовый курс / В.В. Подбельский. - М.: Финансы и статистика, Инфра-М, 2011. - 384 с.
17. Прайс, Джейсон Visual C# 2.0. Полное руководство / Джейсон Прайс , Майк Гандэрлой. - М.: Век +, Корона-Век, Энтроп, 2010. - 736 с.
18. Рихтер, Джеффри CLR via C#. Программирование на платформе Microsoft .NET Framework 4.0 на языке C# – М.: Питер, 2013. - 928 с.
19. Смоленцев, Н. К. MATLAB. Программирование на Visual C#, Borland JBuilder, VBA (+ CD-ROM) / Н.К. Смоленцев. - М.: ДМК Пресс, 2011. - 456 с.
20. Троелсен, Эндрю Язык программирования C# 5.0 и платформа .NET 4.5 / Эндрю Троелсен. - М.: Вильямс, 2015. - 486 с.
21. Троелсен, Эндрю Язык программирования C# 2008 и платформа .NET 3.5 / Эндрю Троелсен. - М.: Вильямс, 2010. - 370 с.
22. Фримен, Адам ASP.NET MVC 3 Framework с примерами на C# для профессионалов / Адам Фримен , Стивен Сандерсон. - М.: Вильямс, 2011. - 672 с.
23. Альфред, В. Ахо Структуры данных и алгоритмы / Альфред В. Ахо, Джон Э. Хопкрофт, Джеффри Д. Ульман. - М.: Вильямс, 2016. - 400 с.
24. Бёрд, Ричард Жемчужины проектирования алгоритмов. Функциональный подход / Ричард Бёрд. - М.: ДМК Пресс, 2015. - 330 с.
25. Вигерс, Карл Разработка требований к программному обеспечению / Карл Вигерс , Джой Битти. - М.: Русская Редакция, БХВ-Петербург, 2016. - 736 с.
26. Головешкин, В. А. Теория рекурсии для программистов / В.А. Головешкин, М.В. Ульянов. - М.: ФИЗМАТЛИТ, 2006. - 296 с.
27. Зелковиц, М. Принципы разработки программного обеспечения / М. Зелковиц, А. Шоу, Дж. Гэннон. - М.: Мир, 1982. - 364 с.

					ВКРМ-123.21.0008.00.00.ПЗ	Арк.
Вым.	Арк.	№ докум.	Підпис	Дата		85

28. Рассел Дж. Интерфейс программирования приложений / Джесси Рассел. - М.: VSD, 2012. - 811 с.

29. Колдаев В.Д. Основы алгоритмизации и программирования: Учебное пособие / В.Д. Колдаев. - М.: Форум, 2015. - 352 с.

30. Макарова, Н.В. Основы программирования. учебник с практикумом - М.: КноРус, 2016. - 112 с.

31. Окулов С.М. Основы программирования / С.М. Окулов. - М.: Бином, 2015. - 336 с.

32. Кью Дж. Объектно-ориентированное программирование / Дж. Кью, М. Джеанини. - М.: Питер, 2015. - 240 с.

33. Хорев, П.Б. Объектно-ориентированное программирование: Учебное пособие / П.Б. Хорев. - М.: Академия, 2018. - 384 с.

Лесневский, А. С. Объектно-ориентированное программирование для начинающих / А.С. Лесневский. - М.: Бином. Лаборатория знаний, 2005. - 232 с

34. Вайсфельд М. Объектно-ориентированное мышление / М. Вайсфельд. - М.: Питер, 2014. - 998 с.

35. Могилев А. Методы программирования. Компьютерные вычисления / А. Могилев, Л. Листрова. - М.: БХВ-Петербург, 2008. - 320 с.

36. Клейнберг, Дж. Алгоритмы. Разработка и применение / Дж. Клейнберг, Е. Тардос. - М.: Питер, 2016. - 800 с.

37. Лафоре, Роберт Структуры данных и алгоритмы в Java / Роберт Лафоре. - М.: Питер, 2016. - 704 с.

38. Магда, Юрий Аппаратное обеспечение и эффективное программирование / Юрий Магда. - М.: Питер, 2007. - 352 с.

39. Макконнелл, Дж. Анализ алгоритмов. Активный обучающий подход / Дж. Макконнелл. - М.: Техносфера, 2009. - 416 с.

40. Робертсон, Л. А. Программирование - это просто. Пошаговый подход / Л.А. Робертсон. - М.: Бином. Лаборатория знаний, 2010. - 384 с.

					ВКРМ-123.21.0008.00.00.ПЗ	Арк.
Вым.	Арк.	№ докум.	Підпис	Дата		86

41. Скиена, Стивен Алгоритмы. Руководство по разработке / Стивен Скиена. - М.: БХВ-Петербург, 2011. - 720 с.

42. Струченков, В. И. Методы оптимизации в прикладных задачах / В.И. Струченков. - М.: Солон-Пресс, 2012. - 167 с.

43. Хант, Э. Программист-прагматик. Путь от подмастерья к мастеру / Э. Хант, Д. Томас. - М.: ЛОРИ, 2013. - 288 с.

44. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. 4-е изд. – СПб.: Питер, 2010. – 944 с.

45. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин: ДСанПІН 3.3.2-007-98 – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/v0007282-98>

46. Закон України «Про охорону праці» від 14.10.1992 р. № 2694-ХІІ. - Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/2694-12>

47. Зеркалов Д. В. Охорона праці в Галузі: Загальні вимоги: навч. посіб. Київ: Основа. 2011. 551 с.

48. Наказ Міністерства соціальної політики України 14.02.2018 № 207 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями». - Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/z0508>

49. Наказ Міністерства будівництва, архітектури та житлово-комунального господарства України № 168 від 15.05.2006 Про затвердження ДБН В.2.5-28-2006 "Природне і штучне освітлення" <https://zakon.rada.gov.ua/rada/show/v0168667-06>

50. Постанова № 42 від 01.12.1999 Головного державного санітарного лікаря України «Санітарні норми мікроклімату виробничих приміщень ДСН 3.3.6.042-99. - Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/va042282-99>

					ВКРМ-123.21.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		87

Додаток А
(обов'язковий)

Технічне завдання

Зміст

1	Найменування та область застосування.....	2
2	Підстава для розробки.....	2
3	Мета та призначення розробки.....	2
4	Джерела розробки.....	2
5	Технічні вимоги.....	2
5.1	Вміст проекту.....	2
5.2	Показники призначення.....	3
5.3	Вимоги до функціональних характеристик.....	3
5.4	Вимоги до архітектури.....	3
5.5	Вимоги до надійності.....	3
5.6	Умови експлуатації.....	4
5.7	Вимоги до складу та параметрів технічних засобів.....	4
5.8	Вимоги до інформаційної і програмної сумісності.....	4
5.8.1	Обладнання.....	4
5.8.2	Мова програмування.....	4
5.8.3	Вхідні дані.....	5
5.8.4	Вихідні дані.....	5
6	Вимоги до програмної документації.....	5
7	Економічні вимоги.....	5
8	Вимоги щодо охорони праці.....	5
9	Перелік документів, що розробляються.....	6
10	Етапи розробки.....	6
11	Порядок контролю та приймання.....	6

					ВКРМ-123.21.0008.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Кіресв С.М.				Дослідження та програмна реалізація системи аудиту мережі теплопостачання з використанням технології IoT	Літ.	Аркуш	Аркушів
Перевірів	Мелешко Є.В.					М	1	6
Н. Контр.	Гермак В.С.				ЦНТУ КІ-20М			
Затв.	Смірнов О.А.							

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи аудиту мережі тепlopостачання з використанням технології IoT.

2 Підстава для розробки

Підставою для розробки служить завдання на кваліфікаційну випускную роботу, видане на кафедрі програмування та захисту інформації (нак. №42-13 від 02.08.2021 року).

3 Мета та призначення розробки

Метою магістерської роботи є дослідження та програмна реалізація системи аудиту мережі тепlopостачання з використанням технології IoT.

4 Джерела розробки

Джерелом цієї магістерської роботи є стосовна до теми література і існуючі аналоги.

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

- вибір і обґрунтування методів реалізації проекту;
- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;

					ВКРМ-123.21.0008.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

- техніко-економічне обґрунтування доцільності прийнятого до розробки програмного забезпечення;
- аналіз умов праці;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

- системи аудиту мережі теплопостачання з використанням технології IoT;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення не повинно мати обмежень на версію драйверів та операційної системи.

5.4 Вимоги до архітектури

Компонент, що розробляється повинен використовувати системні засоби та апаратні засоби, що на даному етапі розвитку обчислювальної техніки найбільше поширені.

5.5 Вимоги до надійності

Програмні модулі написані по всім правилам, які стосуються стандартних викликів процедур, функцій, методів і форм, визначених технічною документацією на середовище розробки.

					ВКРМ-123.21.0008.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

- температура повітря: 19-20 град. по Цельсію;
- відносна вологість повітря до 80%;
- атмосферний тиск 107 кПа.

5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на ПЕОМ архітектури IBM PC, працювати в ОС Windows 7/8/10 і з сумісними з цією платформою пристроями і прикладним програмним забезпеченням.

5.8 Вимоги до інформаційної і програмної сумісності

Переносність програмного забезпечення повинна бути забезпечена за рахунок його реалізації стандартного інтерфейсу взаємодії з ОС, що працюють під управлінням ОС Windows 7/8/10.

5.8.1 Обладнання

Комп'ютер Intel® Celeron/8 Mb/1.2 Gb/SVGA 14" 1Mb або сумісні з ним.

5.8.2 Мова програмування

Мова програмування C#, середовище розробки Microsoft Visual Studio та Unity.

					ВКРМ-123.21.0008.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

Програмна продукція повинна бути представлена у виді опису структури даних, схем та опису алгоритму, а також текстів вихідних модулів програмного забезпечення згідно ЄСПД .

7 Економічні вимоги

7.1 Для ПЗ необхідно виробити функціонально-вартісний аналіз варіантів розробки.

7.2 Виконати розрахунок витрат показників економічного ефекту з урахуванням цін на 3 вересня 2021 року.

8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи повинні бути розглянуті умови праці розробників при розробці та підтримці запропонованого програмного забезпечення.

					ВКРМ-123.21.0008.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

9 Перелік документів, що розробляються

- Наукова новизна – 1 аркуш.
- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Показники економічної ефективності – 1 аркуш.
- Пояснювальна записка – 87 аркушів.

10 Етапи розробки

10.1 Збір і обробка інформації по темі магістерської роботи. Постановка задачі на виконання магістерської роботи (складання ТЗ).

10.2 Проведення досліджень або експериментальних робіт для уточнення основних положень магістерської роботи.

10.3 Розробка функціональних схем, блок схем алгоритмів роботи програмного забезпечення.

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Розрахунок з техніко-економічного обґрунтування.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

11.1 Подання магістерської роботи на попередній захист 10.12.2021 р.

11.2 Подання магістерської роботи на захист .12.2021 р.

					ВКРМ-123.21.0008.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ
Керівник випускної кваліфікаційної роботи
за другим (магістерським) рівнем вищої освіти
_____ Є.В. Мелешко

*Дослідження та програмна реалізація системи аудиту мережі
теплопостачання з використанням технології IoT*

Лістинг програми

Код документу 12

Носій: DVD-диск

Загальна кількість аркушів: 53

Літера: РП

Кропивницький – 2021 року

Основна програма

```

using System.Reflection;
using System.Runtime.InteropServices;

[assembly: AssemblyTitle("Application name")]
[assembly: AssemblyDescription("")]
[assembly: AssemblyConfiguration("")]
[assembly: AssemblyCompany("")]
[assembly: AssemblyProduct("Auditor")]
[assembly: AssemblyCopyright("Copyright © 2021")]
[assembly: AssemblyTrademark("")]
[assembly: AssemblyCulture("")]

// Setting ComVisible to false makes the types in this assembly not visible
// to COM components.  If you need to access a type in this assembly from
// COM, set the ComVisible attribute to true on that type.
[assembly: ComVisible(false)]

// The following GUID is for the ID of the typelib if this project is exposed to
// COM
[assembly: Guid("546c40b4-d139-4595-badf-50af8fed9d43")]

/
[assembly: AssemblyVersion("1.0.0.0")]
[assembly: AssemblyFileVersion("1.0.0.0")]
<%@ Page Title="Change Password" Language="C#" MasterPageFile="~/Site.master"
AutoEventWireup="true"
CodeBehind="ChangePassword.aspx.cs" Inherits="RoleManager.ChangePassword" %>

<asp:Content ID="HeaderContent" runat="server"
ContentPlaceHolderID="HeadContent">
</asp:Content>
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">
<asp:ChangePassword ID="ChangeUserPassword" runat="server"
CancelDestinationPageUrl="~//"
EnableViewState="false" RenderOuterTable="false"
SuccessPageUrl="ChangePasswordSuccess.aspx">
<ChangePasswordTemplate>
<table class="accountTable">
<tr>
<td rowspan="7" style="text-align: center; border-right: 1px
dotted #e5e5e5; padding: 10px;">
<p>
<b>Password change</b>

```

```

        </p>
        
        <br />
        
    </td>
    <td style="padding: 5px; padding-left: 10px;">
        <asp:Label ID="CurrentPasswordLabel" runat="server"
AssociatedControlID="CurrentPassword">Old Password:</asp:Label>
    </td>
    <td style="padding: 5px;">
        <asp:TextBox ID="CurrentPassword" runat="server"
CssClass="passwordEntry" TextMode="Password"></asp:TextBox>
    </td>
</tr>
<tr>
    <td style="padding: 5px; padding-left: 10px;" colspan="2">
        <asp:RequiredFieldValidator ID="RequiredFieldValidator1"
runat="server" ControlToValidate="CurrentPassword"
            CssClass="failureNotification" ErrorMessage="Old
password is required." ToolTip="Old password is required."

ValidationGroup="ChangeUserPasswordValidationGroup">This field is
required.</asp:RequiredFieldValidator>
    </td>
</tr>
<tr>
    <td style="padding: 5px; padding-left: 10px;">
        <asp:Label ID="NewPasswordLabel" runat="server"
AssociatedControlID="NewPassword">New Password:</asp:Label>
    </td>
    <td style="padding: 5px;">
        <asp:TextBox ID="NewPassword" runat="server"
CssClass="passwordEntry" TextMode="Password"></asp:TextBox>
    </td>
</tr>
<tr>
    <td style="padding: 5px; padding-left: 10px;" colspan="2">
        <asp:RequiredFieldValidator ID="NewPasswordRequired"
runat="server" ControlToValidate="NewPassword"
            CssClass="failureNotification" ErrorMessage="New
Password is required." ToolTip="New Password is required."

ValidationGroup="ChangeUserPasswordValidationGroup">This field is
required.</asp:RequiredFieldValidator>
    </td>
</tr>

```

```

<tr>
    <td style="padding: 5px; padding-left: 10px;">
        <asp:Label ID="ConfirmNewPasswordLabel" runat="server"
AssociatedControlID="ConfirmNewPassword">Confirm New Password:</asp:Label>
    </td>
    <td style="padding: 5px;">
        <asp:TextBox ID="ConfirmNewPassword" runat="server"
        CssClass="passwordEntry" TextMode="Password"></asp:TextBox>
    </td>
</tr>
<tr>
    <td style="padding: 5px; padding-left: 10px;" colspan="2">
        <asp:CompareValidator ID="CompareValidator1"
runat="server" ControlToCompare="NewPassword"
        Display="Dynamic"
ControlToValidate="ConfirmNewPassword" CssClass="failureNotification"
        ErrorMessage="Password confirmation error."
ValidationGroup="ChangeUserPasswordValidationGroup">Password confirmation
error.</asp:CompareValidator>
        <asp:RequiredFieldValidator
ID="ConfirmNewPasswordRequired" runat="server"
ControlToValidate="ConfirmNewPassword"
        CssClass="failureNotification" ErrorMessage="New
Password is required." ToolTip="New Password is required."
ValidationGroup="ChangeUserPasswordValidationGroup">This field is
required.</asp:RequiredFieldValidator>
    </td>
</tr>
<tr>
    <td style="padding: 5px;" colspan="2">
        <p class="submitButton">
            <asp:Button ID="ChangePasswordPushButton"
runat="server" CommandName="ChangePassword"
                Text="Change Password" ClientIDMode="Static"
ValidationGroup="ChangeUserPasswordValidationGroup" Width="150"
Height="50"></asp:Button>
            <asp:Button ID="CancelPushButton" runat="server"
CausesValidation="False" CommandName="Cancel" Width="100" Height="50"
                Text="Cancel"></asp:Button>
        </p>
    </td>
</tr>
<tr>
    <td style="padding: 5px;" colspan="3">
        <center>

```

```
<span class="failureNotification">
  <asp:Literal ID="FailureText"
runat="server"></asp:Literal>
</span>
</center>
</td>
</tr>
</table>
</ChangePasswordTemplate>
</asp:ChangePassword>
</asp:Content>
```

Кафедра КБПЗ – 2021 рік

```

using System;

namespace RoleManager
{
    public partial class ChangePassword : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
        }
    }
}
//-----
// <auto-generated>
//     This code was generated by a tool.
//
//     Changes to this file may cause incorrect behavior and will be lost if
//     the code is regenerated.
// </auto-generated>
//-----

namespace RoleManager {

    public partial class ChangePassword {

        /// <summary>
        /// ChangeUserPassword control.
        /// </summary>
        /// <remarks>
        /// Auto-generated field.
        /// To modify move field declaration from designer file to code-behind
file.
        /// </remarks>
        protected global::System.Web.UI.WebControls.ChangePassword
ChangeUserPassword;
    }
}
<%@ Page Title="Change Password" Language="C#" MasterPageFile="~/Site.master"
AutoEventWireup="true"
    CodeBehind="ChangePasswordSuccess.aspx.cs"
Inherits="RoleManager.ChangePasswordSuccess" %>

<asp:Content ID="HeaderContent" runat="server"
ContentPlaceHolderID="HeadContent">
</asp:Content>
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">

```

```

<table class="accountTable">
  <tr>
    <td style="text-align: center">
      <p>
        <b>Password changed!</b>
      </p>
      
    </td>
  </tr>
</table>
<%@ Page Title="Log In" Language="C#" MasterPageFile="~/Site.master"
AutoEventWireup="true"
CodeBehind="Login.aspx.cs" Inherits="RoleManager.Login" %>

<asp:Content ID="HeaderContent" runat="server"
ContentPlaceHolderID="HeadContent">
</asp:Content>
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">
  <asp:Login ID="LoginUser" runat="server" EnableViewState="false"
RenderOuterTable="false">
    <LayoutTemplate>
      <table class="accountTable">
        <tr>
          <td rowspan="3" style="text-align: center; border-right: 1px
dotted #e5e5e5; padding-right: 10px;">
            <p>
              <b>Enter account details</b>
            </p>
            
            <br />
            
          </td>
          <td style="padding: 5px; padding-left: 10px;">
            <asp:Label ID="UserNameLabel" runat="server"
AssociatedControlID="UserName">Login:</asp:Label>
          </td>
          <td style="padding: 5px;">
            <asp:TextBox ID="UserName" runat="server"
CssClass="textEntry"></asp:TextBox>
          </td>
        </tr>
        <tr>
          <td style="padding: 5px; padding-left: 10px;">
            <asp:Label ID="PasswordLabel" runat="server"
AssociatedControlID="Password">Password:</asp:Label>
          </td>

```

```
<td style="padding: 5px;">
    <asp:TextBox ID="Password" runat="server"
    CssClass="passwordEntry" TextMode="Password"></asp:TextBox>
</td>
</tr>
<tr>
    <td colspan="2" style="padding: 5px;">
        <p class="submitButton">
            <asp:Button ID="LoginButton" runat="server"
            CommandName="Login" Text="Log In" ValidationGroup="LoginUserValidationGroup"
            Width="200" Height="50" />
        </p>
    </td>
</tr>
<tr>
    <td colspan="3" style="padding: 5px;">
        <center>
            <span class="failureNotification">
                <asp:Literal ID="FailureText"
                runat="server"></asp:Literal>
            </span>
        </center>
    </td>
</tr>
</table>
</LayoutTemplate>
</asp:Login>
</asp:Content>
```

```
using System.Web;

namespace Auditor
{
    public class ActiveUser : AppUser
    {
        public ActiveUser() : base(GetLogin())
        {
        }

        public static bool IsAuthenticated =>
HttpContext.Current.User.Identity.IsAuthenticated;

        public static bool IsInRole(string role) => (IsAuthenticated &&
HttpContext.Current.User.IsInRole(role));

        private static string GetLogin() => (IsAuthenticated) ?
HttpContext.Current.User.Identity.Name.ToUpper().Trim() : null;
    }
}
```

Кафедра КБПЗ — 2021 рік

```

using System;
using System.Collections.Generic;

namespace Auditor
{
    public class AppUser
    {
        public string UserName { get; }
        public string FirstName { get; }
        public string LastName { get; }
        public string FullName { get; }
        public string Email { get; }
        public bool Exist { get; }

        public AppUser(string userName)
        {
            this.UserName = (userName != null) ? userName.ToUpper().Trim() :
null;

            this.Exist = false;
            if (this.UserName != null && UserExist(userName))
            {
                this.Exist = true;
                string query = @"SELECT
                                [Name],
                                [LastName],
                                [Email]
                                FROM [aspnet_Users]
                                WHERE [LoweredUserName] = @username;";
                var parameters = new Dictionary<string, object>() { {
"username", userName.ToLower().Trim() } };
                var result = DatabaseUtils.ExecuteSelectQuery(query, parameters,
DatabaseUtils.ConnectionTarget.Services);
                if (result != null)
                {
                    this.Email = Convert.ToString(result.Rows[0]["Email"]);
                    this.Email = (MailUtils.EmailAddressValid(this.Email)) ?
this.Email.ToLower().Trim() : null;

                    this.FirstName = Convert.ToString(result.Rows[0]["Name"]);
                    this.FirstName = (this.FirstName != null) ?
this.FirstName.Trim() : null;
                    this.LastName =
Convert.ToString(result.Rows[0]["LastName"]);
                    this.LastName = (this.LastName != null) ?
this.LastName.Trim() : null;

```

```

        this.FullName = this.FirstName + " " + this.LastName;
        this.FullName = (!string.IsNullOrEmpty(this.FullName))
? this.FullName.Trim() : null;
    }
}

public static bool UserExist(string userName)
{
    if (string.IsNullOrEmpty(userName)) return false;
    string query = @"SELECT
                        COUNT(*) AS [result]
                    FROM   [aspnet_Users]
                    WHERE  [LoweredUserName] = @username;";
    var parameters = new Dictionary<string, object>() { { "username",
userName.ToLower().Trim() } };
    var result =
Utils.ConvertToNullableInt(DatabaseUtils.ExecuteScalarSelectQuery(query,
parameters, DatabaseUtils.ConnectionTarget.Services));
    return (result != null && result > 0);
}

public static string GetUserMail(string userName)
{
    if (string.IsNullOrEmpty(userName)) return null;
    string query = @"SELECT
                        [Email]
                    FROM [aspnet_Users]
                    WHERE [LoweredUserName] = @username;";
    var parameters = new Dictionary<string, object>() { { "username",
userName.ToLower().Trim() } };
    var result =
Convert.ToString(DatabaseUtils.ExecuteScalarSelectQuery(query, parameters,
DatabaseUtils.ConnectionTarget.Services));
    return (MailUtils.EmailAddressValid(result)) ?
result.ToLower().Trim() : null;
}
}
}

```

```
using System;
using System.IO;
using System.Reflection;
using System.Text;
using System.Web;

namespace Auditor
{
    public static class AppUtils
    {
        public static string AppName =>
Assembly.GetExecutingAssembly().GetName().Name.Trim();

        public static string PageTitle
        {
            get
            {
                string fileName =
Path.GetFileNameWithoutExtension(HttpContext.Current.Request.FilePath).Trim();
                fileName = fileName.Replace("default",
string.Empty).Replace("Default", string.Empty);

                if (string.IsNullOrEmpty(fileName))
                {
                    return AppName;
                }
                else
                {
                    return $"{AppName}/{fileName}";
                }
            }
        }

        public static string CurrentUrl =>
HttpContext.Current.Request.Url.ToString();

        public static bool LocalhostMode =>
CurrentUrl.ToLower().Contains("localhost");

        public static void LogError(string errorMessage)
        {
            string logFilePath =
HttpContext.Current.Server.MapPath("~/App_Data/errorLog.txt");

            StringBuilder message = new StringBuilder();
            message.AppendLine(DateTime.Now.ToString());

```

```
message.AppendLine($"{CurrentUrl}\n");  
message.AppendLine(errorMessage);  
message.AppendLine("=====\n");  
  
File.AppendAllText(logFilePath, message.ToString());  
}  
}
```

Кафедра КБПЗ – 2021 рік

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Web.Configuration;

namespace Auditor
{
    public static class DatabaseUtils
    {
        public enum ConnectionTarget
        {
            Services,
            App
        }

        public static string ConnectionString(ConnectionTarget target)
        {
            string connectionString;
            switch (target)
            {
                case ConnectionTarget.Services:
                    connectionString =
WebConfigurationManager.ConnectionStrings["csServices"].ConnectionString;
                    break;

                case ConnectionTarget.App:
                    connectionString =
WebConfigurationManager.ConnectionStrings["csApp"].ConnectionString;
                    break;

                default:
                    connectionString = null;
                    break;
            }
            return connectionString;
        }

        /// <summary>
        /// Execute SQL query. Returns value from SCOPE_IDENTITY()
        /// </summary>
        public static int? ExecuteIdentityQuery(string query) =>
ExecuteIdentityQuery(query, null, ConnectionTarget.App);
    }
}
```

```

        public static int? ExecuteIdentityQuery(string query, Dictionary<string,
object> parameters) => ExecuteIdentityQuery(query, parameters,
ConnectionTarget.App);

        public static int? ExecuteIdentityQuery(string query, ConnectionTarget
target) => ExecuteIdentityQuery(query, null, target);

        public static int? ExecuteIdentityQuery(string query, Dictionary<string,
object> parameters, ConnectionTarget target)
        {
            int? scopeIdentity = null;
            QueryValidation(query);
            var parametersList = CreateParametersList(parameters);
            ParametersValidation(parametersList);
            query = query + Environment.NewLine + "SELECT SCOPE_IDENTITY()";
            try
            {
                using (SqlConnection connection = new
SqlConnection(connectionString(target)))
                using (SqlCommand command = new SqlCommand(query, connection))
                {
                    if (parametersList != null)
                    {
                        command.Parameters.AddRange(parametersList.ToArray());
                    }
                    connection.Open();
                    scopeIdentity =
Utils.ConvertToNullableInt(command.ExecuteScalar());
                    command.Parameters.Clear();
                    connection.Close();
                }
            }
            catch (Exception ex)
            {
                throw;
            }
            return scopeIdentity;
        }

        /// <summary>
        /// Execute SQL query. Returns value from ROWS_AFFECTED
        /// </summary>

        public static int ExecuteNonQuery(string query) =>
ExecuteNonQuery(query, null, ConnectionTarget.App);

```

```

public static int ExecuteNonQuery(string query, Dictionary<string,
object> parameters) => ExecuteNonQuery(query, parameters, ConnectionTarget.App);

public static int ExecuteNonQuery(string query, ConnectionTarget target)
=> ExecuteNonQuery(query, null, target);

public static int ExecuteNonQuery(string query, Dictionary<string,
object> parameters, ConnectionTarget target)
{
    int rowsAffected;
    QueryValidation(query);
    var parametersList = CreateParametersList(parameters);
    ParametersValidation(parametersList);
    try
    {
        using (SqlConnection connection = new
SqlConnection(connectionString(target)))
        using (SqlCommand command = new SqlCommand(query, connection))
        {
            if (parametersList != null)
            {
                command.Parameters.AddRange(parametersList.ToArray());
            }
            connection.Open();
            rowsAffected = command.ExecuteNonQuery();
            command.Parameters.Clear();
            connection.Close();
        }
    }
    catch (Exception ex)
    {
        throw;
    }
    return rowsAffected;
}

/// <summary>
/// Execute SELECT SQL query. Returns DataTable
/// </summary>

public static DataTable ExecuteSelectQuery(string query) =>
ExecuteSelectQuery(query, null, ConnectionTarget.App);

public static DataTable ExecuteSelectQuery(string query,
Dictionary<string, object> parameters) => ExecuteSelectQuery(query, parameters,
ConnectionTarget.App);

```

```

    public static DataTable ExecuteSelectQuery(string query,
        ConnectionTarget target) => ExecuteSelectQuery(query, null, target);

    public static DataTable ExecuteSelectQuery(string query,
        Dictionary<string, object> parameters, ConnectionTarget target)
    {
        DataTable result = new DataTable();
        QueryValidation(query);
        var parametersList = CreateParametersList(parameters);
        ParametersValidation(parametersList);
        try
        {
            using (SqlConnection connection = new
                SqlConnection(ConnectionString(target)))
            using (SqlCommand command = new SqlCommand(query, connection))
            using (SqlDataAdapter adapter = new SqlDataAdapter(command))
            {
                if (parametersList != null)
                {
                    command.Parameters.AddRange(parametersList.ToArray());
                }
                connection.Open();
                adapter.Fill(result);
                command.Parameters.Clear();
                connection.Close();
            }
        }
        catch (Exception ex)
        {
            throw;
        }
        return result;
    }

    /// <summary>
    /// Execute SELECT SQL query. Returns object
    /// </summary>
    public static object ExecuteScalarSelectQuery(string query) =>
        ExecuteScalarSelectQuery(query, null, ConnectionTarget.App);

    public static object ExecuteScalarSelectQuery(string query,
        Dictionary<string, object> parameters) => ExecuteScalarSelectQuery(query,
        parameters, ConnectionTarget.App);

```

```

public static object ExecuteScalarSelectQuery(string query,
ConnectionTarget target) => ExecuteScalarSelectQuery(query, null, target);

public static object ExecuteScalarSelectQuery(string query,
Dictionary<string, object> parameters, ConnectionTarget target)
{
    object result;
    QueryValidation(query);
    var parametersList = CreateParametersList(parameters);
    ParametersValidation(parametersList);
    try
    {
        using (SqlConnection connection = new
SqlConnection(ConnectionString(target)))
        using (SqlCommand command = new SqlCommand(query, connection))
        {
            if (parametersList != null)
            {
                command.Parameters.AddRange(parametersList.ToArray());
            }
            connection.Open();
            result = command.ExecuteScalar();
            command.Parameters.Clear();
            connection.Close();
        }
    }
    catch (Exception ex)
    {
        throw;
    }
    return result;
}

private static void QueryValidation(string query)
{
    if (string.IsNullOrEmpty(query))
    {
        throw new Exception("Empty SQL query!");
    }
}

private static List<SqlParameter>
CreateParametersList(Dictionary<string, object> parameters)
{
    var parametersList = new List<SqlParameter>();
    if (parameters != null)

```

```
{
    foreach (var parameter in parameters)
    {
        parametersList.Add(new SqlParameter(parameter.Key,
parameter.Value ?? DBNull.Value));
    }
}
return parametersList.Count > 0 ? parametersList : null;
}

private static void ParametersValidation(List<SqlParameter> parameters)
{
    if (parameters != null)
    {
        int distinctParameterNames = (from x in parameters
select
x.ParameterName).Distinct().Count();
        if (distinctParameterNames != parameters.Count())
        {
            throw new Exception("Parameter names are not unique!");
        }
    }
}
}
```

Кафедра КБПЗ 2021 рік

```
using DevExpress.Export;
using DevExpress.Web;
using DevExpress.XtraPrinting;
using System;
using System.Drawing;

namespace Auditor
{
    public static class GridViewUtils
    {
        public static string ToolbarGrid => "TOOLBAR_GRID";

        public static string ToolbarItem => "TOOLBAR_ITEM";

        public static string ToolbarBatch => "TOOLBAR_BATCH";

        public static int ImageSize => 20;

        public static int ImageToolbarSize => 16;

        public static void ProcessColumnAutoFilter(object sender,
ASPxGridViewAutoFilterEventArgs e)
        {
            e.Column.HeaderStyle.BackColor = ReferenceEquals(e.Criteria, null) ?
ColorTranslator.FromHtml("#DCDCDC") : ColorTranslator.FromHtml("#87C4EE");
        }

        public static void ExporterRenderBrick(object sender,
ASPxGridViewExportRenderingEventArgs e)
        {
            if (e.RowType == GridViewRowType.Header)
            {
                e.Text = e.Text.Replace("<br />", " ");
                e.Text = e.Text.Replace("<br/>", " ");
            }
        }

        public static XlsxExportOptionsEx ExporterOptions => new
XlsxExportOptionsEx() { ExportType = ExportType.WYSIWYG };

        public static void GridViewDefaultInit(object sender, EventArgs e)
        {
            ASPxGridView gridview = sender as ASPxGridView;

            gridview.SettingsDataSecurity.AllowInsert = false;
            gridview.SettingsDataSecurity.AllowEdit = false;
        }
    }
}
```

```
gridview.SettingsDataSecurity.AllowDelete = false;

gridview.SettingsCommandButton.RenderMode =
GridCommandButtonRenderMode.Image;
gridview.SettingsCommandButton.NewButton.Image.AlternateText =
"NEW";
gridview.SettingsCommandButton.NewButton.Image.ToolTip =
gridview.SettingsCommandButton.NewButton.Image.AlternateText;
gridview.SettingsCommandButton.NewButton.Image.Height =
GridViewUtils.ImageSize;
gridview.SettingsCommandButton.NewButton.Image.Width =
GridViewUtils.ImageSize;
gridview.SettingsCommandButton.NewButton.Image.Url =
"Images/add.png";
gridview.SettingsCommandButton.EditButton.Image.AlternateText =
"EDIT";
gridview.SettingsCommandButton.EditButton.Image.ToolTip =
gridview.SettingsCommandButton.EditButton.Image.AlternateText;
gridview.SettingsCommandButton.EditButton.Image.Height =
GridViewUtils.ImageSize;
gridview.SettingsCommandButton.EditButton.Image.Width =
GridViewUtils.ImageSize;
gridview.SettingsCommandButton.EditButton.Image.Url =
"Images/edit.png";
gridview.SettingsCommandButton.DeleteButton.Image.AlternateText =
"DELETE";
gridview.SettingsCommandButton.DeleteButton.Image.ToolTip =
gridview.SettingsCommandButton.DeleteButton.Image.AlternateText;
gridview.SettingsCommandButton.DeleteButton.Image.Height =
GridViewUtils.ImageSize;
gridview.SettingsCommandButton.DeleteButton.Image.Width =
GridViewUtils.ImageSize;
gridview.SettingsCommandButton.DeleteButton.Image.Url =
"Images/trash.png";
gridview.SettingsCommandButton.UpdateButton.Image.AlternateText =
"SAVE";
gridview.SettingsCommandButton.UpdateButton.Image.ToolTip =
gridview.SettingsCommandButton.UpdateButton.Image.AlternateText;
gridview.SettingsCommandButton.UpdateButton.Image.Height =
GridViewUtils.ImageSize;
gridview.SettingsCommandButton.UpdateButton.Image.Width =
GridViewUtils.ImageSize;
gridview.SettingsCommandButton.UpdateButton.Image.Url =
"Images/ok.png";
gridview.SettingsCommandButton.CancelButton.Image.AlternateText =
"CANCEL";
```

```
gridview.SettingsCommandButton.CancelButton.Image.ToolTip =
gridview.SettingsCommandButton.CancelButton.Image.AlternateText;
gridview.SettingsCommandButton.CancelButton.Image.Height =
GridViewUtils.ImageSize;
gridview.SettingsCommandButton.CancelButton.Image.Width =
GridViewUtils.ImageSize;
gridview.SettingsCommandButton.CancelButton.Image.Url =
"Images/nok.png";

gridview.Settings.AutoFilterCondition =
AutoFilterCondition.Contains;
gridview.Settings.ShowFilterRowMenu = true;
gridview.Settings.ShowHeaderFilterButton = true;
gridview.Settings.ShowTitlePanel = true;
gridview.Settings.VerticalScrollBarMode = ScrollBarMode.Visible;

gridview.SettingsBehavior.ConfirmDelete = true;
gridview.SettingsBehavior.AllowFocusedRow = true;

gridview.KeyboardSupport = true;

gridview.SettingsLoadingPanel.Mode =
GridViewLoadingPanelMode.ShowAsPopup;

gridview.SettingsPager.AlwaysShowPager = true;
gridview.SettingsPager.Mode = GridViewPagerMode.ShowPager;
gridview.SettingsPager.PageSize = 500;

gridview.SettingsSearchPanel.AllowTextInputTimer = true;
gridview.SettingsSearchPanel.Delay = 5000;

gridview.SettingsPopup.EditForm.AllowResize = false;
gridview.SettingsPopup.EditForm.HorizontalAlign =
PopupHorizontalAlign.WindowCenter;
gridview.SettingsPopup.EditForm.VerticalAlign =
PopupVerticalAlign.WindowCenter;
gridview.SettingsPopup.EditForm.Modal = true;

gridview.SettingsExport.ExcelExportMode = ExportType.WYSIWYG;
gridview.SettingsExport.EnableClientSideExportAPI = true;
gridview.SettingsExport.FileName = "DATA";

gridview.Styles.AlternatingRow.Enabled =
DevExpress.Utils.DefaultBoolean.True;
gridview.Styles.Cell.HorizontalAlign =
System.Web.UI.WebControls.HorizontalAlign.Center;
```

```

        gridView.Styles.Header.HorizontalAlign =
System.Web.UI.WebControls.HorizontalAlign.Center;
        gridView.Styles.Header.Wrap = DevExpress.Utils.DefaultBoolean.True;
        gridView.Styles.TitlePanel.Font.Bold = true;
        gridView.StylesEditors.ReadOnly.BackColor = Color.Black;
        gridView.StylesEditors.ReadOnly.ForeColor = Color.White;
        gridView.Font.Size = new System.Web.UI.WebControls.FontUnit("10px");
        gridView.Styles.TitlePanel.BackColor =
ColorTranslator.FromHtml("#64645B");
        gridView.Styles.FocusedRow.BackColor =
ColorTranslator.FromHtml("#0080FF");
        gridView.Styles.BatchEditModifiedCell.ForeColor = Color.Black;

        gridView.Styles.LoadingPanel.BackColor =
ColorTranslator.FromHtml("#FFFFB0");
        gridView.Styles.LoadingPanel.ForeColor =
ColorTranslator.FromHtml("#0080FF");
        gridView.Styles.LoadingPanel.Font.Bold = true;
        gridView.Styles.LoadingPanel.Font.Size = new
System.Web.UI.WebControls.FontUnit("12px");
    }

    public static void GridViewToolBarInit(object sender, EventArgs e)
    {
        ASPxGridView gridView = sender as ASPxGridView;

        var exportExcel = new GridViewToolBarItem();
        exportExcel.Command = GridViewToolBarCommand.ExportToXlsx;
        var search = new GridViewToolBarItem();
        search.Command = GridViewToolBarCommand.ShowSearchPanel;
        var refresh = new GridViewToolBarItem();
        refresh.Command = GridViewToolBarCommand.Refresh;
        var clear = new GridViewToolBarItem();
        clear.Command = GridViewToolBarCommand.ClearFilter;
        var newItem = new GridViewToolBarItem();
        newItem.Command = GridViewToolBarCommand.New;
        var deleteItem = new GridViewToolBarItem();
        deleteItem.Command = GridViewToolBarCommand.Delete;
        var editItem = new GridViewToolBarItem();
        editItem.Command = GridViewToolBarCommand.Edit;
        var update = new GridViewToolBarItem();
        update.Command = GridViewToolBarCommand.Update;
        update.Image.Url = "Images/ok.png";
        update.Image.Height = GridViewUtils.ImageToolBarSize;
        update.Image.Width = GridViewUtils.ImageToolBarSize;
        var cancel = new GridViewToolBarItem();

```

```
cancel.Command = GridViewToolBarCommand.Cancel;
cancel.Image.Url = "Images/nok.png";
cancel.Image.Height = GridViewUtils.ImageToolBarSize;
cancel.Image.Width = GridViewUtils.ImageToolBarSize;
var filter = new GridViewToolBarItem();
filter.Command = GridViewToolBarCommand.ShowFilterRow;
filter.Image.Url = "Images/filter.png";
filter.Image.Height = GridViewUtils.ImageToolBarSize;
filter.Image.Width = GridViewUtils.ImageToolBarSize;

var toolbarGrid = new GridViewToolBar();
toolbarGrid.Name = GridViewUtils.ToolbarGrid;
toolbarGrid.Position = GridToolBarPosition.Top;
toolbarGrid.Items.Add(refresh);
toolbarGrid.Items.Add(search);
toolbarGrid.Items.Add(filter);
toolbarGrid.Items.Add(clear);
toolbarGrid.Items.Add(exportExcel);
if (toolbarGrid.Items.Count > 0)
{
    gridview.Toolbars.Add(toolbarGrid);
}

var toolbarItem = new GridViewToolBar();
toolbarItem.Name = GridViewUtils.ToolbarItem;
toolbarItem.Position = GridToolBarPosition.Top;
if (gridview.SettingsDataSecurity.AllowInsert)
{
    toolbarItem.Items.Add(newItem);
}
if (gridview.SettingsDataSecurity.AllowEdit)
{
    toolbarItem.Items.Add(editItem);
}
if (gridview.SettingsDataSecurity.AllowDelete)
{
    toolbarItem.Items.Add(deleteItem);
}
if (toolbarItem.Items.Count > 0)
{
    gridview.Toolbars.Add(toolbarItem);
}

if (gridview.SettingsEditing.Mode == GridViewEditingMode.Batch
    && (gridview.SettingsDataSecurity.AllowInsert
        || gridview.SettingsDataSecurity.AllowEdit
```

```
        || gridView.SettingsDataSecurity.AllowDelete))
    {
        var toolbarBatch = new GridViewToolbar();
        toolbarBatch.Name = GridViewUtils.ToolbarBatch;
        toolbarBatch.Position = GridToolbarPosition.Bottom;
        toolbarBatch.Items.Add(update);
        toolbarBatch.Items.Add(cancel);
        gridView.Settings.ShowStatusBar = GridViewStatusBarMode.Hidden;
        gridView.Toolbars.Add(toolbarBatch);
    }
}
}
```

Кафедра КБПЗ – 2021 рік

```

using System;
using System.Collections.Generic;
using System.Net.Mail;
using System.Text.RegularExpressions;

namespace Auditor
{
    public static class MailUtils
    {
        //TODO change mail server settings
        private const string mailServerIp = "*****";

        private const int mailServerPort = 25;
        private const string mailSender = "AppNotifier@DoNotReply.com";

        public static void SendEmail(string subject, string body, List<string>
recipients) => SendEmail(subject, body, recipients, new List<Attachment>());

        public static void SendEmail(string subject, string body, List<string>
recipients, List<Attachment> attachments)
        {
            try
            {
                using (SmtpClient client = new SmtpClient(mailServerIp,
mailServerPort))
                using (MailMessage message = new MailMessage())
                {
                    message.From = new MailAddress(mailSender);
                    foreach (string mail in recipients)
                    {
                        if (EmailAddressValid(mail))
                        {
                            message.To.Add(mail);
                        }
                    }
                    if (message.To.Count == 0)
                    {
                        throw new Exception("Email sending error - not a single
recipient mail address is valid!");
                    }
                    message.IsBodyHtml = true;
                    message.Subject = subject;
                    message.Body = body;
                    foreach (var attachment in attachments)
                    {
                        message.Attachments.Add(attachment);
                    }
                }
            }
        }
    }
}

```

```

        }
        client.Send(message);
    }
}
catch (Exception ex)
{
    if (ex.Message == "Email sending error - not a single recipient
mail address is valid!")
    {
        string errorMessage = ex.ToString();
        AppUtils.LogError(errorMessage);
    }
    else
    {
        throw;
    }
}
}

public static bool EmailAddressValid(string emailAddress)
{
    if (string.IsNullOrEmpty(emailAddress)) return false;
    string mailPattern = "\\A(?:[a-z0-9!#$%&'*/+=?^_`{|}~-]+(?:\\.([a-z0-9!#$%&'*/+=?^_`{|}~-]+)*@(?:[a-z0-9](?:[a-z0-9-]*[a-z0-9])?\\.)+[a-z0-9](?:[a-z0-9-]*[a-z0-9])?)\\Z";
    bool isEmail = Regex.IsMatch(emailAddress, mailPattern,
RegexOptions.IgnoreCase);
    return isEmail;
}
}
}

```

```

using System;
using System.Collections.Generic;
using System.Web.Security;

namespace Auditor
{
    public static class RoleManagement
    {
        public static void RoleRegister(string roleName, string description)
        {
            if (!Roles.RoleExists(roleName))
            {
                Roles.CreateRole(roleName);
                RoleUpdate(roleName, description);
            }
            else
            {
                throw new Exception("Role already exists!");
            }
        }

        public static void RoleUpdate(string roleName, string description)
        {
            if (Roles.RoleExists(roleName))
            {
                string query = @"UPDATE [aspnet_Roles]
                                SET [description] = @description
                                WHERE [loweredrolename] = @rolename;";
                var parameters = new Dictionary<string, object>() { {
                    "description", description },
                    {
                    "rolename", roleName } };
                DatabaseUtils.ExecuteNonQuery(query, parameters,
                DatabaseUtils.ConnectionTarget.Services);
            }
            else
            {
                throw new Exception("Role does not exist!");
            }
        }

        public static void RoleDelete(string roleName)
        {
            if (Roles.RoleExists(roleName))
            {
                string[] usersInRole = Roles.GetUsersInRole(roleName);

```

```
        if (usersInRole.Length > 0)
        {
            Roles.RemoveUsersFromRole(usersInRole, roleName);
        }
        Roles.DeleteRole(roleName);
    }
    else
    {
        throw new Exception("Role does not exist!");
    }
}
}
```

Кафедра КБПЗ – 2021 рік

```
using System;
using System.Collections.Generic;
using System.Web.Security;

namespace Auditor
{
    public static class UserManagement
    {
        public static void UserRegister(string username, string name, string
lastName, string email)
        {
            if (Membership.GetUser(username) == null)
            {
                var password = GeneratePassword();
                Membership.CreateUser(username, password);
                UserUpdate(username, name, lastName, email);
                SendRegisterNotification(username, password);
            }
            else
            {
                throw new Exception("Username is already in use!");
            }
        }

        public static void UserDelete(string username)
        {
            if (Membership.GetUser(username) != null)
            {
                Membership.DeleteUser(username);
            }
            else
            {
                throw new Exception("User does not exist!");
            }
        }

        public static void UserUpdate(string username, string name, string
lastName, string email)
        {
            if (Membership.GetUser(username) != null)
            {
                string query = @"UPDATE [aspnet_Users]
                                SET
                                    [name] = @name,
                                    [lastname] = @lastname,
                                    [email] = @email
```

```

WHERE
        [loweredusername] = @username;";
        var parameters = new Dictionary<string, object>() { { "name",
name },
        {
"lastname", lastName },
        { "email",
email },
        {
"username", username } };
        DatabaseUtils.ExecuteNonQuery(query, parameters,
DatabaseUtils.ConnectionTarget.Services);
    }
    else
    {
        throw new Exception("User does not exist!");
    }
}

public static void UserUnlock(string username)
{
    var member = Membership.GetUser(username);
    if (member != null)
    {
        member.UnlockUser();
        SendUnlockNotification(username);
    }
    else
    {
        throw new Exception("User does not exist!");
    }
}

public static void UserResetPassword(string username)
{
    var member = Membership.GetUser(username);
    if (member != null)
    {
        var password = GeneratePassword();
        member.UnlockUser();
        member.ChangePassword(member.ResetPassword(), password);
        SendChangePasswordNotification(username, password);
    }
    else
    {
        throw new Exception("User does not exist!");
    }
}

```

```

    }
}

public static void UserRemoveRole(string username, string roleName)
{
    if (Membership.GetUser(username) != null &&
Roles.RoleExists(roleName) && Roles.IsUserInRole(username, roleName))
    {
        Roles.RemoveUserFromRole(username, roleName);
    }
    else
    {
        throw new Exception("User is not in this role! OR Role/user may
not exist!");
    }
}

public static void UserAddRole(string username, string roleName)
{
    if (Membership.GetUser(username) != null &&
Roles.RoleExists(roleName) && !Roles.IsUserInRole(username, roleName))
    {
        Roles.AddUserToRole(username, roleName);
    }
    else
    {
        throw new Exception("User is already in this role! OR Role/user
may not exist!");
    }
}

private static string GeneratePassword() => (new Random()).Next(0,
999999).ToString("D6");

private static void SendRegisterNotification(string username, string
password)
{
    var subject = $"Account {username.ToUpper()} creation!";
    var body = $"Your password: {password}";
    var email = AppUser.GetUserMail(username);
    MailUtils.SendEmail(subject, body, new List<string> { email });
}

private static void SendUnlockNotification(string username)
{
    var subject = $"Account {username.ToUpper()} unlocked!";
}

```

```
var body = string.Empty;
var email = AppUser.GetUserMail(username);
MailUtils.SendEmail(subject, body, new List<string> { email });
}

private static void SendChangePasswordNotification(string username,
string password)
{
    var subject = $"Account {username.ToUpper()} password reseted!";
    var body = $"Your new password: {password}";
    var email = AppUser.GetUserMail(username);
    MailUtils.SendEmail(subject, body, new List<string> { email });
}
}
}
```

Кафедра КБПЗ – 2021 рік

```
using DevExpress.XtraReports.UI;
using System;
using System.IO;
using System.Web;

namespace Auditor
{
    public static class Utils
    {
        public static bool? ConvertToNullableBool(object value)
        {
            bool parsedValue;
            if (value != null && bool.TryParse(value.ToString(), out
parsedValue))
            {
                return parsedValue;
            }
            else
            {
                return null;
            }
        }

        public static int? ConvertToNullableInt(object value)
        {
            int parsedValue;
            if (value != null && int.TryParse(value.ToString(), out
parsedValue))
            {
                return parsedValue;
            }
            else
            {
                return null;
            }
        }

        public static decimal? ConvertToNullableDecimal(object value)
        {
            decimal parsedValue;
            if (value != null && decimal.TryParse(value.ToString().Replace(".",
","), out parsedValue))
            {
                return parsedValue;
            }
            else
            {
                return null;
            }
        }
    }
}
```

```
{
    return null;
}
}

public static DateTime? ConvertToNullableDateTime(object value)
{
    DateTime parsedValue;
    if (value != null && DateTime.TryParse(value.ToString(), out
parsedValue))
    {
        return parsedValue;
    }
    else
    {
        return null;
    }
}

public static string ConvertToTrimmedString(object value)
{
    string parsedValue = Convert.ToString(value);
    if (!string.IsNullOrEmpty(parsedValue))
    {
        return parsedValue.Trim();
    }
    else
    {
        return null;
    }
}

public static void PrintPDF(XtraReport report, string fileName)
{
    fileName = Utils.ConvertToTrimmedString(fileName);
    if (fileName == null)
    {
        throw new Exception("File name can not be empty string!");
    }
    fileName = (fileName.ToLower().EndsWith(".pdf")) ? fileName :
"${fileName}.pdf";
    using (MemoryStream stream = new MemoryStream())
    {
        report.ExportToPdf(stream);
        HttpContext.Current.Response.Clear();
        HttpContext.Current.Response.ContentType = "application/pdf";
    }
}
```

```
HttpContext.Current.Response.AddHeader("Accept-Header",
stream.Length.ToString());
HttpContext.Current.Response.AddHeader("Content-Disposition",
$"Inline; filename={fileName}");
HttpContext.Current.Response.AddHeader("Content-Length",
stream.Length.ToString());
HttpContext.Current.Response.BinaryWrite(stream.ToArray());
HttpContext.Current.Response.End();
    }
}
}
```

Кафедра КБПЗ – 2021 рік

```

using System;
using System.Collections.Generic;

namespace Auditor
{
    public class Action
    {
        #region audit fields

        public int Id { get; }
        public int? AuditId { get; }
        public int? AuditDetailId { get; }
        public string ResponsibleLogin { get; }
        public string ResponsibleFullName { get; }
        public string ActionText { get; }
        public DateTime? PlannedTerm { get; }
        public DateTime? Term { get; }
        public DateTime? LastEditDate { get; }

        #endregion audit fields

        public Action(int id)
        {
            if (ActionExist(id))
            {
                string query = @"SELECT
                                [id],
                                [audit_id],
                                [audit_detail_id],
                                [responsible_login],
                                [responsible_full_name],
                                [action],
                                [planned_term],
                                [term],
                                [last_edit_date]
                                FROM [actions]
                                WHERE [id] = @id;";
                var parameters = new Dictionary<string, object>() { { "id", id } };
            };

            var result = DatabaseUtils.ExecuteSelectQuery(query,
parameters);

            if (result != null)
            {
                this.Id = id;
                this.AuditId =
Utils.ConvertToNullableInt(result.Rows[0]["audit_id"]);

```

```

        this.AuditDetailId =
Utils.ConvertToNullableInt(result.Rows[0]["audit_detail_id"]);
        this.ResponsibleLogin =
Utils.ConvertToTrimmedString(result.Rows[0]["responsible_login"]);
        this.ResponsibleFullName =
Utils.ConvertToTrimmedString(result.Rows[0]["responsible_full_name"]);
        this.ActionText =
Utils.ConvertToTrimmedString(result.Rows[0]["action"]);
        this.PlannedTerm =
Utils.ConvertToNullableDateTime(result.Rows[0]["planned_term"]);
        this.Term =
Utils.ConvertToNullableDateTime(result.Rows[0]["term"]);
        this.LastEditDate =
Utils.ConvertToNullableDateTime(result.Rows[0]["last_edit_date"]);
    }
}
}

public bool ActionExist() => ActionExist(this.Id);

public void Confirm()
{
    string query = @"UPDATE [actions]
        SET
            [term] = GETDATE()
        WHERE [id] = @id";
    var parameters = new Dictionary<string, object>() { { "id", this.Id
} };
    DatabaseUtils.ExecuteNonQuery(query, parameters);
}

public void Clear()
{
    string query = @"UPDATE [actions]
        SET
            [action] = NULL,
            [planned_term] = NULL,
            [term] = NULL,
            [last_edit_date] = NULL
        WHERE [id] = @id";
    var parameters = new Dictionary<string, object>() { { "id", this.Id
} };
    DatabaseUtils.ExecuteNonQuery(query, parameters);
}

public static bool ActionExist(int id)

```

```
{
    string query = "SELECT [id] FROM [actions] WHERE [id] = @id";
    var parameters = new Dictionary<string, object>() { { "id", id } };
    var result =
Utils.ConvertToNullableInt(DatabaseUtils.ExecuteScalarSelectQuery(query,
parameters));
    return result != null;
}
}
```

Кафедра КБПЗ – 2021 рік

```

using System;
using System.Collections.Generic;
using System.Data;
using System.IO;
using System.Net.Mail;

namespace Auditor
{
    public class Audit
    {
        #region audit fields

        public int Id { get; }
        public string Type { get; }
        public string TypeName { get; }
        public string Target { get; }
        public string AuditorLogin { get; }
        public string AuditorFullName { get; }
        public DateTime? StartDate { get; }
        public DateTime? EndDate { get; }
        public decimal? Score { get; }
        public string Comment { get; }
        public string TargetArea { get; }
        public string TargetSubarea { get; }
        public string TargetSection { get; }
        public string ShiftName { get; }

        #endregion audit fields

        #region calculated fields

        public DataTable AuditQuestionGroups { get { return
GetAuditQuestionGroups(); } }
        public int AuditQuestionCount { get { return GetAuditQuestionCount(); } }
    }
    public int AuditQuestionAnsweredCount { get { return
GetAuditQuestionAnsweredCount(); } }
        public decimal AuditProgress { get { return (AuditQuestionCount != 0) ?
100 * AuditQuestionAnsweredCount / AuditQuestionCount : 0; } }
        public decimal AuditScoreCalculated { get { return
AuditTypes.CalculateAuditScore(this); } }

        #endregion calculated fields

        public Audit(int id)
        {

```

```

if (AuditExist(id))
{
    string query = @"SELECT
                                [id],
                                [audit_type],
                                [audit_type_name],
                                [audit_target],
                                [auditor_login],
                                [auditor_full_name],
                                [start_date],
                                [end_date],
                                [score],
                                [comment],
                                [audit_target_area],
                                [audit_target_subarea],
                                [audit_target_section],
                                [audit_shift_name]
                                FROM [audits]
                                WHERE [id] = @id;";

    var parameters = new Dictionary<string, object>() { { "id", id }
};

    var result = DatabaseUtils.ExecuteSelectQuery(query,
parameters);

    if (result != null)
    {
        this.Id = id;
        this.Type =
Utils.ConvertToTrimmedString(result.Rows[0]["audit_type"]);
        this.TypeName =
Utils.ConvertToTrimmedString(result.Rows[0]["audit_type_name"]);
        this.Target =
Utils.ConvertToTrimmedString(result.Rows[0]["audit_target"]);
        this.AuditorLogin =
Utils.ConvertToTrimmedString(result.Rows[0]["auditor_login"]);
        this.AuditorFullName =
Utils.ConvertToTrimmedString(result.Rows[0]["auditor_full_name"]);
        this.StartDate =
Utils.ConvertToNullableDateTime(result.Rows[0]["start_date"]);
        this.EndDate =
Utils.ConvertToNullableDateTime(result.Rows[0]["end_date"]);
        this.Score =
Utils.ConvertToNullableDecimal(result.Rows[0]["score"]);
        this.Comment =
Utils.ConvertToTrimmedString(result.Rows[0]["comment"]);
        this.TargetArea =
Utils.ConvertToTrimmedString(result.Rows[0]["audit_target_area"]);

```

```

        this.TargetSubarea =
Utils.ConvertToTrimmedString(result.Rows[0]["audit_target_subarea"]);
        this.TargetSection =
Utils.ConvertToTrimmedString(result.Rows[0]["audit_target_section"]);
        this.ShiftName =
Utils.ConvertToTrimmedString(result.Rows[0]["audit_shift_name"]);
    }
}

private int GetAuditQuestionCount()
{
    string query = @"SELECT COUNT(*)
                    FROM [audit_details]
                    WHERE [audit_id]= @audit_id;";
    var parameters = new Dictionary<string, object>() { { "audit_id",
this.Id } };
    var result =
Utils.ConvertToNullableInt(DatabaseUtils.ExecuteScalarSelectQuery(query,
parameters));
    return (result != null && result > 0) ? (int)result : 0;
}

private int GetAuditQuestionAnsweredCount()
{
    string query = @"SELECT COUNT(*)
                    FROM [audit_details]
                    WHERE [audit_id]= @audit_id AND [answer] IS NOT
NULL;";
    var parameters = new Dictionary<string, object>() { { "audit_id",
this.Id } };
    var result =
Utils.ConvertToNullableInt(DatabaseUtils.ExecuteScalarSelectQuery(query,
parameters));
    return (result != null && result > 0) ? (int)result : 0;
}

public int GetAuditQuestionAnswersCount(string answer)
{
    string query = @"SELECT COUNT(*)
                    FROM [audit_details]
                    WHERE [audit_id]= @audit_id AND [answer] =
@answer;";
    var parameters = new Dictionary<string, object>() { { "audit_id",
this.Id },

```

```

        { "answer",
answer} };

        var result =
Utils.ConvertToNullableInt(DatabaseUtils.ExecuteScalarSelectQuery(query,
parameters));

        return (result != null && result > 0) ? (int)result : 0;
    }

private DataTable GetAuditQuestionGroups()
{
    string query = @"SELECT DISTINCT
                        [group_position],
                        [group_name],
                        [group_name_ENG]
FROM [audit_details]
WHERE [audit_id]= @audit_id
ORDER BY [group_position];";

    var parameters = new Dictionary<string, object>() { { "audit_id",
this.Id } };

    var result = DatabaseUtils.ExecuteSelectQuery(query, parameters);
    return (result.Rows.Count > 0) ? result : null;
}

public bool AuditClosed() => AuditClosed(this.Id);

public bool AuditExist() => AuditExist(this.Id);

public void End()
{
    string query = @"UPDATE [audits]
                        SET [score] = @score,
                        [end_date] = GETDATE()
                        WHERE [id] = @audit_id;";

    var parameters = new Dictionary<string, object>() { { "audit_id",
this.Id },
                                                    { "score",
this.AuditScoreCalculated } };

    DatabaseUtils.ExecuteNonQuery(query, parameters);
    AuditTypes.GenerateAuditActions(this);
    AuditTypes.NotifySupervisor(this);
    PhotoFiles.EndAuditRemove(this);
}

public void Delete()
{
    PhotoFiles.DeleteAuditRemove(this);
}

```

```

        string query = @"DELETE FROM [audits]
                        WHERE [id] = @audit_id;
                        DELETE FROM [audit_details]
                        WHERE [audit_id] = @audit_id;
                        DELETE FROM [actions]
                        WHERE [audit_id] = @audit_id;";
        var parameters = new Dictionary<string, object>() { { "audit_id",
this.Id } };
        DatabaseUtils.ExecuteNonQuery(query, parameters);
    }

    public static Audit GetUserActiveAudit(string user)
    {
        string query = "SELECT [id] FROM [audits] WHERE [end_date] IS NULL
AND [auditor_login] = @auditor_login";
        var parameters = new Dictionary<string, object>() { {
"auditor_login", user } };
        var result =
Utils.ConvertToNullableInt(DatabaseUtils.ExecuteScalarSelectQuery(query,
parameters));
        return (result != null) ? new Audit((int)result) : null;
    }

    public static Audit CreateNew(string auditType, string auditTarget,
string auditorLogin, string auditorFullName, string auditShiftName)
    {
        var auditId = InsertAuditHeader(auditType, auditTarget,
auditorLogin, auditorFullName, auditShiftName);
        if (auditId != null)
        {
            if (InsertAuditDetails((int)auditId) > 0)
            {
                return new Audit((int)auditId);
            }
            else
            {
                new Audit((int)auditId).Delete();
                return null;
            }
        }
        else
        {
            return null;
        }
    }
}

```

```

public static void AnswerQuestion(int auditDetailsId, string answer)
{
    string query = "UPDATE [audit_details] SET [answer] = @answer WHERE
[id] = @id";
    var parameters = new Dictionary<string, object>() { { "id",
auditDetailsId },
                                                    { "answer",
answer } };
    DatabaseUtils.ExecuteNonQuery(query, parameters);
}

public static bool AuditClosed(int id)
{
    string query = "SELECT [end_date] FROM [audits] WHERE [id] = @id";
    var parameters = new Dictionary<string, object>() { { "id", id } };
    var result =
Utils.ConvertToNullableDateTime(DatabaseUtils.ExecuteScalarSelectQuery(query,
parameters));
    return result != null;
}

public static bool AuditExist(int id)
{
    string query = "SELECT [id] FROM [audits] WHERE [id] = @id";
    var parameters = new Dictionary<string, object>() { { "id", id } };
    var result =
Utils.ConvertToNullableInt(DatabaseUtils.ExecuteScalarSelectQuery(query,
parameters));
    return result != null;
}

public static string GetAuditDetailFolder(int auditDetailId)
{
    string query = "SELECT
'G'+CONVERT(NVARCHAR(10), [group_position])+'Q'+CONVERT(NVARCHAR(10), [question_po
sition]) FROM [audit_details] WHERE [id] = @id";
    var parameters = new Dictionary<string, object>() { { "id",
auditDetailId } };
    var result =
Utils.ConvertToTrimmedString(DatabaseUtils.ExecuteScalarSelectQuery(query,
parameters));
    return result == null ? "error" : result;
}

public void SendEmailWithPdf(string lang)
{

```

```

        var fileName =
$"Audit_ID{this.Id}_{DateTime.Now.ToString("yyyyMMdd")}.pdf";
        var subject = $"Printout {this.TypeName} | {this.Target}";
        var body = "";
        var email = AppUser.GetUserMail(this.AuditorLogin);
        if (email == null)
        {
            throw new Exception("You can not send an e-mail with the report
- no e-mail in the system / wrong format!");
        }
        var report = AuditTypes.GetXtraReport(this, lang);
        using (MemoryStream stream = new MemoryStream())
        {
            report.ExportToPdf(stream);
            stream.Position = 0;
            var attach = new Attachment(stream, fileName,
"application/pdf");
            MailUtils.SendEmail(subject, body, new List<string> { email },
new List<Attachment> { attach });
        }
        throw new Exception("An email with a PDF printout has been sent!
Check your inbox!");
    }

    private static int? InsertAuditHeader(string auditType, string
auditTarget, string auditorLogin, string auditorFullName, string auditShiftName)
    {
        string query = @"INSERT INTO [audits]
([audit_type],[audit_type_name],[audit_target],[audit_target_area],[audit_target
_subarea],[audit_target_section],[auditor_login],[auditor_full_name],[start_date
],[audit_shift_name])

SELECT TOP 1
        [types].[audit_type],
        [types].[name] AS [audit_type_name],
        [targets].[audit_target],
        [targets].[area] AS [audit_target_area],
        [targets].[subarea] AS
[audit_target_subarea],
        [targets].[section] AS
[audit_target_section],
        @auditor_login AS [auditor_login],
        @auditor_full_name AS
[auditor_full_name],
        GETDATE() AS [start_date],
        @audit_shift_name AS [audit_shift_name]
FROM [setts_audit_types] [types]

```

```

LEFT JOIN [setts_audit_targets] [targets] ON
[types].[audit_type] = [targets].[audit_type]
WHERE [types].[audit_type] = @audit_type AND
[targets].[audit_target] = @audit_target";
var parameters = new Dictionary<string, object>() { { "audit_type",
auditType } ,
{
"audit_target", auditTarget } ,
{
"auditor_login", auditorLogin },
{
"auditor_full_name", auditorFullName },
{
"audit_shift_name", auditShiftName } };
return DatabaseUtils.ExecuteIdentityQuery(query, parameters);
}

private static int InsertAuditDetails(int auditId)
{
string query = @"INSERT INTO [audit_details]
([audit_id],[group_position],[group_name],[group_name_ENG],[question_position],[
question],[question_ENG],[answer_OK],[answer_NOK],[answer_NA],[answer_NC])
SELECT
[audit].[id] AS [audit_id],
[groups].[group_position],
[groups].[group_name],
[groups].[group_name_ENG],
[questions].[question_position],
[questions].[question],
[questions].[question_ENG],
[questions].[answer_OK],
[questions].[answer_NOK],
[questions].[answer_NA],
[questions].[answer_NC]
FROM [audits] AS [audit]
LEFT JOIN [setts_audit_question_groups] AS
[groups] ON [groups].[audit_type] = [audit].[audit_type]
LEFT JOIN [setts_audit_questions] AS
[questions] ON [questions].[group_id] = [groups].[id]
WHERE [audit].[id] = @audit_id;";
var parameters = new Dictionary<string, object>() { { "audit_id",
auditId } };
return DatabaseUtils.ExecuteNonQuery(query, parameters);
}
}
}

```

```

using DevExpress.Web;
using DevExpress.XtraReports.UI;
using System;
using System.Collections.Generic;
using System.IO;
using System.Net.Mail;

namespace Auditor
{
    public abstract class AuditTypes
    {
        //TODO change audit types according to requirements

        public const string Audit5sLpaProduction = "5s_lpa_prod";
        public const string Audit5sAdministration = "5s_admin";
        public static List<string> AuditTypesList = new List<string>() {
Audit5sLpaProduction, Audit5sAdministration };

        public static void AdjustTemplateToAuditType(string auditType,
ASPxComboBox cbAuditTarget, ASPxComboBox cbAuditShiftName)
        {
            cbAuditTarget.Value = null;
            cbAuditShiftName.Value = null;

            cbAuditTarget.Visible = auditType != null;
            cbAuditShiftName.Visible = auditType == Audit5sLpaProduction;

            if (cbAuditTarget.Visible)
            {
                foreach (ListBoxColumn column in cbAuditTarget.Columns)
                {
                    column.Visible = false;
                }
                cbAuditTarget.Columns[0].Visible = true; //target
                if (auditType == Audit5sAdministration || auditType ==
Audit5sLpaProduction)
                {
                    cbAuditTarget.Columns[1].Visible = true; //area
                    cbAuditTarget.Columns[2].Visible = true; //subarea
                    cbAuditTarget.Columns[4].Visible = true; //supervisor
                }
            }
        }

        public static bool ShiftNameObligatory(string auditType) => auditType ==
Audit5sLpaProduction;

```

```

public static decimal CalculateAuditScore(Audit audit)
{
    decimal result = 0;
    var cntQuestion = audit.AuditQuestionCount;
    var cntOK = audit.GetAuditQuestionAnswersCount(Answers.OK);
    var cntNOK = audit.GetAuditQuestionAnswersCount(Answers.NOK);
    var cntNC = audit.GetAuditQuestionAnswersCount(Answers.NC);
    var cntNA = audit.GetAuditQuestionAnswersCount(Answers.NA);
    switch (audit.Type)
    {
        default:
            result = (cntQuestion != cntNA) ? 100m * cntOK /
(cntQuestion - cntNA) : 0;
            break;
    }
    return result;
}

public static void GenerateAuditActions(Audit audit)
{
    string query = @"SELECT 0";
    var parameters = new Dictionary<string, object>();

    if (audit.Type == Audit5sAdministration || audit.Type ==
Audit5sLpaProduction)
    {
        query = @"INSERT INTO [actions]
([audit_id],[audit_detail_id],[responsible_login],[responsible_full_name])
        SELECT
            [ad].[audit_id],
            [ad].[id] AS
[audit_detail_id],
            [t].[supervisor_login] AS
[responsible_login],
            [t].[supervisor] AS
[responsible_full_name]
        FROM [audit_details] AS [ad]
        LEFT JOIN [audits] AS [a] ON [a].[id] =
[ad].[audit_id]
        LEFT JOIN [setts_audit_targets] AS [t] ON
[t].[audit_type] = [a].[audit_type]
        AND [t].[audit_target] = [a].[audit_target]
        WHERE [ad].[answer] = @answer
        AND [ad].[audit_id] = @audit_id;";
    }
}

```

```

        parameters.Add("answer", Answers.NOK);
        parameters.Add("audit_id", audit.Id);
    }
    DatabaseUtils.ExecuteNonQuery(query, parameters);
}

public static bool UserInAuditorList(string auditorLogin, string
auditType, string auditTarget)
{
    string query = @"SELECT 0";
    var parameters = new Dictionary<string, object>();
    switch (auditType)
    {
        case Audit5sLpaProduction:
            query = @"SELECT COUNT(*)
                FROM
                (
                    SELECT [auditor_login]
                    FROM [setts_auditors]
                    WHERE [audit_type] = @audit_type

                    UNION

                    SELECT [supervisor_login]
                    FROM [setts_audit_targets]
                    WHERE [audit_type] = @audit_type AND
[audit_target] = @audit_target
                ) [data]
                WHERE [auditor_login] = @auditor_login";
            parameters.Add("audit_type", auditType);
            parameters.Add("audit_target", auditTarget);
            parameters.Add("auditor_login", auditorLogin);
            break;

        default:
            query = @"SELECT COUNT(*)
                FROM [setts_auditors]
                WHERE [audit_type] = @audit_type
                    AND [auditor_login] = @auditor_login";
            parameters.Add("audit_type", auditType);
            parameters.Add("auditor_login", auditorLogin);
            break;
    }

    var result =
Utils.ConvertToNullableInt(DatabaseUtils.ExecuteScalarSelectQuery(query,
parameters));

```

```

        return (result != null && result > 0);
    }

    public static XtraReport GetXtraReport(Audit audit, string lang)
    {
        return new PdfReport(audit.Id, lang);
    }

    public static string GetTargetSupervisorLogin(string auditType, string
auditTarget)
    {
        string query = "SELECT [supervisor_login] FROM [setts_audit_targets]
WHERE [audit_type] = @audit_type AND audit_target = @audit_target";
        var parameters = new Dictionary<string, object>() { { "audit_type",
auditType }, { "audit_target", auditTarget } };
        var result =
Utils.ConvertToTrimmedString(DatabaseUtils.ExecuteScalarSelectQuery(query,
parameters));
        return result;
    }

    public static void NotifySupervisor(Audit audit)
    {
        if (audit.Type == Audit5sAdministration || audit.Type ==
Audit5sIpaProduction)
        {
            var supervisorLogin = GetTargetSupervisorLogin(audit.Type,
audit.Target);
            var fileName =
$"Audit_ID{audit.Id}_{DateTime.Now.ToString("yyyyMMdd")}.pdf";
            var subject = $"Ended audit {audit.TypeName} | {audit.Target}";
            var body = "";
            var email = AppUser.GetUserMail(supervisorLogin);
            if (email == null)
            {
                throw new Exception("You can not send an e-mail with the
report - no e-mail in the system / wrong format!");
            }
            var report = GetXtraReport(audit, Languages.Default);
            using (MemoryStream stream = new MemoryStream())
            {
                report.ExportToPdf(stream);
                stream.Position = 0;
                var attach = new Attachment(stream, fileName,
"application/pdf");

```

```

        MailUtils.SendEmail(subject, body, new List<string> { email
    }, new List<Attachment> { attach });
    }
}

public static string GetAuditGridTitle(Audit audit)
{
    var result = $"{s}{ audit.TypeName }{e} | {s}{audit.Target}{e} |
Auditor: {s}{audit.AuditorFullName}{e} | Progress:
{s}{audit.AuditProgress}%{e}";
    if (audit.Type == Audit5sAdministration || audit.Type ==
Audit5sLpaProduction)
    {
        result = $"{s}{ audit.TypeName }{e} | {s}{audit.Target}{e} |
Auditor: {s}{audit.AuditorFullName}{e} | Progress: {s}{audit.AuditProgress}%{e}
| Max 45 min";
    }
    return result;
}

public static string GetAuditGridCaption(string groupPosition, string
groupName)
{
    return $"{s}{groupPosition}. {groupName}{e}";
}

private const string s = "<span style=\"color: #FFD500; font-weight:
bold;\">>"; //s - html span start
private const string e = "</span>"; //e - html span end

public static string VerifyAuditTargetEntry(string newAuditType, string
newArea, string newSubarea, string newSection, string newSupervisorLogin)
{
    var message = "Error!";
    if (newAuditType == Audit5sAdministration || newAuditType ==
Audit5sLpaProduction)
    {
        if (newArea == null
            || newSubarea == null
            || newSupervisorLogin == null)
        {
            message = "Complete the AREA, SUBAREA and MANAGER fields!";
        }
        else
        {

```

```
        message = null;
    }
}
return message;
}
}
```

Кафедра КБПЗ – 2021 рік