

# КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

*бакалавр*

(назва освітнього ступеня)

на тему: Система моніторингу роботи програмного забезпечення в локальній мережі підприємства

Виконав: студент IV курсу, групи СІс-41  
спеціальності 123 «Комп'ютерна інженерія»

(шифр і назва спеціальності)

(підпис)

Шевчук М. С.

(прізвище та ініціали)

Керівник

(підпис)

Тим С.В.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Луцик Н.С.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Осухівська Г.М.

(прізвище та ініціали)

Рецензент

(підпис)

Дуда О.М.

(прізвище та ініціали)

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії  
(повна назва факультету)

Кафедра комп'ютерних систем та мереж  
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Осухівська Г.М.

(підпис)

(прізвище та ініціали)

«\_\_\_» \_\_\_\_\_ 2023 р.

**ЗАВДАННЯ**  
**НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня бакалавр  
(назва освітнього ступеня)

за спеціальністю 123 «Комп'ютерна інженерія»  
(шифр і назва спеціальності)

студента Шевчука Миколи Сергійовича  
(прізвище, ім'я, по батькові)

1. Тема роботи Система моніторингу роботи програмного забезпечення в локальній мережі підприємства

Керівник роботи кандидат технічних наук, доцент кафедри КС Тиш Євгенія Володимирівна  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «28» лютого 2023 року № 4/7-237

2. Термін подання студентом завершеної роботи 19.06.2023 р.

3. Вихідні дані до роботи Технічне завдання

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1. Аналіз технічного завдання

2. Проектна частина

3. Практична частина

4. Безпека життєдіяльності, основи охорони праці.

Висновки

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

1. Структурна схема системи

2. Діаграми варіантів використання

3. Діаграма розгортання системи

4. Результати роботи системи

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
<i>Безпека життєдіяльності, основи охорони праці</i>	<i>Пилипець М.І., д.т.н., професор кафедри МТ</i>		

7. Дата видачі завдання 02.03.2023

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	<i>Розробка технічного завдання</i>	<i>02.03-25.03.2023</i>	
2	<i>Аналіз технічного завдання</i>	<i>26.03-01.04.2023</i>	
3	<i>Аналіз вимог до системи моніторингу роботи ПЗ</i>	<i>02.04-16.04.2023</i>	
4	<i>Проектування структури системи</i>	<i>17.04-04.05.2023</i>	
5	<i>Проектування і реалізація програмної частини</i>	<i>05.05-31.05.2023</i>	
7	<i>Налаштування і тестування системи моніторингу ПЗ</i>	<i>01.06-06.06.2023</i>	
8	<i>Безпека життєдіяльності, основи охорони праці</i>	<i>07.06-08.06.2023</i>	
9	<i>Оформлення кваліфікаційної роботи</i>	<i>05.06-12.06.2023</i>	
10	<i>Попередній захист кваліфікаційної роботи</i>	<i>12.06-16.06.2023</i>	
11	<i>Захист кваліфікаційної роботи</i>	<i>19.06-24.06.2023</i>	

Студент

\_\_\_\_\_ (підпис)

*Шевчук Микола Сергійович*

\_\_\_\_\_ (прізвище та ініціали)

Керівник роботи

\_\_\_\_\_ (підпис)

*Тиш Євгенія Володимирівна*

\_\_\_\_\_ (прізвище та ініціали)

## АНОТАЦІЯ

Система моніторингу роботи програмного забезпечення в локальній мережі підприємства // Кваліфікаційна робота на здобуття освітнього ступеня бакалавр // Шевчук Микола Сергійович // ТНТУ, спеціальність 123 «Комп'ютерна інженерія» // Тернопіль, 2023 // с.– 57, рис. – 27 , табл. – 10, аркушів А1 – 4, бібліогр. – 16.

Ключові слова: моніторинг, клієнт, локальна мережа, адміністрування протокол.

У даній кваліфікаційній роботі виконано розробку системи для моніторингу роботи локальної мережі підприємства.

Пояснювальна записка містить 4 розділи.

В першому розділі здійснюється аналіз предметної області. Проведений огляд існуючих рішень показав функціональні можливості систем моніторингу. На основі аналізу визначені задачі проєктованої системи моніторингу ПЗ.

В другому розділі розроблено структуру системи. Здійснене логічне і концептуальне проєктування системи

В третьому розділі наведені результати роботи системи.

В четвертому розділі розглянуті питання охорони праці та вимоги з техніки безпеки.

## ABSTRACT

Software monitoring system in the enterprise local network // Qualification work for the bachelor's degree // Shevchuk Mykola Serhiyovych // TNTU, specialty 123 "Computer engineering" // Ternopil, 2023 // p.– 57, fig. – 27, tab. - 10, sheets A1 - 4, bibliography. - 16.

Keywords: monitoring, client, local network, protocol administration.

In this qualification work, the development of a system for monitoring the operation of the enterprise's local network was carried out.

The explanatory note contains 4 sections.

In the first section, the analysis of the subject area is carried out. The review of existing solutions showed the functionality of monitoring systems. Based on the analysis, the tasks of the designed software monitoring system were defined.

In the second chapter, the structure of the system is developed. The logical and conceptual design of the system was carried out

The third section presents the results of the system.

The fourth chapter deals with occupational health and safety requirements..

## ЗМІСТ

ВСТУП	.....	9
РОЗДІЛ 1	АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ	11
1.1	Огляд предметної області	11
1.2	Огляд існуючих рішень	12
1.3	Вимоги до системи моніторингу	18
1.4	Постановка задач кваліфікаційної роботи	18
РОЗДІЛ 2	ПРОЄКТНА ЧАСТИНА	20
2.1	Розробка структури системи	20
2.2	Діаграми класів	26
2.3	Діаграма розгортання	29
2.4	База даних	30
2.4.1	Концептуальне проектування	30
2.4.2	Логічне проектування	31
РОЗДІЛ 3	ПРАКТИЧНА ЧАСТИНА	38
3.1	Алгоритм авторизації користувача	38
3.2	Алгоритм роботи Агента	39
3.3	Тестування роботи	42
3.4	Висновок	48
РОЗДІЛ 4	БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	50
4.1	Санітарно-гігієнічні вимоги до умов праці	50

					КС КРБ 123.364.00.00 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Шевчук М. С.			Літ.	Арк.	Аркуші
Перевір.		Тиш Є.В.			6		
Реценз.		Дуда О.М.			ТНТУ, каф. КС, гр. СІс-41		
Н. Контр.		Луцик Н.С.					
Затверд.		Осухівська Г.М.					
Система моніторингу роботи програмного забезпечення в локальній мережі підприємства							

4.2	Надзвичайні ситуації: визначення причини, класифікація. ....	52
	ВИСНОВКИ.....	55
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	56
	Додаток А. Технічне завдання.....	58
	Додаток Б. Лістинги програмного забезпечення .....	72

					<i>КС КРБ 123.364.00.00 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

## СПИСОК СКОРОЧЕНЬ

ER - Entity-relationship;

MIB - Management Information Base;

SNMP - Simple Network Management Protocol;

БД - Бази даних;

ОС (OS) – операційна система;

ПЗ - програмне забезпечення;

ПК - персональний комп'ютер.

					<i>КС КРБ 123.364.00.00 ПЗ</i>	Арк.
						8
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		



## ВСТУП

У сучасному світі програмне забезпечення відіграє важливу роль у роботі підприємств. Встановлення та використання програм зазвичай супроводжується ліцензійними угодами, які регулюють права та обмеження щодо використання цих програм. Однак, через недбалість або незнання користувачів, часто виникають ситуації, коли умови ліцензійних угод порушуються, що може призвести до юридичних проблем та штрафних санкцій.

Для підприємств важливо мати контроль над використанням програмного забезпечення в своїй локальній мережі. Це допомагає забезпечити відповідність ліцензійним угодам, уникнути порушення авторських прав та мінімізувати ризики юридичних санкцій.

Створення системи моніторингу роботи програмного забезпечення в локальній мережі підприємства дозволяє отримати наступні переваги:

1) Контроль за використанням ліцензій. Система дозволяє оперативно виявляти, які програми встановлені на комп'ютерах користувачів та якими ліцензіями вони володіють. Це дає змогу перевірити, чи використовуються програми в межах ліцензійних угод та визначити, чи потрібно додатково придбати ліцензії.

2) Оптимізація витрат. Шляхом моніторингу програмного забезпечення можна виявити, які програми не використовуються або використовуються рідко, що дозволяє підприємству зекономити кошти на непотрібних ліцензіях або планувати раціональне використання програм.

3) Підвищення безпеки. Моніторинг допомагає виявити некоректне використання програмного забезпечення, таке як встановлення піратських копій або використання нелегальних ключів активації. Це дозволяє запобігти поширенню шкідливих програм та забезпечити високий рівень безпеки мережі.

4) Планування оновлень та підтримки. Збір інформації про версії

					<b>КС КРБ 123.364.00.00 ПЗ</b>	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

встановлених програм дозволяє підприємству планувати оновлення та підтримку програмного забезпечення. Це допомагає забезпечити актуальність програм, враховувати нові функціональні можливості та виправляти вразливості безпеки.

Створення системи моніторингу роботи програмного забезпечення в локальній мережі підприємства є важливим етапом у забезпеченні відповідності ліцензійним угодам, зменшенні юридичних ризиків та оптимізації використання програм.

Для вирішення поставленого завдання розробимо інформаційну систему онлайн моніторингу програмного забезпечення, встановленого на комп'ютерах організації. Вона необхідна для:

- 1) Забезпечення простого доступу до звітів про використання програм та відповідність ліцензій;
- 2) Інформації про придбання програм, типи ліцензії, їх кількість та їх серійні номери;
- 3) Мінімізації юридичних ризиків під час використання програм.

					<b>КС КРБ 123.364.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

## РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ

### 1.1 Огляд предметної області

Моніторинг мережі та програмного забезпечення є важливою складовою інфраструктури будь-якої організації. Він дозволяє відстежувати працездатність мережі, серверів, комп'ютерів та інших пристроїв, а також виявляти проблеми, збої та загрози безпеці.

Для моніторингу мережі можна використовувати різноманітні інструменти, такі як:

1) Системи моніторингу мережі (Network Monitoring Systems): Наприклад, Nagios, Zabbix, PRTG Network Monitor тощо. Вони дозволяють відстежувати стан мережевих пристроїв, моніторити пропускну здатність, діагностувати проблеми з підключенням та забезпечувати реал-тайм сповіщення про відмови.

2) Системи моніторингу серверів (Server Monitoring Systems): Наприклад, Zabbix, Nagios, Datadog, New Relic тощо. Вони слідкують за роботою серверів, моніторять процесорне навантаження, використання пам'яті, обсяг дискового простору, стан сервісів та інші параметри, які можуть вказувати на проблеми або загрози безпеці.

3) Системи моніторингу додатків (Application Monitoring Systems): Наприклад, New Relic, AppDynamics, Dynatrace [1] тощо. Вони дозволяють слідкувати за працездатністю додатків, моніторити час відгуку, завантаженість серверів, виявляти проблеми з продуктивністю та виконанням запитів.

4) Моніторинг безпеки (Security Monitoring): Використовуються

					<b>КС КРБ 123.364.00.00 ПЗ</b>			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		<i>Шевчук М. С.</i>			<i>Аналіз технічного завдання</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Перевір.</i>		<i>Тиш Є.В.</i>					11	
<i>Реценз.</i>		<i>Дуда О.М.</i>				<i>ТНТУ, каф. КС, гр. СІс-41</i>		
<i>Н. Контр.</i>		<i>Луцик Н.С.</i>						
<i>Затверд.</i>		<i>Осухівська Г.М.</i>						

системи, які дозволяють виявляти загрози безпеці, моніторити незвичайну активність, виявляти вторгнення, шкідливі програми та інші загрози. Деякі з відомих систем моніторингу безпеки включають SIEM (Security Information and Event Management) рішення, такі як Splunk, ArcSight, IBM QRadar тощо.

Загалом моніторинг програмного забезпечення вимагає використання спеціалізованих інструментів, які допомагають відстежувати працездатність та безпеку інфраструктури, а також оперативно реагувати на проблеми та загрози.

Зараз у мережі є безліч інформаційних систем з широким спектром функціоналу. У всіх у них є ряд рис для надання якісної та стабільної роботи, що підвищує її ефективність:

- автоматизованість;
- з інформуванням;
- вся інформація має бути короткою, точною;
- деталізований графічний інтерфейс;
- зворотним зв'язком;
- виведенням інформації із системи та її введення в систему.

## 1.2 Огляд існуючих рішень

Для аналізу предметної області розглянемо низку аналогічних рішень, знайдених у мережі.

New Relic [2] є однією з провідних систем моніторингу додатків та інфраструктури. Вона надає комплексні інструменти для відстежування продуктивності додатків, серверів, баз даних та інших компонентів інфраструктури. Ось деякі характеристики, переваги та недоліки New Relic:

					<b>КС КРБ 123.364.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

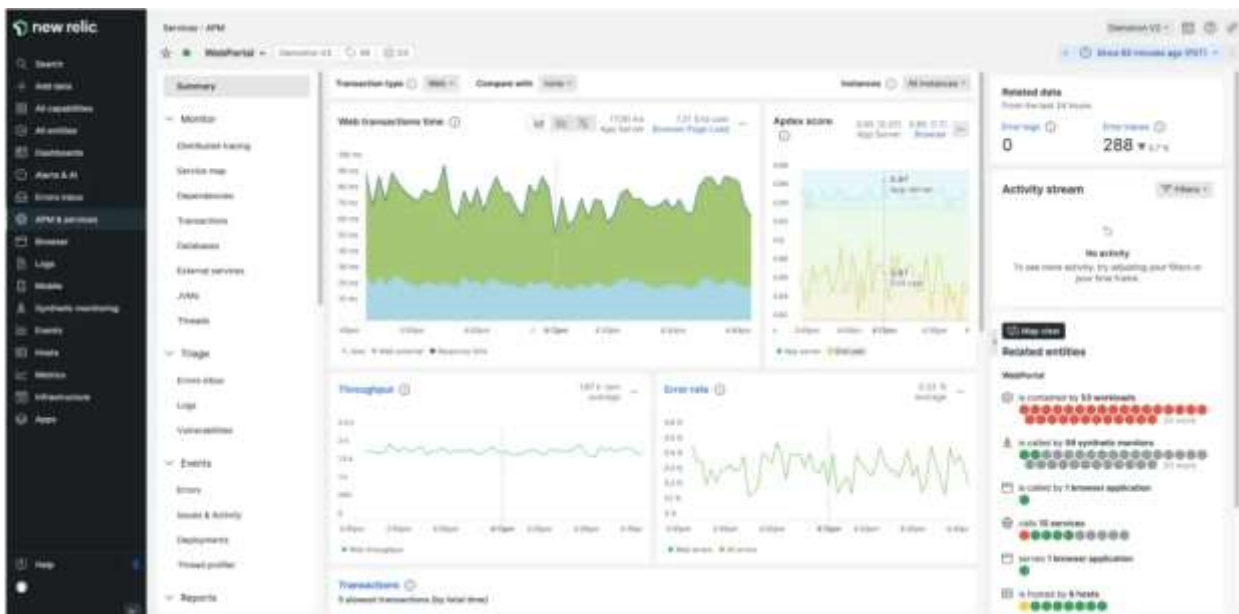


Рисунок 1.1 – Інтерфейс New Relic

### Характеристики:

**Моніторинг додатків.** New Relic надає докладний перегляд продуктивності додатків, включаючи швидкість відгуку, завантаження сервера, час виконання запитів, стек викликів та інші метрики, які допомагають виявити проблеми продуктивності.

**Моніторинг інфраструктури.** New Relic дозволяє відстежувати стан серверів, мережі, баз даних та інших компонентів інфраструктури, що допомагає виявляти загрози та проблеми з продуктивністю.

**Аналіз та діагностика.** New Relic надає інструменти для аналізу проблем продуктивності, включаючи глибоке дослідження коду, аналіз стеку викликів, трасування та інші функції, які допомагають вирішувати проблеми ефективно.

**Автоматичне масштабування.** New Relic може динамічно масштабувати ресурси на основі потреб додатків, що допомагає забезпечити оптимальну продуктивність.

### Переваги:

1) **Легкість використання:** New Relic має дружній інтерфейс та простий процес налаштування, що дозволяє швидко почати моніторинг.

2) Розширені можливості моніторингу: Вона забезпечує широкий спектр метрик та інструментів моніторингу для детального аналізу продуктивності.

3) Інтеграція з іншими інструментами: New Relic підтримує інтеграцію з багатьма іншими інструментами та платформами, такими як Slack, Jira, AWS, Azure тощо.

4) Автоматичне сповіщення: New Relic надає можливість налаштувати сповіщення у реальному часі про проблеми або незвичайну активність, допомагаючи оперативно реагувати на проблеми.

Недоліки:

1) Вартість: Використання New Relic може бути витратним, особливо для малих підприємств або стартапів.

2) Обмеження планів: Деякі функції та можливості доступні лише у платних планах, що може бути обмеженням для деяких організацій.

3) Залежність від інтернет-з'єднання: Для використання New Relic потрібне стабільне інтернет-з'єднання, інакше моніторинг може бути обмеженим або недоступним.

Загалом, New Relic є потужним інструментом для моніторингу продуктивності додатків та інфраструктури з багатьма перевагами. Однак, перед вибором системи моніторингу слід розглянути вимоги організації та порівняти їх з можливостями та вартістю New Relic.

AppDynamics [3] - це програмний продукт, розроблений для моніторингу та управління продуктивністю програмного забезпечення та додатків. Він дозволяє компаніям виявляти, діагностувати та вирішувати проблеми з продуктивністю в реальному часі, забезпечуючи швидке реагування на проблеми перед тим, як вони вплинуть на користувачів.

					<b>КС КРБ 123.364.00.00 ПЗ</b>	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 1.2 – Моніторинг з допомогою AppDynamics

Ось деякі переваги AppDynamics:

- 1) Моніторинг в реальному часі: AppDynamics забезпечує відстеження та аналіз роботи додатків у реальному часі, що дозволяє оперативно виявляти проблеми з продуктивністю та негайно реагувати на них.
- 2) Кореляція бізнес-метрик: Програма не лише надає технічну інформацію про продуктивність, але також дозволяє корелювати ці дані з бізнес-метриками. Це дозволяє компаніям зрозуміти, як проблеми з продуктивністю впливають на їхні бізнес-показники.
- 3) Візуалізація: AppDynamics надає графічні зображення та діаграми, що допомагають візуалізувати продуктивність додатків. Це дозволяє операторам легко аналізувати дані та шукати проблеми.
- 4) Автоматизоване масштабування: AppDynamics може автоматично реагувати на збільшення навантаження на додаток і масштабувати його, щоб забезпечити стабільну продуктивність.

Незважаючи на переваги, AppDynamics також має кілька недоліків:

- 1) Висока вартість: Ліцензування та використання AppDynamics може бути досить дорогим для невеликих компаній або стартапів.

2) Складність налаштування: Початкове налаштування та конфігурація AppDynamics можуть бути складними, особливо для користувачів без попереднього досвіду в цій області.

3) Обмежена підтримка мов: AppDynamics краще працює з певними мовами програмування, такими як Java або .NET, і може бути менш ефективним для інших мов програмування.

4) Залежність від агентів: Для збору даних про продуктивність додатків, AppDynamics вимагає встановлення агентів на сервери. Це може бути проблемою для деяких компаній з охороною даних або вимогами до безпеки.

Dynatrace [4] - це інтелектуальна платформа моніторингу та управління продуктивністю, яка допомагає компаніям виявляти, діагностувати та вирішувати проблеми з продуктивністю їхніх додатків та інфраструктури.

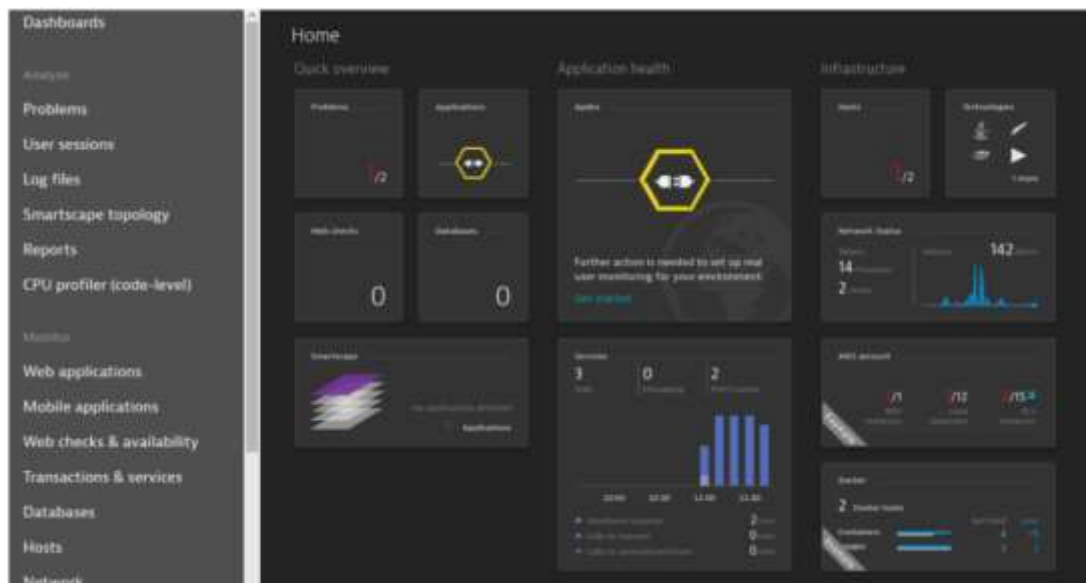


Рисунок 1.3 - Платформа моніторингу та управління продуктивністю Dynatrace

Переваги Dynatrace:

1) Повна видимість та автоматичне виявлення проблем: Dynatrace надає повну видимість стеку додатків та інфраструктури, що дозволяє



операторам відстежувати продуктивність в реальному часі. Платформа автоматично виявляє проблеми та навіть надає прогнози щодо майбутніх проблем.

2) Штучний інтелект та автоматизація: Dynatrace використовує штучний інтелект для аналізу та інтерпретації даних про продуктивність. Він автоматично аналізує велику кількість даних та надає рекомендації щодо оптимізації продуктивності додатків.

3) Широкий охоплення технологій: Dynatrace підтримує різні технології, включаючи контейнери, хмарні сервіси, мікросервіси, бази даних та інші. Це робить його універсальним інструментом для моніторингу та управління продуктивністю.

4) Візуалізація та аналітика: Dynatrace надає розширені можливості візуалізації та аналітики даних. Інтерфейс платформи є зручним та інтуїтивно зрозумілим, а аналітичні звіти та графіки допомагають зрозуміти проблеми та приймати ефективні рішення.

#### Недоліки Dynatrace:

1) Висока вартість: Dynatrace є досить дорогим рішенням, особливо для менших компаній або стартапів.

2) Складність налаштування: Налаштування Dynatrace може бути складним завданням, особливо для користувачів без досвіду в області моніторингу продуктивності.

3) Залежність від агентів: Dynatrace також потребує встановлення агентів на серверах для збору даних про продуктивність. Це може бути проблемою для компаній з обмеженими вимогами до безпеки.

4) Обмеження на кількість моніторингованих об'єктів: Деякі версії Dynatrace можуть мати обмеження на кількість моніторингованих об'єктів, що може бути проблемою для великих підприємств або компаній зі складною інфраструктурою.

					<b>КС КРБ 123.364.00.00 ПЗ</b>	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

Необхідно враховувати ці переваги та недоліки Dynatrace при виборі платформи моніторингу та управління продуктивністю для конкретних потреб компанії.

### 1.3 Вимоги до системи моніторингу

Програма є інформаційною системою, створеною для збору інформації з комп'ютерів, які підключені до локальної мережі. Для отримання інформації з робочих станцій передбачено використання вбудованих агентів, які збирають інформацію. Зокрема це:

- Ім'я комп'ютера;
- Характеристика апаратного забезпечення;
- Перелік програм і їх ліцензії;
- Відхилення в шаблонах (ряд програм, які відповідають за раніше заданим).

Ці отримані відомості надходять адміністратору на екран у вигляді таблиць на веб сторінці або в форматі електронних таблиць Excel. Надалі адміністратор може використовувати весь отриманий матеріал для:

- Складання звіту про стан ліцензійного забезпечення організації для подальшого продовження строків дії ліцензійної угоди;
- Пошуку необхідних програм на комп'ютерах, якщо така є;
- Пошуку комп'ютера, що підходить для встановлення програмного забезпечення за мінімальними системними вимогами.

### 1.4 Постановка задач кваліфікаційної роботи

Для вирішення поставленого завдання розробки інформаційної системи для онлайн моніторингу програмного забезпечення, встановленого на комп'ютерах організації, ми можемо виділити певні елементи з наявних

					<b>КС КРБ 123.364.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

аналогічних рішень. Враховуючи ці елементи, наша система буде мати наступні особливості:

1) Сучасний інтерфейс. Система буде мати зручний та інтуїтивно зрозумілий інтерфейс, який дозволить користувачам легко отримати доступ до необхідної інформації про комп'ютери та програмне забезпечення.

2) Функціональність. Забезпечимо нашу систему всіма необхідними функціями, включаючи перевірку програмного забезпечення на комп'ютерах організації, інформацію про стан ліцензійних продуктів та інші потрібні дані.

3) Стабільна робота. Система буде розроблена з урахуванням стабільної та надійної роботи, щоб забезпечити безперебійний доступ до інформації та виконання необхідних завдань.

4) Мінімізація витрат. Ми плануємо розробити систему з урахуванням ефективного використання ресурсів та оптимізації витрат на розробку, щоб забезпечити економічно вигідне рішення.

5) Надання детальної інформації. Система буде здатна надавати користувачам детальну інформацію про комплектуючі комп'ютерів, програмне забезпечення та строки дії ліцензійних угод, щоб дати їм повний обсяг необхідних даних.

6) Легка установка. Ми спроектуємо систему таким чином, щоб її встановлення та налаштування були якомога простішими та зручними для користувачів.

Загальний результат нашої системи буде включати простий доступ до інформації, зручне користування, економічну ефективність, користувацьку дружельюбність та надання конкретної інформації на запит користувача.

					<b>КС КРБ 123.364.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

## РОЗДІЛ 2 ПРОЄКТНА ЧАСТИНА

Для досягнення поставленого завдання потрібно наступне програмне забезпечення:

- 1) MySQL, база даних, в якій зберігатиметься інформація про цю мережу;
- 2) Мова програмування PHP з розширенням SNMP, для простого та зручного збору даних комп'ютерів у мережі організації;
- 3) Використовувати CSS [5], формальну мову програмування для опису зовнішнього вигляду нашої роботи;
- 4) Середовище розробки Eclipse PHP, з плагіном CKEditor.

### 2.1 Розробка структури системи

Структурна схема програми моніторингу програмного забезпечення на комп'ютерах в локальній мережі. Ця структурна схема показує взаємозв'язок між різними компонентами системи моніторингу програмного забезпечення та шляхи передачі інформації між ними.

Ця структурна схема описує систему моніторингу програмного забезпечення, яка складається з трьох основних компонентів: збір інформації агентом, аналіз та обробка інформації та зберігання інформації.

Перший компонент - агент, встановлений на комп'ютерах користувачів, відповідає за збір інформації про встановлене програмне забезпечення на кожному комп'ютері. Агент може зчитувати дані про встановлені програми, їх версії, налаштування та інші важливі атрибути.

Після збору інформації агент надсилає ці дані через локальну мережу на центральний сервер де встановлена система моніторингу, що є другим

					<b>КС КРБ 123.364.00.00 ПЗ</b>			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		<i>Шевчук М. С.</i>			<b>Проектна частина</b>	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Перевір.</i>		<i>Тиш Є.В.</i>					20	
<i>Реценз.</i>		<i>Дуда О.М.</i>				<b>ТНТУ, каф. КС, гр. СІс-41</b>		
<i>Н. Контр.</i>		<i>Луцик Н.С.</i>						
<i>Затверд.</i>		<i>Осухівська Г.М.</i>						

компонентом системи. Система відповідає за аналіз та обробку інформації, отриманої від агентів. Процес аналізу та обробки може включати перевірку актуальності версій програмного забезпечення, виявлення проблем з ліцензіями ПЗ.

Отримана та оброблена інформація зберігається на локальному сервері, що є третім компонентом системи. Збережені дані можуть використовуватися для створення звітів, статистичного аналізу, моніторингу змін у програмному забезпеченні, а також для планування оновлень та забезпечення безпеки системи.

Загалом, система моніторингу програмного забезпечення забезпечує централізований збір, аналіз та збереження даних про встановлене програмне забезпечення на комп'ютерах користувачів, що дозволяє забезпечити ефективне управління програмними ресурсами, підтримувати актуальні версії ПЗ та забезпечувати безпеку системи.

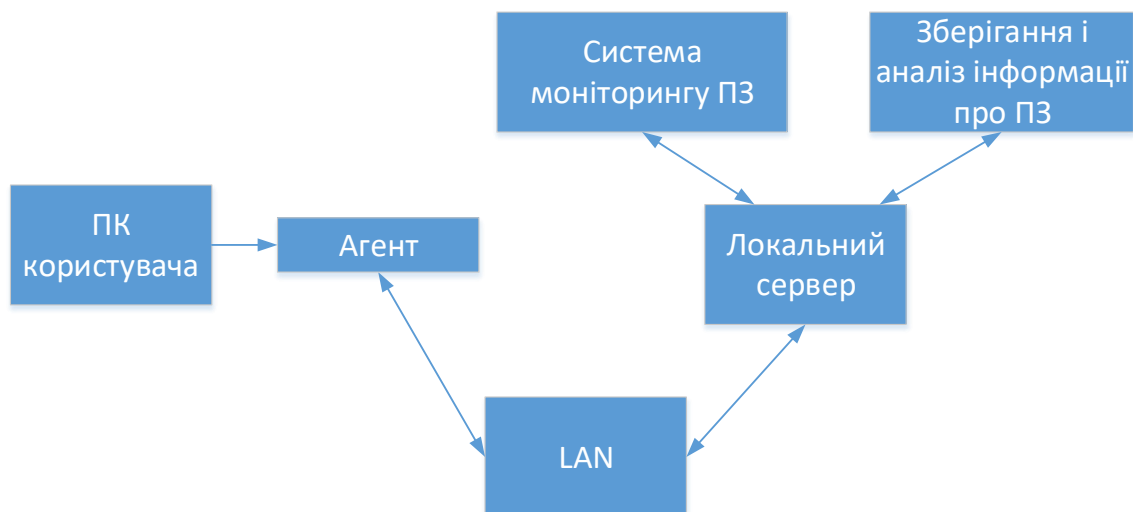


Рисунок 2.1 – Структурна схема системи моніторингу програмного забезпечення в локальній мережі

Для більш детального представлення функцій системи скористаємося діаграмами використання.

На рисунку 2.2 присутні два актори: «Адміністратор» і «Користувач». «Адміністратор» відповідає за керування системою моніторингу програмного забезпечення, тоді як «Користувач» є звичайним користувачем системи.

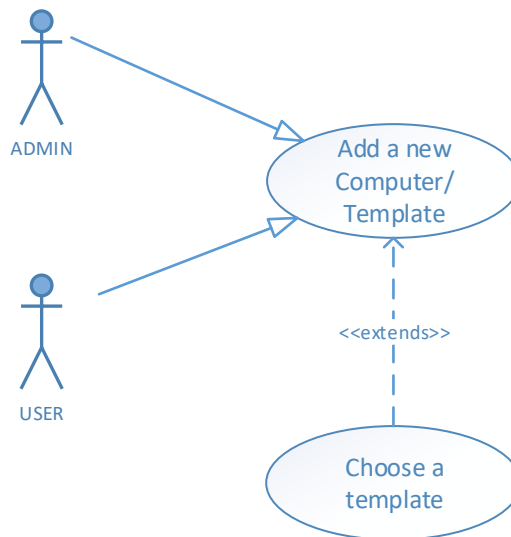


Рисунок 2.2 – Додавання нового Комп'ютера/Шаблону

У випадку додавання нового комп'ютера до системи, «Адміністратор» або «Користувач» можуть вибрати один з попередньо створених шаблонів або ввести новий шаблон. Шаблони пов'язані з комп'ютерами і містять інформацію про очікуване програмне забезпечення, яке має бути встановлене на комп'ютері.

Після вибору або створення шаблону, система починає перевірку на відповідність. Це означає, що система порівнює програмне забезпечення, встановлене на комп'ютері, з інформацією, що міститься в шаблоні. Якщо встановлене програмне забезпечення відповідає шаблону, система вважає, що комп'ютер знаходиться у «нормальному/робочому стані». Якщо є відхилення від шаблону, система може позначити комп'ютер як «неперевірений» або потребує уваги адміністратора.

Користувачі, окрім адміністратора, не можуть надавати повноцінну інформацію про нову програму. Тому, якщо відхилення від шаблону виникає

через звичайного користувача, система може сигналізувати адміністратору про невідповідність ПЗ, поки він не підтвердить або внесе необхідні зміни до шаблону.

Загалом, ця діаграма варіанта використання допомагає уявити процес додавання нового комп'ютера до системи моніторингу програмного забезпечення та вибір або створення шаблону для контролю за програмними засобами на ньому.

На рисунку 2.3 зображено процес встановлення нового програмного забезпечення в систему і вибір типу ліцензії для нього. Давайте розглянемо цю діаграму детальніше.

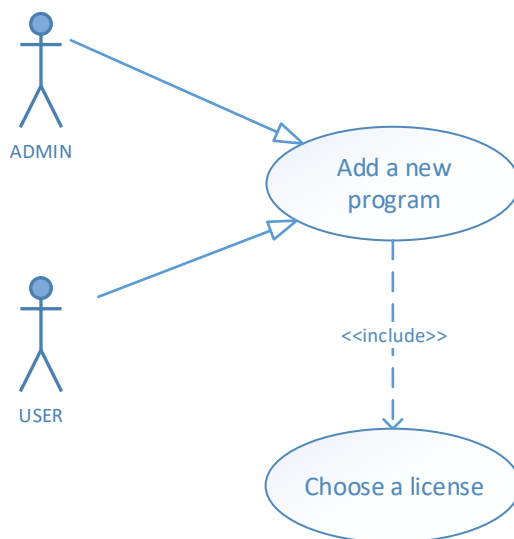


Рисунок 2.3 – Додавання нової програми до системи

Процес починається з вибору нового програмного забезпечення для встановлення. Після вибору, користувач або адміністратор мають можливість вибрати тип ліцензії з переліку, де доступні різні варіанти, такі як «Commercial», «Undefined», «Freeware» тощо.

Для прикладу, якщо обраний тип ліцензії - «Commercial», необхідне введення додаткової інформації про термін дії ліцензії, кількість комп'ютерів, на яких дозволено використовувати цю ліцензію, або тип поширення. Ця інформація дозволяє системі вести облік ліцензій та контролювати їх

використання на комп'ютерах мережі.

У разі, якщо обрано інший тип ліцензії, відповідні розділи про термін дії ліцензії, кількість комп'ютерів або тип поширення можуть бути пропущені, оскільки ці додаткові відомості не потрібні для таких типів ліцензій.

Ця діаграма варіанта використання дозволяє користувачам встановлювати нове програмне забезпечення і визначати тип ліцензії для нього, а також допомагає адміністраторам контролювати використання ліцензованих програм в мережі та вирішувати питання про продовження термінів ліцензій.

Діаграма на рисунку 2.3 показує процес збору інформації про комп'ютери, підключені до мережі організації агентом інформаційної системи. Цей агент використовує протокол SNMP (Simple Network Management Protocol), який є стандартним протоколом для управління пристроями в IP-мережах.

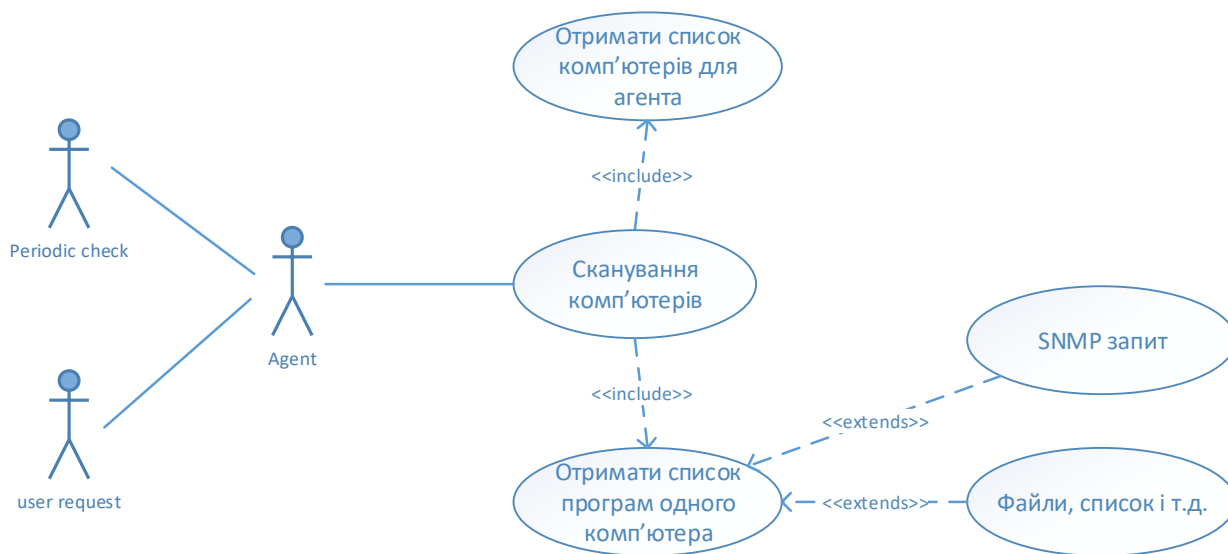


Рисунок 2 3 – Сканування комп'ютерів системи

Протокол SNMP дозволяє агенту отримати доступ до МІВ (Management Information Base) - це дерево, яке містить інформацію про пристрої і програмне забезпечення, що використовується в мережі. МІВ містить різноманітні параметри, такі як системна інформація, стан мережевих інтерфейсів,



конфігураційні дані і багато іншого.

Агент звертається до МІВ і отримує необхідну інформацію про комп'ютери, підключені до мережі організації. Після цього агент перетворює це МІВ-дерево на список або таблицю, де кожен рядок таблиці представляє окремий комп'ютер або відділення, а стовпці містять різні характеристики інформації про ці комп'ютери, такі як назва комп'ютера, IP-адреса, версія операційної системи, встановлені програми тощо.

Отримана таблиця потім може бути використана для надання інформації користувачам. Наприклад, адміністратор може звертатися до агента для отримання списку комп'ютерів і їх характеристик або використовувати цю інформацію для моніторингу, управління або інших цілей.

Збір інформації може відбуватися на запит користувача, або періодично по графіку, коли агент автоматично збирає та оновлює інформацію з МІВ-дерева.

На рисунку 2.4 показана діаграма, яка ілюструє процес додавання нової ліцензії користувачами інформаційної системи.

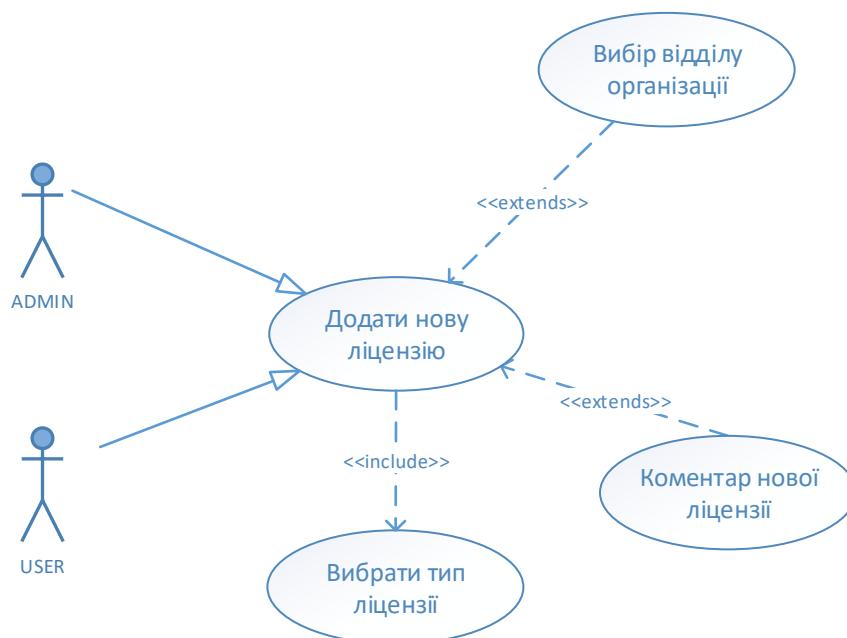


Рисунок 2.4 – Додавання нової ліцензії до системи

Цей процес дозволяє відстежувати, яке програмне забезпечення в

Змн.	Арк.	№ докум.	Підпис	Дата

організації має ліцензію і до якого відділення воно відноситься.

Користувачі вводять нову ліцензію із наступними деталями:

- 1) Тип ліцензії. Користувач обирає тип ліцензії «Commercial», «Open Source», «Freeware» тощо.
- 2) Відділення. Користувач вказує, до якого відділення організації відноситься ця ліцензія. Це може бути назва або ідентифікатор відділення.
- 3) Коментар. Користувач може додати коментар або опис до цієї ліцензії, наприклад, вказати для якого конкретного програмного забезпечення вона придбана або будь-яку іншу важливу інформацію.

Ці дані, введені користувачем, зберігаються в інформаційній системі і можуть бути використані для формування списку ліцензій, які є в організації. Цей список може бути використаний для внутрішньої інвентаризації, управління ліцензіями, а також для забезпечення відповідності з вимогами ліцензійних угод та ліцензійного програмного забезпечення.

## 2.2 Діаграми класів

Інформаційна система, що розробляється, складається з модулів, які є основою представлення у веб-додатку. Кожен з цих модулів представлений класами і виконує певні функції у веб-додатку. Давайте розглянемо ці модулі детальніше:

- 1) Модуль «License type». Класи в цьому модулі відповідають за надання інформації про типи ліцензій програмного забезпечення, які використовуються в системі організації. Наприклад, це можуть бути класи, які описують типи ліцензій (Commercial, Undefined, Freeware) та їх характеристики. Модуль може забезпечувати функціональність, пов'язану з додаванням, редагуванням або видаленням типів ліцензій, а також отриманням інформації про наявні типи ліцензій.

- 2) Модуль «Software\_Catalogue». Цей модуль містить інформацію

					<b>КС КРБ 123.364.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

про програми, які є доступними в організації. Він містить класи, які описують програми за допомогою різних атрибутів, таких як мінімальні системні вимоги, тип програми, шаблони встановлення програмного забезпечення, необхідний софт для встановлення програми тощо. Модуль може надавати функції для додавання нових програм до каталогу, редагування або видалення програм, а також отримання інформації про наявні програми.

Ці два модулі разом формують основу для представлення у веб-додатку інформації про типи ліцензій та програми, які доступні в організації. Вони можуть бути інтегровані з іншими модулями систем організації для забезпечення повноцінного функціоналу та задоволення потреб користувачів.

На діаграмі класів «License type», зображеній на рисунку 2.5, демонструються класи «License type» і «License catalog».

1) Клас «License type». Цей клас представляє типи ліцензій програмного забезпечення. У цьому класі визначені класи, які успадковуються від «License type» і представляють конкретні типи ліцензій, такі як «Undefined», «Freeware» та «Commercial». Кожен тип ліцензії може мати додаткові підтипи, які також визначені як класи або підкласи класу «License type».

2) Клас «License catalog». Цей клас представляє «Каталог ліцензій» і містить інформацію про кожну ліцензію. У цьому класі визначені атрибути, такі як «Дата покупки», «Ціна», «Апаратний ключ» та інші, які дозволяють зберігати деталі про кожну ліцензію.

					<b>КС КРБ 123.364.00.00 ПЗ</b>	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

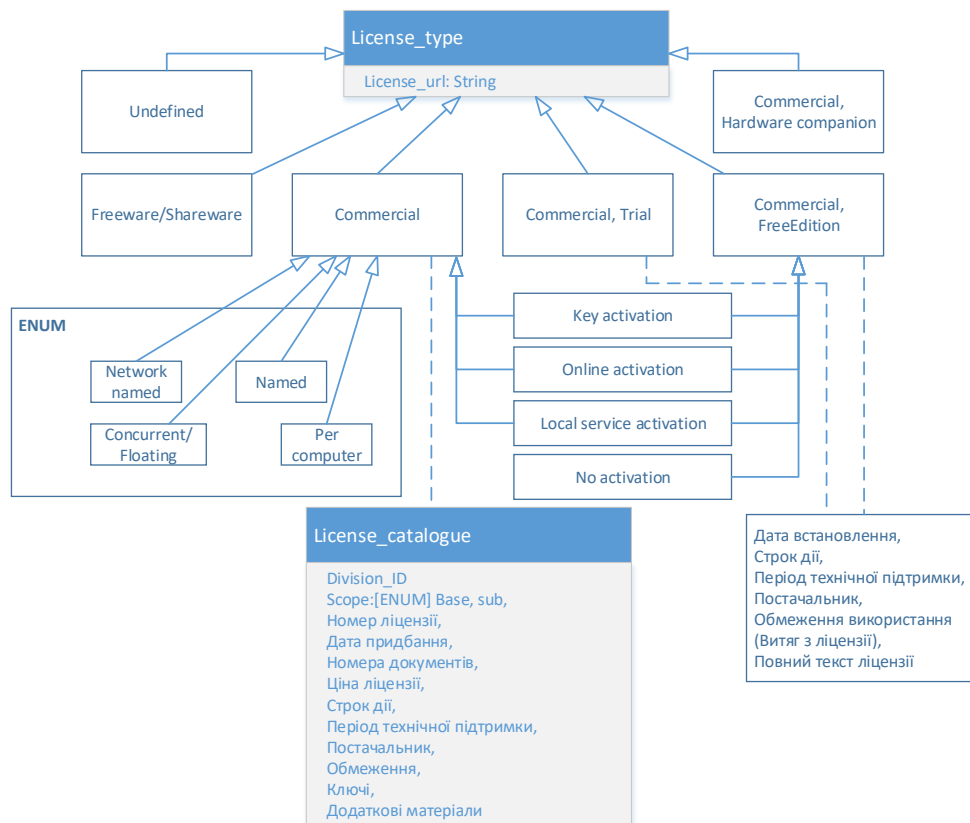


Рисунок 2.5 – Діаграма класів «License type»

На рисунку 2.6. наведено діаграму класів «Software\_Catalogue».

У класі «Software\_Catalogue» зберігаються атрибути: «Soft\_id», «Text\_pattern», «TitleForReport», «LicenseType», «SoftType».

Клас «Software\_Catalogue» має три залежності до інших класів: «Soft\_type», «Software Text\_id\_Type» та «Hardware Requirements». Це означає, що «Software\_Catalogue» використовує ці класи для отримання додаткової інформації про програми.

Клас «Soft\_type». У цьому класі зібрана інформація про типи додатків, що зберігаються в інформаційній системі, а також про права користувачів на їх використання.

Клас «Software Text\_id\_Type». У цьому класі зібрана інформація про те, як агент перевіряє списки шаблонів, по словах або повністю весь шаблон.

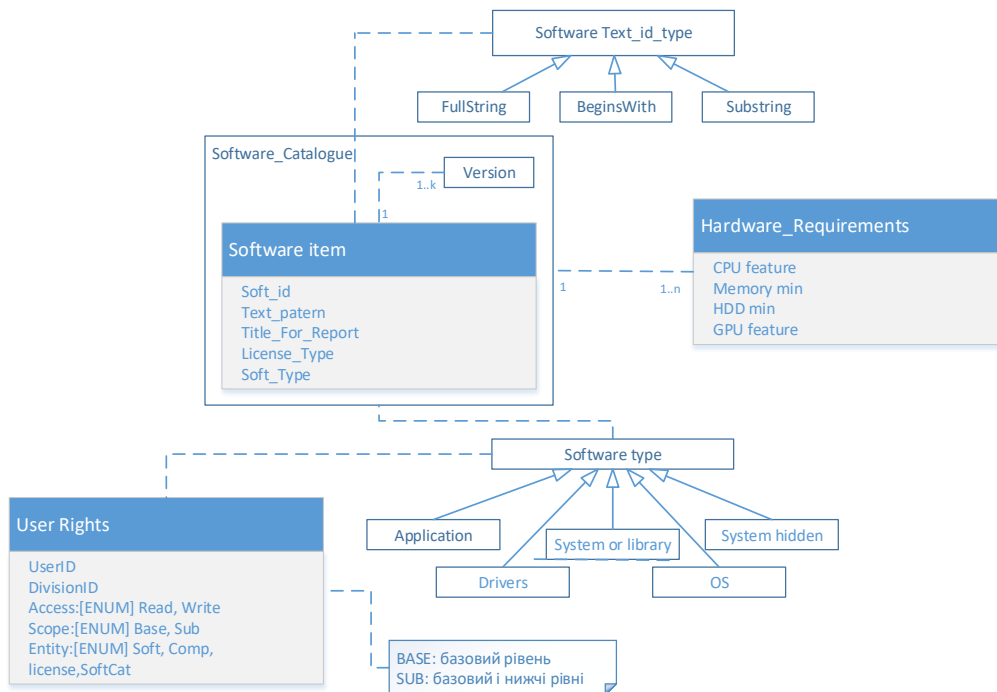


Рисунок 2.6 – Діаграма класів «Software\_Catalogue»

Клас «Hardware Requirements». У цьому класі зібрана інформація про мінімальні системні вимоги програм, яка збирається інформаційною системою.

### 2.3 Діаграма розгортання

Побудуємо діаграму розгортання для аналізу системи та її наступної модифікації(рис. 2.7).

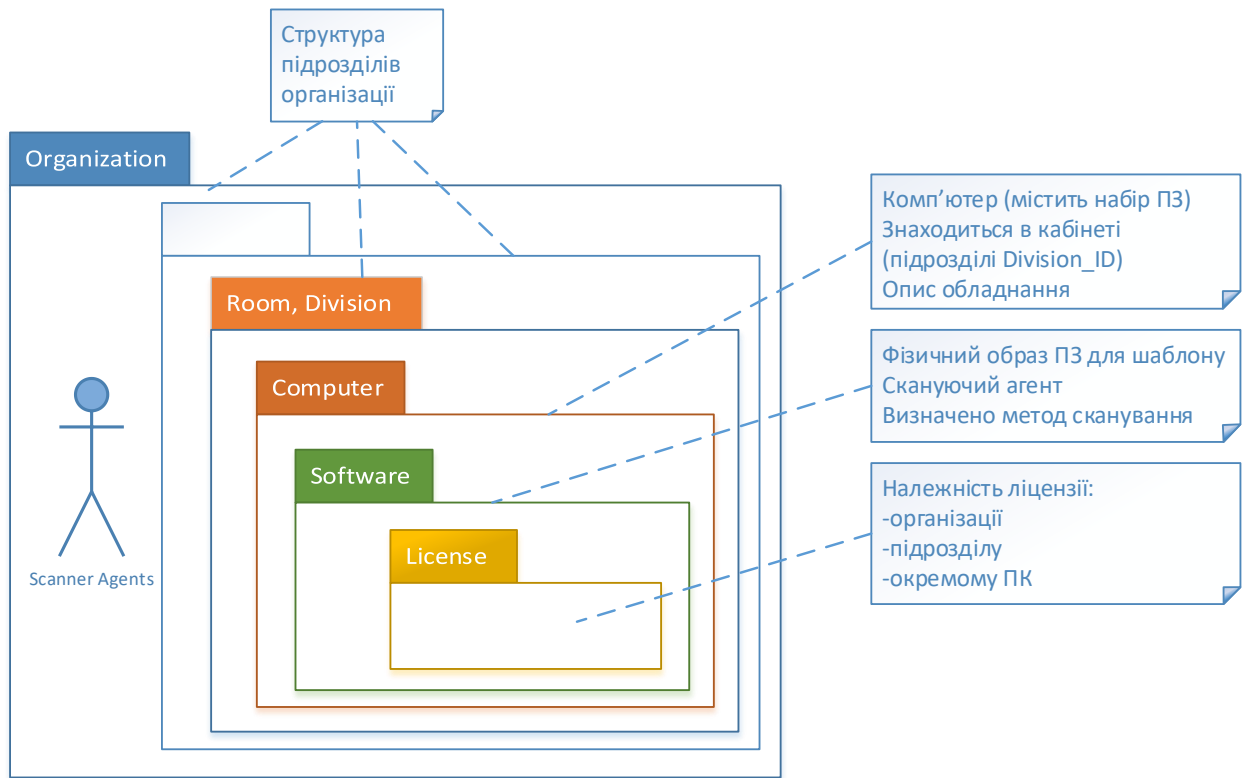


Рисунок 2.7 - Діаграма розгортання

На діаграмі описана структура організації, підрозділу та як ліцензії поширюються по всій компанії.

## 2.4 База даних

Для зберігання всієї інформації, зібраної з комп'ютерів, було побудовано базу даних. Бази даних (БД) - сукупність самостійних матеріалів, пов'язаних один з одним, так що матеріали були знайдені і оброблені. Для проектування БД було проведено концептуальне та логічне програмування.

### 2.4.1 Концептуальне проектування

Першим етапом розробки є концептуальне проектування, зробимо збір, аналіз та редагування вимог до даних.

При вході в систему визначаються права на адміністратора та користувача.

Користувач може побачити необхідну йому інформацію про ПЗ в системі організації або знайти комп'ютер, який задовольняє його вимогам до встановленого ПЗ.

Адміністратор, бачить стан комп'ютерів, терміни дії ліцензії, розбіжності у шаблонах. Заповнює інформацію про програми, може додавати нових користувачів, нові комп'ютери, нові програми, нові шаблони або здійснювати коригування старих.

Неавторизований користувач не зможе зайти в систему.

На основі цього було виділено 9 сутностей, 9 зв'язків типу «один до багатьох» і побудована ER- діаграма (рис. 2.8)

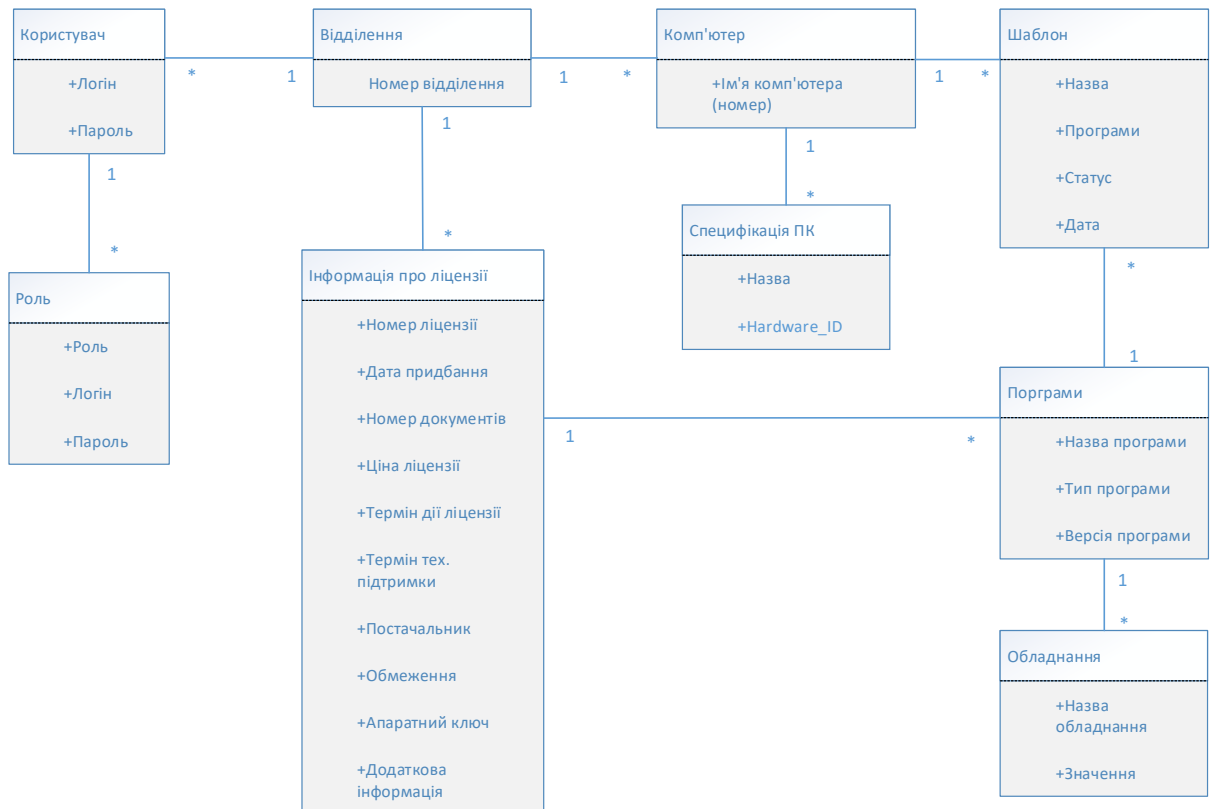


Рисунок 2.8 - ER- діаграма

#### 2.4.2 Логічне проектування

Другим етапом розробки є логічне проектування, тут було проведено переведення ER -моделі в реляційну форму, специфікуються обмеження та

правила підтримки цілісності на реляційному рівні (рис. 2.9)

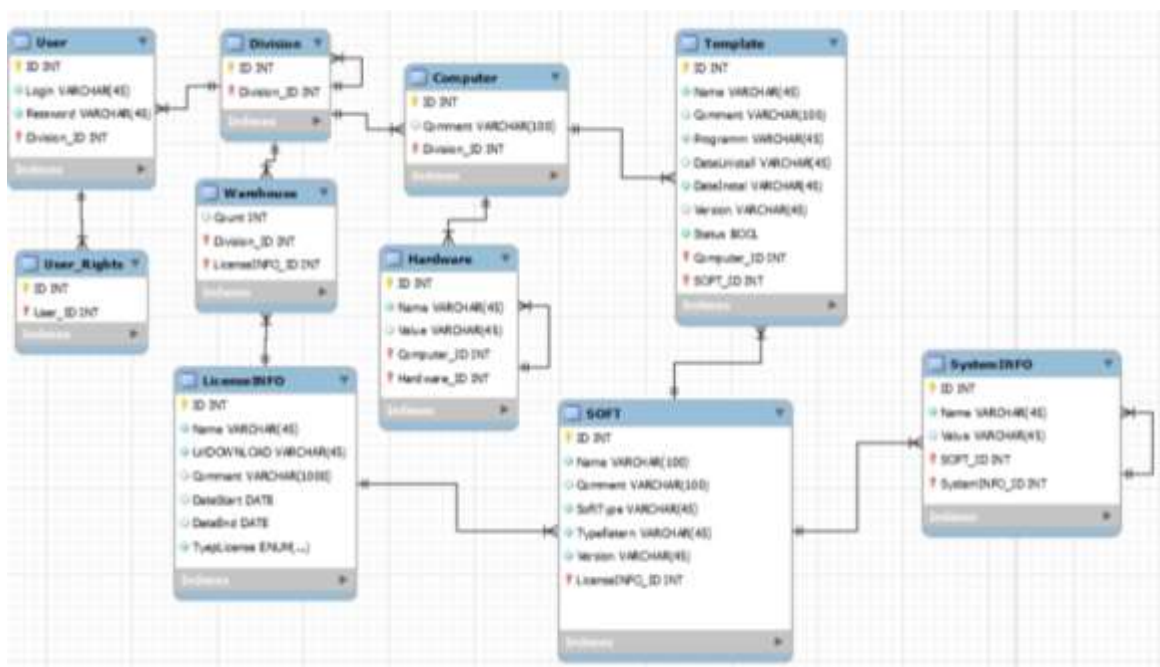


Рисунок 2.9 - Схема бази даних

У розробленій реляційній формі глобальної моделі використовують 10 таблиць. Розглянемо кожну їх докладніше.

Таблиця «Users». У цій таблиці буде зберігатися інформація про зареєстрованих користувачів системи.

Таблиця 2.1 - «Users»

Назва	Тип даних	Значення за замовчуванням	Обов'язковість	Первинний ключ	Зовнішній ключ
UserID	int	-	+	+	-
Login	varchar(45)	-	+	-	-
Password	varchar(45)	-	+	-	-
Division_ID	int	-	+	-	+



Таблиця «User\_Rights» зберігає інформацію про права користувачів.

Таблиця 2.2 - «User\_Rights»

Назва	Тип даних	Значення за замовчуванням	Обов'язковість	Первинний ключ	Зовнішній ключ
ID	int	-	+	+	-
User_ID	nvarchar(20)	-	+	-	+

Таблиця «Division». зберігає інформацію про організації та їх відділення.

Таблиця 2.3 - «Division»

Назва	Тип даних	Значення за замовчуванням	Обов'язковість	Первинний ключ	Зовнішній ключ
ID	int	-	+	+	-
Division_ID	int	-	+	-	+

Таблиця «Computer». Дана таблиця зберігає інформація про комп'ютери, в якому відділенні він розташовується і опис до нього.

Таблиця 2.4 - «Computer»

Назва	Тип даних	Значення за замовчуванням	Обов'язковість	Первинний ключ	Зовнішній ключ
ID	int	-	+	+	-
Comment	varchar(100)	-	-	-	-
DivisionID	int	-	+	-	+

Змн.	Арк.	№ докум.	Підпис	Дата

КС КРБ 123.364.00.00 ПЗ

Арк.

33

Таблиця «Warehouse». У цій таблиці зібрана вся інформація про кількість ліцензій та до якої про відокремлення вона приписана.

Таблиця 2.5 - «Warehouse»

Назва	Тип даних	Значення за замовчуванням	Обов'язковість	Первинний ключ	Зовнішній ключ
Count	int	-	-	-	-
Division_ID	Int	-	+	-	+
LicenseINFO_	int	-	+	-	+

Таблиця «LicenseINFO». У цій таблиці зберігається вся інформація про ліцензії, їх тип, дата старту та закінчення терміну дії.

Таблиця 2.6 - «LicenseINFO»

Назва	Тип даних	Значення за замовчуванням	Обов'язковість	Первинний ключ	Зовнішній ключ
ID	int	-	+	+	-
Name	varchar(45)	-	+	-	-
UrlDOWNLO	varchar(45)	-	+	-	-
Comment	varchar(1000)	-	-	-	-
DateStart	date	-	-	-	-
DateEnd	date	-	-	-	-
TypeLicense	enum()	+	+	-	-

Таблиця «Soft». У цій таблиці зібрана вся інформація про програми, встановлені в системі.

Таблиця 2.7 - «Soft»

Назва	Тип даних	Значення за замовчуванням	Обов'язковість	Первинний ключ	Зовнішній ключ
ID	Int	-	+	+	-
Name	varchar(100)	-	+	-	-
Comment	varchar(100)	-	-	-	-
SoftType	int	-	+	-	-
TypePatern	varchar(45)	-	+	-	-
Version	varchar(45)	-	+	-	-
LicenseINFO_I	int	-	+	-	+

Таблиця «Template». Ця таблиця зберігає матеріали за шаблонами, таку як ім'я шаблону, опис до нього, програми, складові його, дата встановлення та видалення програми з нього версію та статус.

Таблиця 2.8 - «Template»

Назва	Тип даних	Значення за замовчуванням	Обов'язковість	Первинний ключ	Зовнішній ключ
ID	int	-	+	+	-
Name	varchar(45)	-	+	-	-
Comment	varchar(100)	-	-	-	-
Programm	varchar(45)	-	+	-	-
DateUninstall	varchar(45)	-	-	-	-
DateInstal	varchar(45)	-	+	-	-
Version	varchar(45)	-	-	-	-
Status	bool	-	+	-	-
Computer_ID	int	-	+	-	+
Soft_ID	int	-	+	-	+

Змн.	Арк.	№ докум.	Підпис	Дата

КС КРБ 123.364.00.00 ПЗ

Арк.

35

Таблиця «SystemINFO». У таблиці зібрана інформація про необхідні мінімальні системні вимогах комп'ютера необхідні програмі.

Таблиця 2.9 - «SystemINFO»

Назва	Тип даних	Значення за замовчуванням	Обов'язковість	Первинний ключ	Зовнішній ключ
ID	int	-	+	+	-
Name	varchar(45)	-	+	-	-
Value	varchar(45)	-	-	-	-
SOFT_ID	int	-	+	-	+
SystemINFO_I	int	-	+	-	+

Таблиця «Hardware». У цій таблиці зберігається інформація про складові комп'ютера.

Таблиця 2.10 - «Hardware»

Назва	Тип даних	Значення за замовчуванням	Обов'язковість	Первинний ключ	Зовнішній ключ
ID	int	-	+	+	-
Name	nvarchar(20)	-	+	-	-
Value	nvarchar(20)	-	-	-	-
Computer_ID	int	-	+	-	+
Hardware_ID	int	-	+	-	+

В даному розділі проведено проектування інформаційної системи. За допомогою діаграм була описана архітектура розроблюваної системи.

Структура системи дозволяє визначити загальну організацію системи,

включаючи модулі та компоненти, їх взаємозв'язки та ієрархію.

Діаграма прецедентів (використання) вказує на основні функції та можливості системи, а також на акторів, що взаємодіють з системою. Це дозволяє уявити, як користувачі будуть використовувати систему та які результати вони очікують.

Діаграми класів відображають структуру класів системи, їх атрибути та взаємозв'язки. Вони допомагають зрозуміти, які класи існують у системі та як вони взаємодіють між собою.

Діаграма розгортання показує фізичну конфігурацію апаратного та програмного забезпечення системи, а також взаємозв'язки між ними. Вона дозволяє зрозуміти, як система буде розгортатися на фізичних ресурсах та як вони будуть використовуватися.

Розробили концептуальну модель інформаційної системи, виявили сутності та зв'язки між ними та побудували ER -модель. Після цього на основі ER -моделі побудували реляційну базу даних і навели опис усіх таблиць та атрибутів, що знаходяться в ній.

Всі ці діаграми і таблиці разом надають комплексне уявлення про архітектуру та функціональні можливості розроблюваної інформаційної системи. Вони є важливим інструментом для розуміння та комунікації в рамках проектування системи.

					<b>КС КРБ 123.364.00.00 ПЗ</b>	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

## РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА

У цьому розділі наведено опис алгоритмів роботи програми та приклади роботи інформаційної системи.

### 3.1 Алгоритм авторизації користувача

Для забезпечення безпеки роботи програми моніторингу і забезпечення доступу до її ресурсів окремих користувачів необхідно попередньо здійснити авторизацію. Суть її полягає в перевірці введених користувачем облікових даних (таких як ім'я користувача і пароль) для визначення правильності цих даних і надання доступу до системи, якщо перевірка пройшла успішно. Цей алгоритм наведено на рисунку 3.1

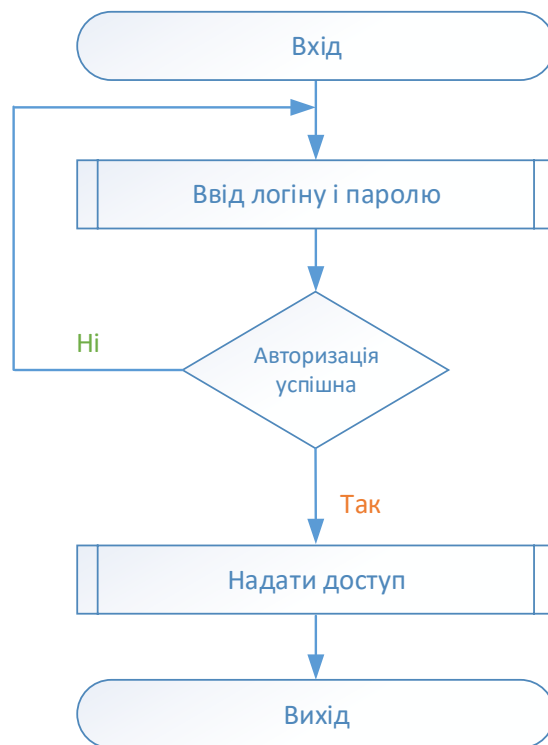


Рисунок 3.1 - Алгоритм авторизації користувача

					<b>КС КРБ 123.364.00.00 ПЗ</b>			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Шевчук М. С.			<b>Практична частина</b>	Літ.	Арк.	Аркушів
Перевір.		Тиш Є.В.					38	
Реценз.		Дуда О.М.				<b>ТНТУ, каф. КС, гр. СІс-41</b>		
Н. Контр.		Луцик Н.С.						
Затверд.		Осухівська Г.М.						

Сценарій алгоритму авторизації наступний:

1. Користувач вводить свої облікові дані (ім'я користувача та пароль).
2. Система перевіряє ці дані на коректність.
3. Якщо дані вірні, користувачеві надається доступ до системи.
4. Якщо дані невірні, система виводить повідомлення про помилку на екран і просить користувача ввести коректні дані.
5. Процес повторюється, поки користувач не надасть правильні дані або не вийде з процедури авторизації.

Якщо перевірка не вдалася, алгоритм виводить помилку на екран і запрошує користувача ввести коректні дані (рис. 3.2).

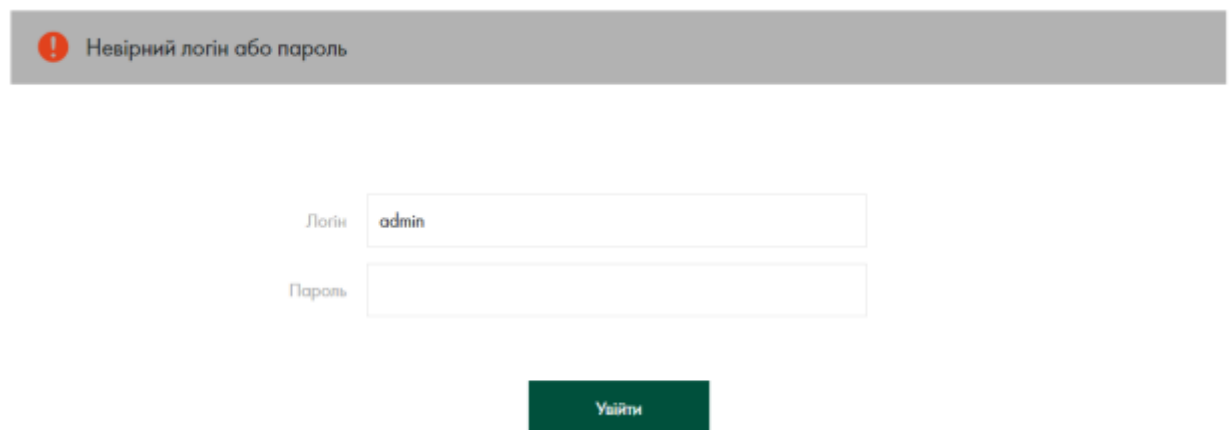


Рисунок 3.2 - Приклад неправильного введення логіну/паролю

### 3.2 Алгоритм роботи Агента

Одним із важливих модулів в системі моніторингу ПЗ є «Агент» який попередньо встановлюється на комп'ютерах в локальній мережі організації і використовується для збору інформації з комп'ютерів організації та їх сортування якщо є задані ліцензії.

Якщо не було ліцензії, які могли відповідати патернам, то їм надається значення FullString, а тип ліцензії Undefined. Так як більшість програм в операційній системі Windows визначаються SNMP [9] протоколом, як

					<b>КС КРБ 123.364.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		39

application ,це можна побачити на рисунку 3.3, то за замовчуванням всі програми на комп'ютерах в мережі отримують тип Application.

```
182 HOST-RESOURCES-MIB::hrSWInstalledType.2 = INTEGER: application(4)
183 HOST-RESOURCES-MIB::hrSWInstalledType.3 = INTEGER: application(4)
184 HOST-RESOURCES-MIB::hrSWInstalledType.4 = INTEGER: application(4)
185 HOST-RESOURCES-MIB::hrSWInstalledType.5 = INTEGER: application(4)
186 HOST-RESOURCES-MIB::hrSWInstalledType.6 = INTEGER: application(4)
187 HOST-RESOURCES-MIB::hrSWInstalledType.7 = INTEGER: application(4)
188 HOST-RESOURCES-MIB::hrSWInstalledType.8 = INTEGER: application(4)
```

Рисунок 3.3 – Приклад MIB – дерева

При натисканні кнопки "Сканування ПЗ" користувач інформаційної системи запускає процес сканування комп'ютера. Під час цього процесу «Агент» звертається до комп'ютера і отримує список програм, які встановлені на ньому.

Далі «Агент» перевіряє цей список програм з базою даних, яка містить відомості про програми які встановлені на ПК організації. Якщо знайдені програми вже є в базі даних і відповідають записам, то список залишається без змін.

Проте, якщо знайдені програми не відповідають жодному запису в базі даних, то це вважається відмінністю. В такому випадку «Агент» додає нові програми до бази даних, розширюючи список встановлених програм.

Цей процес дозволяє адміністратору мережі отримувати актуальний список програм, встановлених на комп'ютері, і спостерігати за змінами, які можуть виникли після сканування (рис. 3.4).



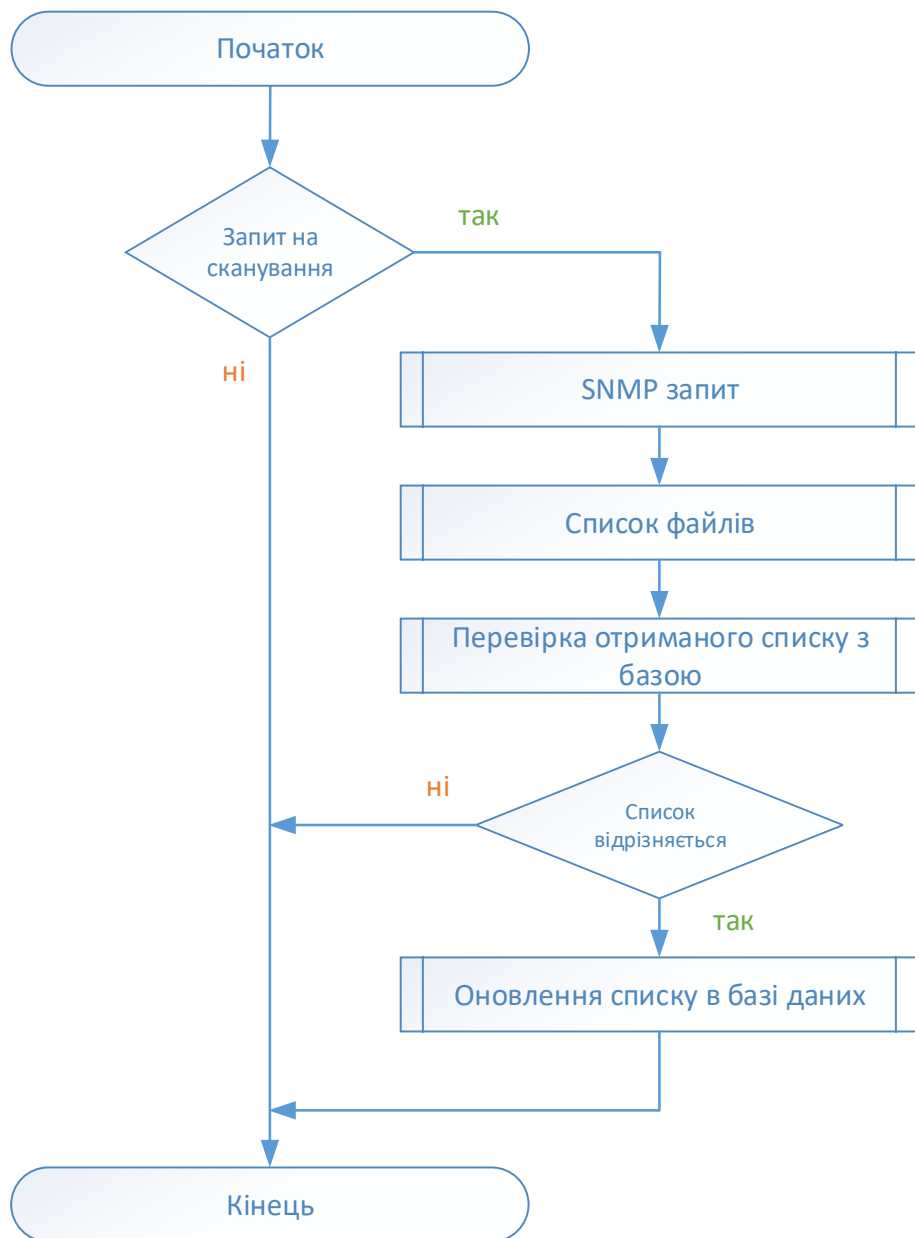


Рисунок 3.4 - Алгоритм роботи агента

На першому етапі необхідно провести сканування програмного забезпечення що встановлено на ПК. Ця інформація може використовуватися для формування шаблону програмного забезпечення, яке присутнє на комп'ютері, і для контролю версії ПЗ та ліцензій. Для тестування роботи системи використовуються два типи ліцензій: "Undefinit" і "Microsoft". Для ліцензій "Microsoft" визначено тип патерну "SubString", а для програм, розроблених іншими виробниками, визначено тип патерну "FullString". Для більшості програм, які встановлені на комп'ютері з операційною системою

Змн.	Арк.	№ докум.	Підпис	Дата

Windows, за замовчуванням встановлюється тип "Application".

Приклад результату сканування:

1. Назва програми: Microsoft Office 365 Версія: 16.0.12345 Ліцензія: Microsoft Тип патерну: SubString Тип програми: Application
2. Назва програми: Google Chrome Версія: 93.0.4577.63 Ліцензія: Undefinit Тип патерну: FullString Тип програми: Application
3. Назва програми: WinRAR Версія: 6.02 Ліцензія: Undefinit Тип патерну: FullString Тип програми: Application

На рисунку 3.5 представлений результат сканування в табличному вигляді. Маючи права Адміністратора можна додати до цього списку додаткові програми та відповідну інформацію, якщо вони присутні на комп'ютері.

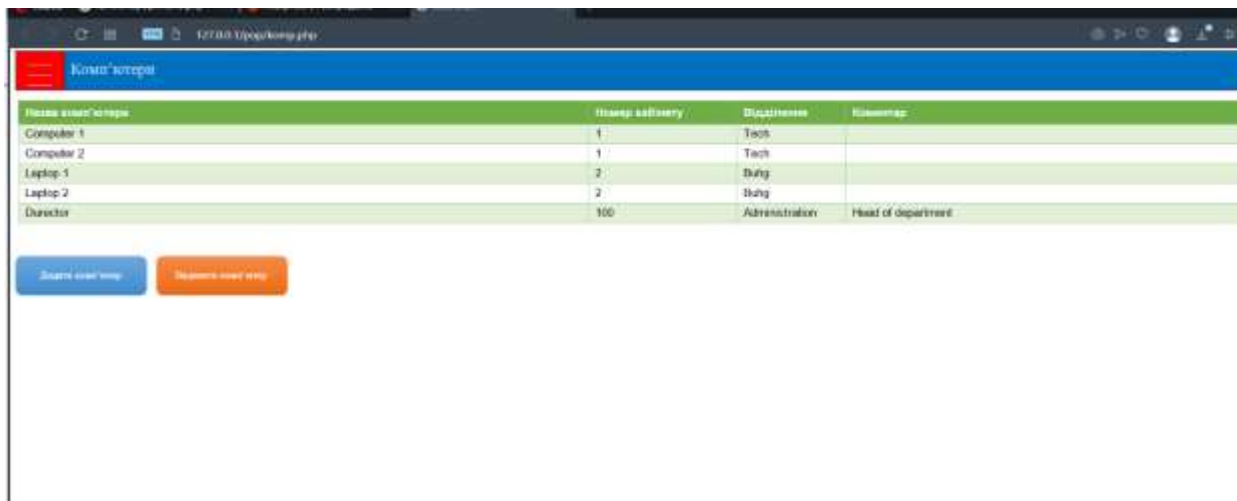


Назва програми	Версія ПЗ	Тип програми	Тип ліцензії	Кількість
SkypeClient-Components	10.0.4711.1001	Application	SubString	1
Viber	17.6.1.11	Application	FullString	1
АйБіМ екстрем (6.7)	7.22.11	Application	SubString	1
Microsoft Visio Professional 2013	10.0.4093.1006	Application	SubString	1
Microsoft Excel 16.0 (Business) 2013	10.0.4093.1006	Application	SubString	1
Microsoft PowerPoint 16.0 (Business) 2013	10.0.4093.1006	Application	SubString	1
Microsoft .NET Targeting Pack - 6.0.5 (x64)	40.23.48045	Application	SubString	1
Microsoft .NET SDK (Main Manifest) 6.0.208	24.75.5513	Application	SubString	1
Microsoft .NET Workload Mono Tasklist Manifest	40.2.39916	Application	SubString	1
Microsoft Visual C++ 2012 x64 Additional Runtime	10.32.31306	Application	SubString	1
Visual C++ Library CRT Appx Resource Package	10.32.31306	Application	SubString	1
Windows SDK for Windows Store Apps Metadata	10.1.19041.800	Application	SubString	1
Windows Desktop Extension SDK Linkage	10.1.19041.800	Application	SubString	1
WindowsStore/Views	7.6	Application	FullString	1
Google Chrome	114.0.5735.110	Application	FullString	1
Microsoft .NET Toolset 6.0.208 (x64)	24.4.01005	Application	SubString	1
Microsoft ASP.NET Core 6.0 Shared Framework (x64)	6.0.6.22102	Application	SubString	1
Microsoft .NET Framework 4.8 Targeting Pack	4.8.03781	Application	SubString	1
Microsoft .NET Core/Runtime/Targeting Pack	10.0.0	Application	SubString	1
Windows SDK Desktop Tools x64	10.1.19041.800	Application	SubString	1
Visual C++ Library CRT Appx Package	10.32.31306	Application	SubString	1
Microsoft .NET SDK (OS Manifest) 6.0.208	10.32.31306	Application	SubString	1
Microsoft .NET 6.0 Template 6.0.208 (x64)	10.0.24225.01	Application	FullString	1
Microsoft .NET 6.0 Template 6.0.208 (x86)	10.0.24225.01	Application	FullString	1
OperaMini	40.0.4762.47	Application	FullString	1
OperaMini/Windows	6.1.30	Application	FullString	1
VisualStudioCommunity 2022	17.2.3	Application	SubString	1
ms-rmfl	6.11.0	Application	FullString	1

Рисунок 3.5 - Приклад після сканування ПЗ

### 3.3 Тестування роботи

Після входу в систему користувачу відображається вікно з інформацією про комп'ютери, включаючи їх ім'я та розташування (див. рис. 3.6). У цьому вікні доступні дві кнопки: "Додати новий комп'ютер" і "Видалити комп'ютер".



Назва комп'ютера	Номер апаратури	Відділення	Посада
Computer 1	1	Tech	
Computer 2	1	Tech	
Laptop 1	2	Busng	
Laptop 2	2	Busng	
Director	100	Administration	Head of department

Рисунок 3.6 - Стан комп'ютерів

При натисканні кнопки "Додати новий комп'ютер" відкривається вікно (рис. 3.7), де користувач може ввести ім'я комп'ютера, вказати кабінет, а також вибрати відділи, які вже маються в базі даних. Після заповнення всіх необхідних полів користувач підтверджує додавання, натискаючи кнопку "Додати комп'ютер". Вся введена інформація про комп'ютер заноситься до бази даних, а вікно закривається.

Цей процес дозволяє користувачу додавати нові комп'ютери до системи та пов'язувати їх з відповідними кабінетами та відділами, що спрощує налагодження і контроль за комп'ютерами в організації.

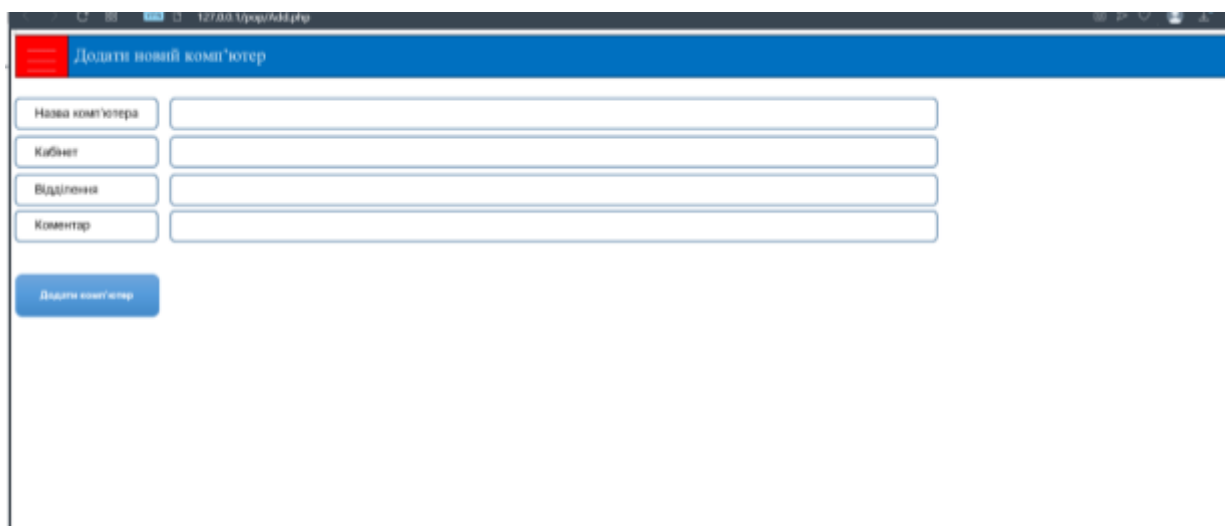


Рисунок 3.7 - Приклад додавання нового комп'ютера до системи

Для видалення комп'ютера з програми необхідно виконати наступні кроки:

1. У вікні з інформацією про комп'ютери користувач повинен натиснути кнопку "Видалити комп'ютер".

2. Відкриється вікно (рис. 3.8), де користувач повинен ввести інформацію про комп'ютер, який потрібно видалити. Ця інформація включає ім'я комп'ютера і кабінет.

3. Після внесення всієї необхідної інформації користувач підтверджує видалення, натискаючи кнопку "Видалити комп'ютер".

4. Інформація про комп'ютер, яка була введена раніше, буде видалена з бази даних, а вікно буде закрито.

Цей функціонал дозволяє користувачу видаляти інформацію про комп'ютери з системи, які більше не потрібні або були помилково додані.

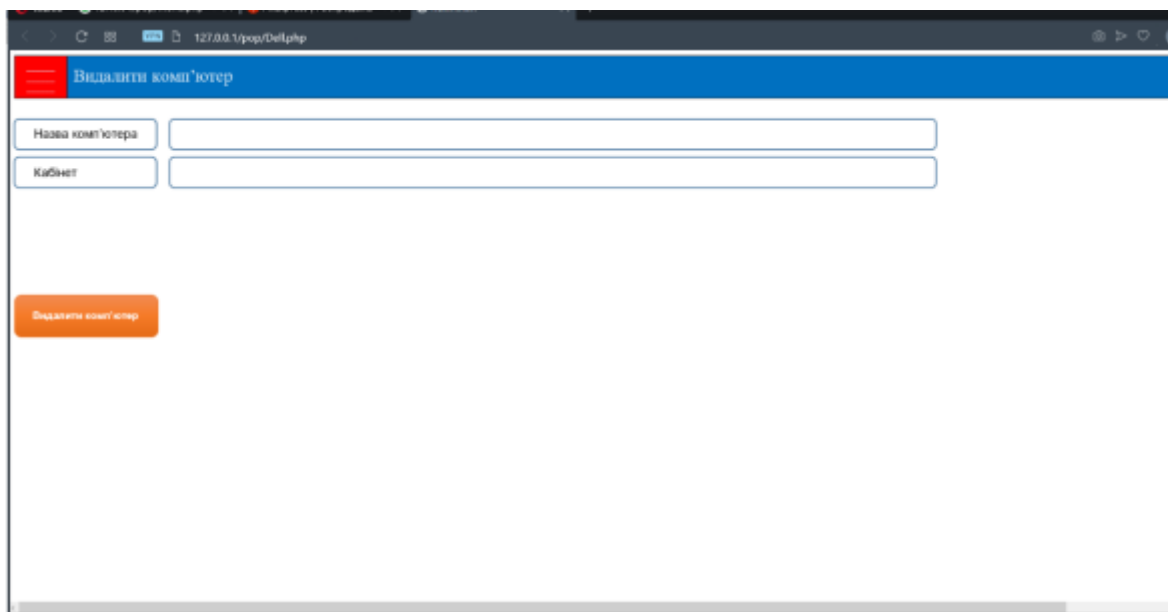


Рисунок 3.8 - Приклад видалення комп'ютера із системи

У лівій верхній частині екрана є кнопка, яка відкриває меню програми (рис. 3.9) з пунктами «Перелік шаблонів», «Перелік комп'ютерів», «Програми», «Пошук» і «Вихід».

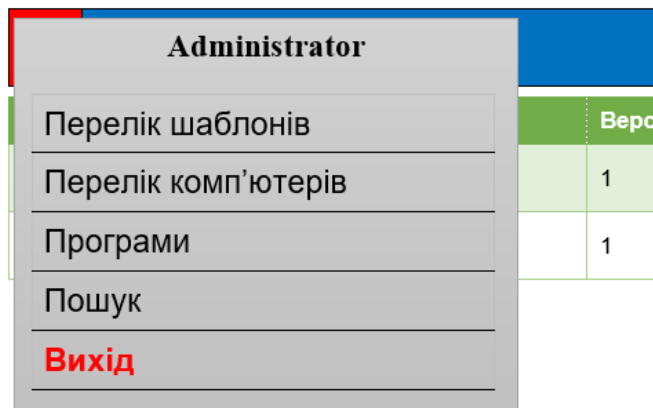


Рисунок 3.9 - Меню програми

Шаблони є одним з основних елементів системи моніторингу. Шаблон створюється для кожного компютера в локальній мережі і містить повну інформацію про ПК, перелік програмного забезпечення яке на ньому встановлено, версії програмного забезпечення і інформацію про ліцензії. Задяки порівнянню шаблонів адміністратор отримує інформацію про те яке ПЗ булоо встановлено, яке необхідно оновити, і чи не закінчився термін ліцензії.

У вікні «Шаблони» (ВІКНО з ШАБЛОНАМИ) користувач може побачити інформацію про шаблони: Назва шаблону, Версія, Комп'ютер (до якого відноситься шаблон). Є три кнопки «Створити шаблон», «Видалити шаблон», «Інформація/редагувати шаблон».

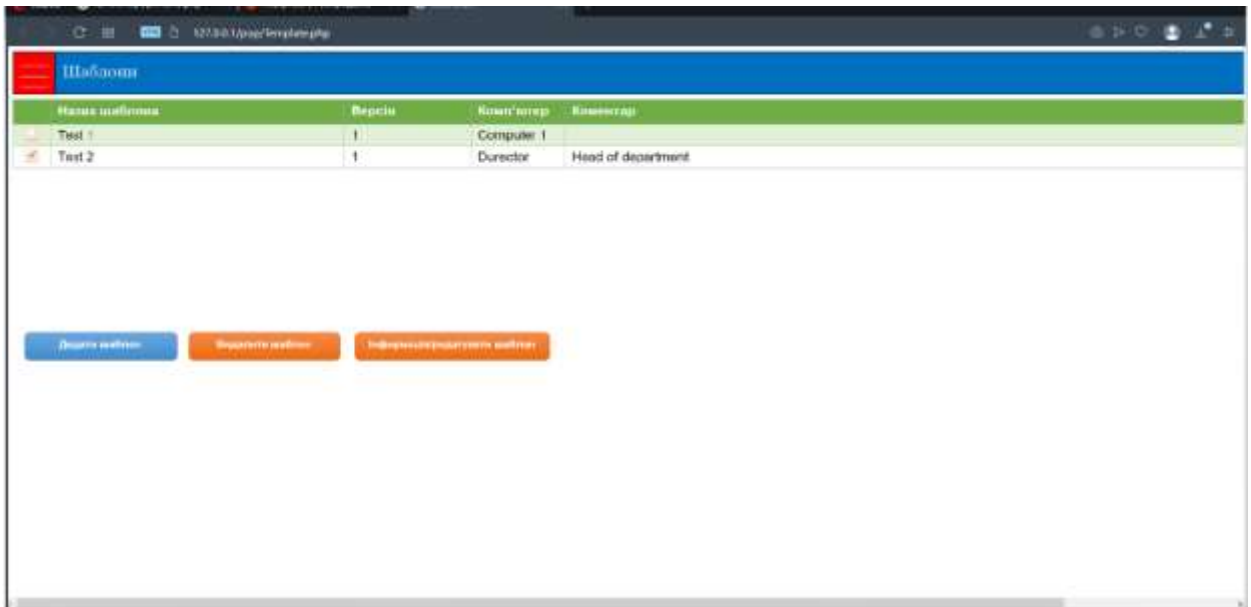


Рисунок 3.10 - Вікно шаблону

Коли користувач натискає на «Додати шаблон», з'явиться вікно з полями «Назва шаблону», «Коментар», «Програми», «Назва комп'ютера» і «Номер кабінету» (рис. 3.11).

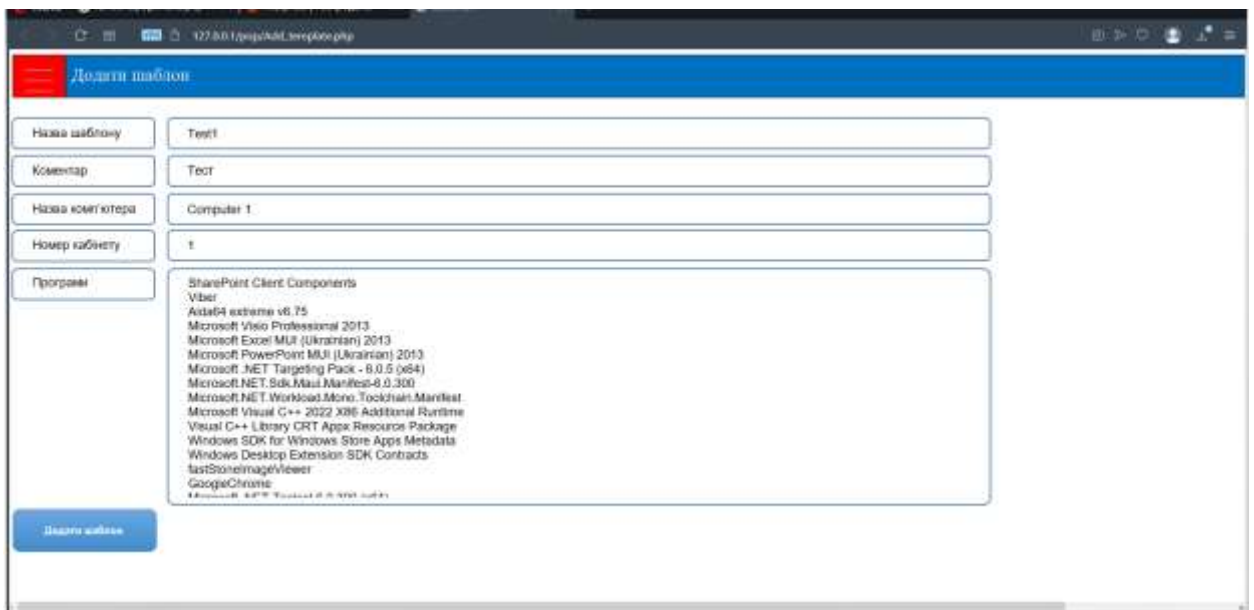


Рисунок 3.11 - Приклад додавання нового шаблону

Інформація вводиться у відповідні поля назва шаблону, назва комп'ютера, номер кабінету де встановлено комп'ютер, в полі програми

«Програми» вноситься перелік програм зібраних агентом. При натисканні кнопки «Додати шаблон» вся раніше введена інформація зберігається і вікно закривається.

Для видалення шаблону необхідно вибрати шаблон який потрібно видалити і натиснути кнопку «Видалити шаблон» (рис.3.12) його буде видалено, а вікно закрито.



Рисунок 3.12 - Видалення шаблону

Третя кнопка «Інформація/редагування шаблону» відкриває вікно наведене на рисунку 3.13, або можна скористатись пошуком шаблонів.

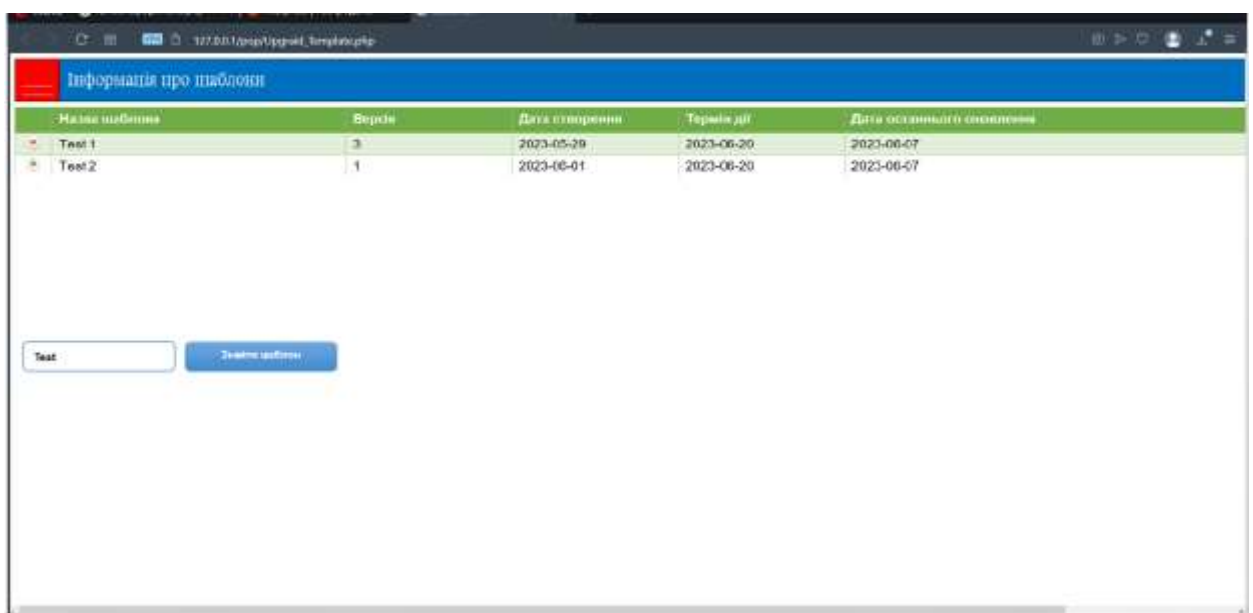


Рисунок 3.13 - Приклад інформації про шаблони

Крім того в першому полі є індикатор актуальності шаблону. Тобто якщо індикатор зелений то програмне забезпечення відповідає даним які внесені до шаблону, якщо колір червоний це означає що були зміни в структурі програмного забезпечення комп'ютера до якого прив'язаний шаблон, або закінчилась ліцензія на програмне забезпечення.

Як вже згадувалось раніше звірка даних з шаблоном може відбуватися періодично, або може бути ініційована адміністратором вручну.

У вікні пошуку користувач може побачити вікно з параметрами пошуку і полем для введення слова, яке потрібно знайти в системі. Після введення ключа пошуку та вибору необхідних пунктів будується таблиця результатів (рис. 3.14).



Рисунок 3.14 - Приклад знайденої інформації

### 3.4 Висновок

У даному розділі було наведено алгоритм роботи агента в системі моніторингу програмного забезпечення. Було описано процес збору інформації з комп'ютерів організації та їх сортування, використання ліцензій та патернів для визначення типу програмного забезпечення, а також взаємодію агента з сервером для оновлення бази даних.



Додатково, було надано приклад роботи програми, де здійснювалося сканування програмного забезпечення, перевірка наявності в базі даних, а також додавання нових програм при виявленні відмінностей. Цей приклад показав коректну роботу системи моніторингу програмного забезпечення, що дозволяє контролювати версії ПЗ та ліцензії на комп'ютерах організації.

Щодо інтерфейсу то, його було реалізовано за допомогою Web інтерфейсу користувача, що забезпечує зручну взаємодію з системою моніторингу, через вікна, кнопки та текстові поля.

На підставі наданого прикладу роботи програми можна зробити висновок про правильну роботу системи моніторингу програмного забезпечення, яка забезпечує збір інформації, контроль версій ПЗ та ліцензій на комп'ютерах організації.

					<b>КС КРБ 123.364.00.00 ПЗ</b>	Арк.
						49
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

## РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

### 4.1 Санітарно-гігієнічні вимоги до умов праці

Важливу роль у експлуатації кабельних систем різного призначення відіграє охорона праці, оскільки в ній закладені сукупність всіх взаємозв'язаних правових, організаційно-технічних, санітарно-гігієнічних, соціально-економічних, лікувально-профілактичних заходів та управлінських рішень, спрямованих на збереження життя, здоров'я та працездатності працівників у процесі їх трудової діяльності.

В даній кваліфікаційній роботі розробляється комп'ютерна система моніторингу програмного забезпечення локальної мережі підприємства. Тому дотримання норм з охорони праці є особливо актуальним. На прикладі одного з приміщень, я проаналізую комплекс заходів, що забезпечать санітарно-гігієнічні та безпечні умови праці.

Розглянемо типовий приклад кімнати персоналу, що обслуговує і адмініструє локальну мережу.

Мікроклімат у такому приміщенні забезпечується системою індивідуального кондиціонування, централізованою системою витяжної вентиляції. В світлу пору доби використовується природне освітлення, по необхідності є можливість забезпечувати робочі місця, як індивідуальним так і централізованим освітленням люмінесцентних ламп відповідно до ДБН В.2.5-28:2018 [12].

В даному приміщенні передбачено десять електромережових розеток на напругу 220 В, частоту 50 Гц, що крім контактів фазового та нульового робочого провідників, мають спеціальні контакти для підключення нульового захисного провідника. Їхня конструкція є такою, щоб приєднання нульового

					<b>КС КРБ 123.364.00.00 ПЗ</b>			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		Шевчук М. С.			<i>Безпека життєдіяльності, основи охорони праці</i>	<i>Лім.</i>	<i>Арк.</i>	<i>Аркуші</i>
<i>Перевірів</i>		Тиш Є.В.					50	
<i>Консульт.</i>						<i>ТНТУ, каф. КС, гр. СІс-41</i>		
<i>Н. Контр.</i>		Луцик Н.С.						
<i>Затверд.</i>		Осухівська Г.М.						

захисного провідника відбувалося раніше, ніж приєднання фазового та нульового робочого провідників. Не допускається підключати ЕОМ з ВДТ і ПП до звичайної двопровідної електромережі, в тому числі з використанням перехідних пристроїв.

Площа та об'єм для одного робочого місця визначаються згідно з вимогами ДСанПіН 3.3.2-007-98 [13] і повинна становити не менше 6,0 м<sup>2</sup>, об'ємом не менше 20,0 м<sup>3</sup>. Згідно цих вимог та характеристики приміщення (загальна площа 13 м<sup>2</sup>) я розмістив в ньому 2 робочі місця. Схема підключення до комп'ютерної мережі виконана таким чином, щоб забезпечити максимальну ергономічність кожного робочого місця, які б в повному обсязі відповідали вимогам Державних санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин, затверджених постановою Головного державного санітарного лікаря України від 10.12.98 №7 (ДСанПіН 3.3.2-007-98). Кількість інформаційних розеток для підключення мережевих пристроїв розміщується рівномірно по периметру кімнати.

Так як в приміщенні передбачено 2 робочих місця на яких крім ЕОМ встановлено інше обладнання, передбачено аварійний резервний вимикач, який може повністю вимкнути електричне живлення приміщення, крім освітлення.

Ця норма взята з вимог електробезпеки під час експлуатації ЕОМ з ВДТ і ПП, а саме ЕОМ з ВДТ і ПП, інше устаткування (апарати управління, контрольно-вимірювальні прилади, світильники), електропроводи та кабелі за виконанням і ступенем захисту мають відповідати класу зони за НПАОП 40.1-1.01-97 [14], мати апаратуру захисту від струму короткого замикання та інших аварійних режимів.

Отже, при виконанні кваліфікаційної роботи були враховані норми безпеки та охорони праці, що забезпечить усунення причин, які призводять до нещасних випадків, виробничих травм та професійних захворювань.

					<b>КС КРБ 123.364.00.00 ПЗ</b>	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

## 4.2 Надзвичайні ситуації: визначення причини, класифікація.

Надзвичайна ситуація - це ситуація на об'єкті чи окремій території, яка викликана подіями природного, екологічного, технічного, соціального, військового та іншого характеру, що призвела чи може призвести до значної шкоди, порушення нормальної життєдіяльності та загибелі людей [15].

Серед причин, які викликають НС, потрібно виділити такі, як аварії, катастрофи, стихійні лиха.

У сучасному розумінні поняття «аварія» - це пошкодження, вихід із ладу будь-якого механізму, технологічної лінії, руйнування будинків, мостів, транспортних магістралей та вихід із ладу виробництв. Джерелом аварії можуть бути транспортні засоби, заводи, відсталі технології, застаріле обладнання гідроелектростанцій, АЕС.

Катастрофа- це раптове лихо чи велика подія, яка спричиняє важкі наслідки для людини, тваринного чи рослинного світу, змінюючи умови середовища існування.

Залежно від масштабності та тривалості впливу на природне середовище катастрофи поділяють на локальні, регіональні (національні) та глобальні. Прикладами глобальних катастроф можуть служити як особливо важкі аварії, регіональні військові конфлікти, так і різні стихійні лиха, що завдають великої шкоди.

Але не масштабом розвитку стихійних лих визначається вагомість катастроф, а інтенсивністю природних явищ, числом жертв та ступенем впливу на природне середовище, що в цілому можна назвати значною шкодою стосовно конкретної системи.

Серед природних явищ розрізняють небезпечні природні явища та стихійні лиха.

					<b>КС КРБ 123.364.00.00 ПЗ</b>	Арк.
						52
Змн.	Арк.	№ докум.	Підпис	Дата		

Небезпечне природне явище - стихійна подія природного характеру, яка за своєю інтенсивністю, масштабом поширення та тривалості може викликати негативні наслідки.

Стихійне лихо - катастрофічне природне явище чи процес, які можуть спричинювати людські жертви, значну матеріальну шкоду та інші важкі наслідки. Стихійні лиха переважно пов'язані із природними процесами і можуть бути спровоковані антропогенною діяльністю.

Небезпечні природні явища (стихійні лиха) можуть мати геофізичне, геологічне (екзогенне), метеорологічне, агрометеорологічне та інше походження.

Загальноживаної класифікації НС не існує. В основу існуючих класифікацій кладуть масштаб та глибину НС. Згідно з цим принципом НС поділяють на:

- Локальні - це така ситуація, коли загроза її виникнення чи поширення наслідків (у випадку настання) обмежена виробничим приміщенням;
- об'єктові - це така ситуація, коли загроза її виникнення чи поширення наслідків (у випадку настання) обмежена об'ємом та територією об'єкта;
- місцеві - це така ситуація, коли загроза її виникнення чи поширення наслідків (у випадку настання) обмежена територією міста (району) чи області.;
- регіональні - це така ситуація, коли загроза її виникнення чи поширення наслідків (у випадку настання) обмежена територією краю, декількох областей;
- глобальні- це така ситуація, коли загроза її виникнення чи поширення наслідків (у випадку настання) обмежена територією декількох суміжних країн.

Повніша класифікація враховує сутність та причини виникнення подій НС. Це дає змогу виділити такі НС:

- техногенні;
- природні;
- екологічні;
- соціальні.

У свою чергу, надзвичайні події, що лежать в основі НС, можуть бути класифіковані за:

- суттю та характером подій і процесів, які становлять їх основу;
- найважливішими ознаками прояву;
- характером вражаючих чинників чи джерел небезпеки (теплові, хімічні, радіаційні, біологічні та ін.);
- місцем виникнення чи відомчої належності;
- основними причинами виникнення (конструктивні, виробничі, експлуатаційні, погодні та ін.);
- інтенсивністю перебігу;
- масштабами ураження чи впливу;
- характером впливу (руйнування, зараження, затоплення та ін.).

У кінцевому підсумку в загальній класифікації НС техногенного, природного, екологічного та соціального характеру, виконаної для практичних цілей, за головну ознаку буде взята систематизація за суттю та характером базових подій та процесів, які мають місце у НС із врахуванням важливих ознак їх вияву.

Серед НС особливе місце займають події, пов'язані із застосуванням сучасних засобів ураження, вплив яких на об'єкти та людей дорівнює розмірам стихійних лих чи катастроф або перевищує їх.

Таким чином в даному розділі були розглянуті питання, що дозволять забезпечити безпечні умови роботи і по можливості уникнути надзвичайних ситуацій.

					<b>КС КРБ 123.364.00.00 ПЗ</b>	Арк.
						54
Змн.	Арк.	№ докум.	Підпис	Дата		

## ВИСНОВКИ

Цей проект був спрямований на розробку інформаційної системи для моніторингу програмного забезпечення, що встановлено на комп'ютерах організації. Проведено аналіз існуючих рішень на ринку, щоб визначити їх переваги та недоліки і використано цю інформацію для розробки нашої системи.

Важливою складовою розробки було створення ефективної бази даних, в якій зберігається вся інформація про програмне забезпечення на комп'ютерах організації. Ця база даних дозволяє ефективно організувати дані та забезпечити швидкий доступ до них для подальшої обробки.

Алгоритм роботи програми був ретельно розроблений, враховуючи всі вимоги та потреби системи. Він дозволяє здійснювати сканування програмного забезпечення на комп'ютерах, порівнювати його з базою даних та оновлювати інформацію про нові програми або зміни в існуючих.

Інтерфейс користувача був розроблений з орієнтацією на зручну взаємодію з системою, що дозволяє користувачу легко вводити та переглядати інформацію про комп'ютери та їх програмне забезпечення.

Під час тестування системи було перевірено її працездатність і відповідність вимогам. Тестування допомогло виявити й усунути можливі помилки та недоліки, що підвищило якість та надійність системи.

В цілому, розроблена інформаційна система для моніторингу програмного забезпечення є ефективним інструментом для контролю версій та ліцензій ПЗ на комп'ютерах організації. Вона дозволяє зручно збирати, зберігати та аналізувати інформацію про програмне забезпечення, забезпечуючи контроль і оновлення даних.

В результаті виконання кваліфікаційної роботи було розроблено інформаційну систему, яка відповідає нашим початковим вимогам. Таким чином були виконані всі поставлені завдання та досягнуто мети роботи.

					<b>КС КРБ 123.364.00.00 ПЗ</b>	Арк.
						55
Змн.	Арк.	№ докум.	Підпис	Дата		

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Dynatrace vs. Cisco AppDynamics vs. New Relic. URL: <https://www.dynatrace.com/platform/comparison/apm-dynatrace-vs-appdynamics-vs-newrelic/> (дата звернення: 30.05.2023).
2. Application Performance Monitoring. URL: <https://newrelic.com/platform/application-monitoring> (дата звернення: 18.05.2023).
3. AppDynamics Solutions Overview. An answer for every modern enterprise challenge. URL: <https://www.appdynamics.com/solutions> (дата звернення: 17.05.2023).
4. Infrastructure Monitoring. URL: <https://www.dynatrace.com/platform/infrastructure-monitoring/> (дата звернення: 17.05.2023).
5. CSS: Український веб-довідник. URL: <https://css.in.ua/> (дата звернення: 30.05.2023).
6. Методи та інструменти моделювання інформаційних процесів. URL: [http://science.lp.edu.ua/sites/default/files/Papers/40\\_81.pdf](http://science.lp.edu.ua/sites/default/files/Papers/40_81.pdf) (дата звернення: 30.05.2023).
7. Martin Fowler UML Distilled: A Brief Guide to the Standard Object Modeling Language 3rd Edition Addison-Wesley Professional; 3rd edition, 2003. 208 p.
8. PHP.NET. URL: <https://www.php.net/manual/en/index.php> (дата звернення: 05.06.2023).
9. SNMP протокол (основи). URL: <https://www.k-max.name/linux/snmp-protocol/> (дата звернення: 18.05.2023).
10. Yatsyshyn V., Pastukh O., Palamar A., Zharovskyi R. Technology of relational database management systems performance evaluation during computer systems design. Scientific Journal of TNTU.Tern.: TNTU. 2023. Vol 109. No 1. P. 54–65.

					<b>КС КРБ 123.364.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		56



11. Yatsyshyn V., Pastukh O., Zharovskyi R., Shabliy N. Software tool for productivity metrics measure of relational database management system. *Mathematical Modeling*. No 1 (48). 2023. P. 7-17.

12. ДБН В.2.5-28:2018 «Природне і штучне освітлення». Київ. 2018.

13. ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми. Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин». Київ. 1998.

14. НПАОП 40.1-1.01-97 «Правила безпечної експлуатації електроустановок (ДНАОП 1.1.10-1.01-97)» Київ. 1997.

15. Бедрій І.Я. Безпека життєдіяльності. Навчальний посібник. – Київ: Кондор 2009. 286 с.

16. Осухівська Г.М., Тиш Є.В., Луцик Н.С., Паламар А.М. Методичні вказівки до виконання кваліфікаційних робіт здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 123 «Комп'ютерна інженерія» усіх форм навчання. Тернопіль, ТНТУ. 2022. 28 с.

					<b>КС КРБ 123.364.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		57

Додаток А.  
Технічне завдання

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії

Кафедра комп'ютерних систем та мереж

**“Затверджую”**

Завідувач кафедри КС

\_\_\_\_\_ Осухівська Г.М.

“\_\_\_” \_\_\_\_\_ 2023 р

СИСТЕМА МОНІТОРИНГУ РОБОТИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В  
ЛОКАЛЬНІЙ МЕРЕЖІ ПІДПРИЄМСТВА

**ТЕХНІЧНЕ ЗАВДАННЯ**

на 9 листках

**Вид робіт:**

Кваліфікаційна робота

На здобуття освітнього ступеня «Бакалавр»

**Спеціальність 123 «Комп'ютерна інженерія»**

«УЗГОДЖЕНО»

«ВИКОНАВЕЦЬ»

Керівник кваліфікаційної роботи

Студент групи СІс-41

\_\_\_\_\_ к.т.н., доц. Тиш Є.В.

\_\_\_\_\_ Шевчук М. С.

«\_\_\_» \_\_\_\_\_ 2023 р.

«\_\_\_» \_\_\_\_\_ 2023 р.

**Тернопіль 2023**

## 1 Загальні відомості

### 1.1 Повна назва та її умовне позначення

Повна назва теми кваліфікаційної роботи: «Система моніторингу роботи програмного забезпечення в локальній мережі підприємства».

Умовне позначення кваліфікаційної роботи: КС КРБ 123.364.00.00

### 1.2 Виконавець

Студент групи СІс-41, факультету комп'ютерно-інформаційних систем і програмної інженерії, кафедри комп'ютерних систем та мереж, Тернопільського національного технічного університету імені Івана Пулюя, Шевчук Микола Сергійович.

### 1.3 Підстава для виконання роботи

Підставою для виконання кваліфікаційної роботи є наказ по університету (№4/7-237 від 28.02.2023 р.)

### 1.4 Планові терміни початку та завершення роботи

Плановий термін початку виконання кваліфікаційної роботи – 01.03.2023 р.

Плановий термін завершення виконання кваліфікаційної роботи – 19.06.2023 р.

### 1.5 Порядок оформлення та пред'явлення результатів роботи

Порядок оформлення пояснювальної записки та графічного матеріалу здійснюється у відповідності до чинних норм та правил ISO, ЕСКД, ЕСПД та ДСТУ.

Пред'явлення проміжних результатів роботи з виконання кваліфікаційної роботи здійснюється у відповідності до графіку, затвердженого керівником роботи. Попередній захист кваліфікаційної роботи відбувається при готовності роботи на 90% , наявності пояснювальної записки та графічного матеріалу.

Пред'явлення результатів кваліфікаційної роботи відбувається шляхом захисту на відповідному засіданні ЕК, ілюстрацією основних досягнень за допомогою графічного матеріалу.

## 2 Призначення і цілі створення системи

### 2.1 Призначення системи

Система моніторингу що розробляється призначена для:

- Збору інформації про встановлене програмне забезпечення комп'ютерів локальної мережі,
- Забезпечення контролю за ліцензіями на програмне забезпечення,
- Створення і для роботи з шаблонами програмного забезпечення які встановлені на комп'ютерах,
- Пошуку необхідної інформації користувачами системи.

### 2.2 Мета створення системи

Метою кваліфікаційної роботи є розробка системи моніторингу програмного забезпечення на комп'ютерах що підключені до локальної мережі, для контролю і подальшого продовження термінів дії ліцензійних угод, для виявлення самовільно встановленого програмного забезпечення, для пошуку комп'ютера, що підходить для встановлення програмного забезпечення за мінімальними системними вимогами.

## 2.3 Характеристика об'єкту

Характеристика об'єкту проектування, для якого розробляється система моніторингу програмного забезпечення:

1. Організація. Об'єктом проектування є організація, яка володіє локальною мережею.

2. Локальна мережа. В організації існує локальна мережа, яка об'єднує комп'ютери. Ця мережа може бути заснована на провідному або безпроводному з'єднанні і дозволяє обмін даними та ресурсами між комп'ютерами.

3. Комп'ютери. У локальній мережі розташовані різноманітні комп'ютери, які використовуються в організації. Ці комп'ютери можуть мати різне апаратне забезпечення у відповідності до якого необхідно здійснювати підбір ПЗ.

4. Програмне забезпечення. На комп'ютерах встановлене програмне забезпечення, це можуть бути офісні програми, спеціалізоване ПЗ, операційні системи та інше. Система моніторингу повинна мати контроль над використанням цього ПЗ, зокрема ліцензій.

5. Контроль ліцензій. Система повинна відстежувати, яке ПЗ встановлене на кожному комп'ютері, які ліцензії на нього є, чи використовуються ці ліцензії легально і чи не встановлене користувачами своє ПЗ.

## 3 Вимоги до системи

### 3.1 Вимоги до системи в цілому

#### 3.1.1 Вимоги до структури та функціонування системи

Комп'ютерна система моніторингу програмного забезпечення повинна складатись з:

- Агент. Це програмне забезпечення, що відповідає за збір даних про роботу програмного забезпечення та надсилання цих даних на сервер моніторингу.

- Мережеві протоколи. Для передачі даних між агентами та сервером використовуються протокол SNMP (Simple Network Management Protocol), який дозволяє отримувати статистику та керувати мережевими пристроями..

- Сервер моніторингу. Це центральний компонент системи, який отримує дані від агентів і зберігає їх для подальшої обробки та візуалізації. Сервер може мати базу даних для зберігання історичних даних, а також може виконувати аналіз і сповіщення про стан програмного забезпечення.

- Інтерфейс користувача. Веб-інтерфейс, який дозволяє користувачам переглядати та аналізувати дані моніторингу. Через інтерфейс користувач може переглядати статистику, отримувати сповіщення про проблеми, налаштовувати моніторингові правила тощо.

### 3.1.2 Вимоги до способів та засобів зв'язку між компонентами системи

Основна вимога, яка ставиться до способів та засобів інформаційного обміну – це їх узгодженість.

### 3.1.3 Вимоги до режимів функціонування системи

Для системи визначено два режими функціонування:

- нормальний режим функціонування;
- аварійний режим функціонування.

Основним режимом функціонування є нормальний режим.

Для забезпечення нормального режиму функціонування системи необхідно виконувати вимоги і дотримуватись умов експлуатації програмного забезпечення і комплексу технічних засобів системи, вказані у відповідних технічних документах (технічна документація, інструкції з експлуатації і т. д.).

Аварійний режим функціонування системи характеризується відмовою одного або декількох компонент програмного і (або) технічного забезпечення. При цьому функції роботи системи продовжують підтримувати роботу системи моніторингу в межах базових налаштувань.

#### 3.1.4 Вимоги по діагностуванню системи

Для діагностування системи моніторингу використовуються інструменти діагностування основних процесів системи, які вмонтовані в операційну систему, а також програм для діагностики роботи комп'ютерних мереж.

Інструменти повинні забезпечувати зручний інтерфейс для можливості перегляду діагностичних подій, моніторингу процесу виконання програм.

#### 3.1.5 Перспективи розвитку, проектування системи

Дана система може бути розширена завдяки використанню протоколу SNMP, що дозволяє отримувати більшу кількість інформації, яку можна використати для моніторингу роботи програмного забезпечення. Наприклад, використовувати дані про версії програмного забезпечення для виявлення застарілих або вразливих версій, передбачити можливість автоматичного оновлення, або для виявлення неполадок.

Проектування системи з використанням протоколу SNMP потребує розробки відповідних модулів системи моніторингу для взаємодії з пристроями, обробки отриманих даних та їх подальшого. Важливо також забезпечити безпеку підключення до пристроїв через протокол SNMP та належну обробку та збереження отриманої інформації.

### 3.2 Показники призначення

Система повинна передбачати можливість масштабування. Можливості масштабування повинні забезпечуватися засобами використовуваного



базового програмного і технічного забезпечення.

### 3.2.1 Вимоги до надійності

Система повинна забезпечувати працездатність та відновлення своїх функцій при виникненні наступних ситуацій:

- при збоях в системі електропостачання апаратної частини;
- при помилках в роботі апаратних засобів;
- при помилках, пов'язаних з програмним забезпеченням (ОС і драйвери пристроїв).

Для захисту апаратури від стрибків напруги і комутаційних завад повинні застосовуватися мережні фільтри.

### 3.3 Вимоги до безпеки

Зовнішні елементи технічних засобів системи, що перебувають під напругою, повинні мати захист від випадкового дотику, а самі технічні засоби мати занулення або захисне заземлення .

Система електроживлення повинна забезпечувати захисне вимикання при перевантаженнях і коротких замиканнях в колах навантаження, а також аварійне ручне вимикання.

Загальні вимоги пожежної безпеки повинні відповідати нормам на побутове електрообладнання. У разі пожежі не має виділятися отруйних газів і димів. Після зняття електроживлення має бути доступне застосування будь-яких засобів пожежогасіння.

#### 3.3.1 Вимоги до експлуатації, технічного обслуговування, ремонту і зберігання компонентів системи

Мікроклімат в приміщеннях повинен відповідати нормам виробничого мікроклімату по ДСН 3.3.6.042-99:

- температуру повітря в межах від +10°C до +35°C;
- відносну вологість повітря при 25°C в межах від 30% до 80%;
- атмосферний тиск 760±25 мм рт. ст.

Періодичне технічне обслуговування використовуваних технічних засобів має проводитися відповідно до вимог технічної документації, але не рідше ніж один раз на рік.

Періодичне технічне обслуговування і тестування технічних засобів повинні включати обслуговування і тестування всіх використовуваних засобів, датчики, контролери, системи передачі даних, пристрої безперебійного живлення.

На підставі результатів тестування технічних засобів повинні проводитися аналіз причин виникнення виявлених дефектів і прийматися заходи по їх ліквідації.

#### 3.4 Вимоги до захисту інформації від несанкціонованого доступу

Система повинна забезпечувати захист від несанкціонованого доступу на рівні не нижче встановленого вимогами, що пред'являються до категорії 1Д по класифікації документа, що діє, “Автоматизовані системи. Захист від несанкціонованого доступу до інформації. Класифікація автоматизованих систем”.

Компоненти підсистеми захисту від НСД повинні забезпечувати:

- ідентифікацію користувача;
- перевірку повноважень користувача при роботі з системою;
- розмежування доступу користувачів.

Рівень захищеності від несанкціонованого доступу засобів обчислювальної техніки, що здійснюють обробку конфіденційної інформації, повинен відповідати вимогам класу захищеності згідно вимогам документу “Засоби обчислювальної техніки. Захист від несанкціонованого доступу до

інформації. Показники захищеності від несанкціонованого доступу до інформації”.

#### 3.4.1 Вимоги по збереженню інформації при аваріях

Інформація, при виникненні аварійних ситуацій повинна бути збережена на резервних носіях.

#### 3.4.2 Вимоги по стандартизації і уніфікації

Система повинна відповідати вимогам ергономіки і зручності користування за умови комплектування високоякісним обладнанням (ЕОМ, монітор і інше обладнання), що має необхідні сертифікати відповідності і безпеки.

#### 3.4.3 Вимоги до функцій (завдань), що виконуються системою:

- забезпечення зручного інтерфейсу;
- забезпечення безпечного зберігання шаблонів;
- пошук необхідної інформації;
- забезпечення високої швидкодії.

### 4 Вимоги до документації

Документація повинна відповідати вимогам ЄСКД та ДСТУ

Комплект документації повинен складатись з:

- пояснювальної записки;
- графічного матеріалу:

Структурна схема системи.

Діаграми варіантів використання.

Діаграма розгортання системи.

Результати роботи системи.

\*Примітка: У комплект документації можуть вноситися міни та доповнення в процесі розробки.

## 5 Стадії та етапи проектування

Таблиця 1 – Стадії та етапи виконання кваліфікаційної роботи бакалавра

№ етапу	Назва етапу виконання кваліфікаційної роботи	Термін виконання
1	Розробка технічного завдання	01.03-25.03.2023
2	Аналіз технічного завдання	26.03-01.04.2023
3	Аналіз вимог до системи моніторингу роботи ПЗ	02.04-16.04.2023
4	Проектування структури системи	17.04-04.05.2023
5	Проектування і реалізація програмної частини	05.05-31.05.2023
7	Налаштування і тестування системи моніторингу ПЗ	01.06-06.06.2023
8	Безпека життєдіяльності, основи охорони праці	07.06-08.06.2023
9	Оформлення кваліфікаційної роботи	05.06-12.06.2023
10	Попередній захист кваліфікаційної роботи	12.06-16.06.2023
11	Захист кваліфікаційної роботи	19.06-24.06.2023

## 6 Додаткові умови виконання кваліфікаційної роботи

Під час виконання кваліфікаційної роботи у дане технічне завдання можуть вноситися зміни та доповнення.

## Додаток Б.

### Лістинги програмного забезпечення

#### Лістинг Б.1 - Menu.php

```
<?php
    session_start();
    header("Content-Type: text/html; charset=windows-1251");
?>
<html>
<head>
    <link rel="stylesheet" type="text/css" href="style.css">
    <script type="text/javascript" src="Javas.js"></script>
    <meta http-equiv="Content-Type" content="text/html; charset=windows-
1251" />
    <title>Стан</title>
    <style>
    <meta charset="UTF-8">
    #t{ overflow: scroll;
    }
    </style>
</head>
<body>
    <div id="main">
    <?php
        $db = mysqli_connect("localhost","root","") or die ("Неможливе
підключення до MySQL");
        mysqli_select_db($db, "mydb") or die ("Неможливо відкрити таблицю
з даними");
        mysqli_query($db,"SET NAMES 'UTF-8'");
        $res = mysqli_query($db, "select Comment as 'Комп'ютер',
                                Division_ID as 'Відділення' from
                                computer");

        echo "<table id='t'>";
        echo "<caption align='center'>Комп'ютери</caption>";
        $row = mysqli_fetch_assoc($res);
        foreach ($row as $key => $value) {
            echo("<th>$key</th>");
        }
        echo("<tr id = '0'>");
        foreach ($row as $key => $value) {
            if (is_numeric($value)) { echo("<td> $value</td>");
            }else{ echo ("<td>$value</td>");
            }
        }
        echo("</tr>");
        $id=1;
        while($row = mysqli_fetch_assoc($res)){ echo("<tr id='.$id.'>");
        foreach ($row as $key => $value) {
            if (is_numeric($value)) { echo("<td> $value</td>");
            }else{
            echo ("<td>$value</td>");
            }
        }
        $id++;
        echo("</tr>");
    }
    echo "</table>";
?>
</script>
```

```

var step = 0;
var old=-1;
document.body.onclick = function(e) {
var elem = e ? e.target : window.event.srcElement;
while(!(elem.id || (elem == document.body))) elem =
elem.parentNode;
if (! elem.id) return; else id = elem.id;
if (old != id || id == "information"){
if (step! = 0)
{
if (old == 0|| old%2==0)
document.getElementById(old).style.backgroundColor
= "#E8E6D1";
else
document.getElementById(old).style.backgroundColor
= "white";
}
}
old=id;
if (id != "information")
document.getElementById(id).style.backgroundColor =
"#C0C0C0";
step++;
}
}
</script>
<!-- <a href="#" class="green" id="but">Кнопка</a> -->
<input type="checkbox" id="nav-toggle" hidden>
<label for="nav-toggle" class="nav-toggle" onclick hidden></label>
<nav class="nav">
<label for="nav-toggle" class="nav-toggle" onclick></label>
<?php
echo "<h2 class='logo'>".$_SESSION["login"]."</h2>";
?>
<!-- <h2 class="logo"> Ім'я користувача</h2> -->
<ul>
<li><a href="Template.php">Шаблони</a>
<li> <a href="Soft.php"> Програми</a>
<li><a href="Find.php">Пошук</a>
</ul>
</nav>
<script language="javascript">
function newwindow1()
{
win = window.open("add_computer.php", "win", "height=500,width=500");
}
function newwindow3()
{
win = window.open("dell_computer.php", "win", "height=500,width=500");
}
//-->
</script>
<form>
<input type="button" value=" Додати новий комп'ютер"
onclick="newwindow1()" id="openWind">&nbsp;
<input type="button" value="Видалити комп'ютер" onclick="newwindow3()"
id="openWind">
</form>
<!-- <div id='information'>&nbsp;<font face="Lucida Sans
Unicode">Інформація
</font>
</div>-->
</div>
</body>

```

```
</html>
```

## Лістинг Б.2 - Soft.php

```
<?php
    session_start(); ?>
<html>
<head>
    <link rel="stylesheet" type="text/css" href="style.css">
    <script type="text/javascript" src="Javas.js"></script>
    <meta http-equiv="Content-Type" content="text/html; charset=windows-1251"
/>
    <title>програми</title>
    <style>
    <meta charset="UTF-8">
    #t{ overflow: scroll;
    }
    </style>
</head>
<body>
<?php
    header("Content-Type: text/html; charset=windows-1251");
    $db = mysqli_connect("localhost","root","") or die ("Неможливе
підключення до MySQL");
    mysqli_select_db($db, "mydb") or die ("Неможливо відкрити таблицю з
даними");
    mysqli_query($db,"SET NAMES 'UTF-8'");
    $res = mysqli_query($db, "select Name as 'Назва програми', Comment as
'Коментар', SoftType as 'Тип програми',
TypePatern as 'Тип паттерна', Version as
'Версія' from soft");

    echo "<table id='t'>";
    echo "<caption align='center'>
        Програми
    </caption>";
    $row = mysqli_fetch_assoc($res); foreach ($row as $key => $value)
    {
        echo("<th>$key</th>");
    }
    echo("<tr id = '0'>");
    foreach ($row as $key => $value) {
        if (is_numeric($value)) { echo("<td> $value</td>");
        }else{ echo ("<td>$value</td>");
        }
    }
    echo("</tr>");
    $id=1;
    while($row = mysqli fetch assoc($res)){
    echo("<tr id=".$id.">");
    foreach ($row as $key => $value) {
        if (is numeric($value)) { echo("<td> $value</td>"); }else{
        echo ("<td>$value</td>"); }
    }
    $id++;
    echo("</tr>");
    }
    echo "</table>";
?>
<script>
    var step = 0;
    var old=-1;
    document.body.onclick = function(e) {
    var elem = e? e.target: window.event.srcElement;
    while(!(elem.id || (elem == document.body))) elem =
```

```

        elem.parentNode; if (! elem.id) return; else id = elem.id; if
        (old != id || id == "information"){ if (step!=0) {
            if (old == 0||old%2==0)
                document.getElementById(old).style.backgroundColor =
"#E8E6D1";
            else
                document.getElementById(old).style.backgroundColor =
"white";
        }
        }
        old=id;
        if (id != "information")
            document.getElementById(id).style.backgroundColor = "#C0C0C0";
        step++;
    }
}
</script>
<br>
<a href="#" class="button7">Сканування ПЗ</a>
<br>
<br>
<div id="information">
</div>
<input type="checkbox" id="nav-toggle" hidden>
<label for="nav-toggle" class="nav-toggle" onclick hidden></label>
<nav class="nav">
<label for="nav-toggle" class="nav-toggle" onclick></label>
<?php
echo "<h2 class='logo'>".$ SESSION["login"]."</h2>"; ?>
<!-- <h2 class="logo"> Ім'я користувача</b2> -->
<ul>
<li><a href="Menu.php">Стани</a>
<li><a href="Template.php">Шаблони</a>
<li><a href="Find.php">Пошук</a>
</ul>
</nav>
<form>
<input type="button" value="Знайти інформацію" onclick="newwindow1()"
id="openWind">
</form>
<?php
if ($_GET['id']!= NULL){
$db = mysqli_connect("localhost","root","") or die ("Неможливе
підключення до MySQL");
mysqli_select_db($db, "mydb") or die ("Неможливо відкрити таблицю з
даними");
mysqli_query($db,"SET NAMES 'UTF-8'");
$res = mysqli_query($db, "select Name as 'Ім'я ліцензії', UrlDOWNLOAD as
'Сайт видавця', Comment as 'Коментар',
DateStart as 'Дата початку', DateEnd as 'Дата кінця'
від licenseinfo де ID = 0 ");
$row = mysqli_fetch_assoc($res); foreach ($row as $key => $value) {
echo "<th>$key</th>";
} foreach ($row as $key => $value) { if (is_numeric($value)) {
echo("<td> $value</td>"); }else{ echo ("<td>$value</td>");
} } echo("</tr>");
while($row = mysqli_fetch_assoc($res)){ echo("<tr id=".$row["id"].">");
foreach ($row as $key => $value) {
if (is_numeric($value)) { echo("<td> $value</td>");
}else{ echo ("<td>$value</td>");
} } echo("</tr>");
} echo "</table>";
} ?>
<script language="javascript"> function newwindow1()

```



```

    {
    win = window.open("connect.php", "win", "height=500, width=500");
    }
</script>
</body>
</html>

```

### Лістинг Б.3 - Template.php

```

<?php
    session_start();
    header("Content-Type: text/html; charset=windows-1251"); ?>
<html>
<head>
    <link rel="stylesheet" type="text/css" href="style.css">
    <Title>Шаблони</Title>
    <style>
    </style>
</head>
<body><! -- onload="print()" -->
<?php
    header("Content-Type: text/html; charset=windows-1251");
    $db = mysqli_connect("localhost","root","") or die ("Неможливе
підключення до MySQL");
    mysqli_select_db($db, "mydb") or die ("Неможливо відкрити таблицю
з даними");
    mysqli_query($db,"SET NAMES 'UTF-8'");
    $res = mysqli_query($db, "select distinct Name as 'Ім'я шаблону',
                                Status as 'Статус', Computer_ID as
                                'Комп'ютер' from template");

    echo "<table>";
    echo "<caption align='center'>
        Шаблони
    </caption>";
    $row = mysqli_fetch_assoc($res);
    foreach ($row as $key => $value) {
        echo("<th>$key</th>");
    }
    echo("<tr id = '0'>");
    foreach ($row as $key => $value) {
        if (is_numeric($value)) { echo("<td> $value</td>");
        }else{ echo ("<td>$value</td>");
        }
    }
    echo("</tr>");
    $id=1;
    while($row = mysqli_fetch_assoc($res)){
        echo("<tr id='.$id.'>");
        foreach ($row as $key => $value) {
            if (is_numeric($value)) { echo("<td> $value</td>");
            }else{ echo ("<td>$value</td>");
            }
        }
        $id++;
        echo("</tr>");
    }
    echo "</table>";
?>
<script>
    var step = 0;
    var old=-1;
    var step = 0;
    var str;

```

```

var re = / / gi;
var t;
document.body.onclick = function(e) {
var elem = e ? e.target : window.event.srcElement;
while(!(elem.id || (elem == document.body))) elem =
elem.parentNode;
if (! elem.id) return; else id = elem.id;
if (old != id || id == "information"){
if (step! = 0)
{
if (old == 0|| old%2==0)
document.getElementById(old).style.backgroundColor
= "#E8E6D1";
else
document.getElementById(old).style.backgroundColor
= "white";
}
}
old=id;
if (id != "information")
document.getElementById(id).style.backgroundColor =
"#C0C0C0";
step++;
}
}
</script>
<input type="checkbox" id="nav-toggle" hidden>
<label for="nav-toggle" class="nav-toggle" onclick hidden></label>
<nav class="nav">
<label for="nav-toggle" class="nav-toggle" onclick></label>
<?php
echo "<h2 class='logo'>".$_SESSION["login"]."</h2>";
?>
<!-- <h2 class="logo"> Ім'я користувача</h2> -->
<ul>
<li><a href="Menu.php">Стани</a>
<li> <a href="Soft.php">Програми</a>
<li><a href="Find.php">Пошук</a>
</ul>
</nav>
<input type="button" value="Створити шаблон" onclick="newwindow1()"
id="openWind">
<input type="button" value="Видалити шаблон" onclick="newwindow3()"
id="openWind">
<input type="button" value="Інформація про шаблон"
onclick="newwindow2()"
id="openWind">
<script language="javascript">
function newwindow1()
{
win = window.open("add_template.php", "win", "height=500,width=500");
}
function newwindow3()
{
win = window.open("dell_template.php", "win", "height=500,width=500");
}
function newwindow2()
{
win = window.open("upgrate_template.php", "win", "height=2500,
width=2500");
}
//-->
</script>

```

```
</body>
</html>
```

## Лістинг Б.4 - Find.php

```
<?php
    session_start();
    header("Content-Type: text/html; charset=windows-1251");
?>
<html>
<head>
    <link rel="stylesheet" type="text/css" href="style.css">
    <title>Пошук</title>
    <style>
    </style>
</head>
<body>
    <div id="main">
        <br>
        <input type="checkbox" id="nav-toggle" hidden>
        <label for="nav-toggle" class="nav-toggle" onclick hidden></label>
        <nav class="nav">
            <label for="nav-toggle" class="nav-toggle" onclick></label>
            <?php
            echo "<h2 class='logo'>".$_SESSION["login"]."</h2>"; ?>
            <!-- <h2 class="logo"> Ім'я користувача</h2> -->
            <ul>
                <li><a href="Menu.php">Стани</a>
                <li><a href="Template.php">Шаблони</a>
                <li><a href="Soft.php">програми</a>
            </ul>
        </nav>
        <fieldset style="font-variant:small-caps" class="fieldset2">
        <legend>напаМемпби Пошука</legend>
        <div class="fieldset2">
            <input type="checkbox" пате="Box1">Комп'ютер<br>
            <input type="checkbox" пате="Box2">Відділення<br>
            <input type="checkbox" name="Box3">Шаблон<br>
            <input type="checkbox" name="Box4">Програма<br>
        </div>
        <div id="turn">
            <input type="text" required="required" min=1 maxlength=100
            name="Name">
            <input type="submit" value='^CKamb" name="find">
        </div>
        </fieldset>
    <?php
        if (isset($_POST['Name']) && (isset($_POST['Box1']) ||
isset($_POST['Box2']) || isset($_POST['Box3']) || isset($_POST['Box4'])))
            header("Content-Type: text/html; charset=windows-1251");
            $db = mysqli_connect("localhost","root","") or die ("Неможливе
підключення до MySQL");
            mysqli_select_db($db, "mydb") or die ("Неможливо відкрити таблицю з
даними");
            mysqli_query($db,"SET NAMES 'UTF-8'");
            $res = mysqli_query($db, "select distinct Name as 'Ім'я шаблону',
                Status as 'Статус', Computer_ID as
                'Комп'ютер' from template");

            echo "<table>";
            echo "<caption align='center'> Пошук
                </caption>";
            $row = mysqli_fetch_assoc($res);
            foreach ($row as $key => $value) {
                echo("<th>$key</th>");
```

```

    }
    echo("<tr id = '0'>");
    foreach ($row as $key => $value) {
        if (is_numeric($value)) { echo("<td> $value</td>");
        }else{ echo ("<td>$value</td>");
        }
    }
    echo("</tr>");
    $id=1;
    while($row = mysqli_fetch_assoc($res)){
        echo("<tr id=".$id.">");
        foreach ($row as $key => $value) {
            if (is_numeric($value)) { echo("<td> $value</td>");
            }else{ echo ("<td>$value</td>");
            }
        }
        $id++;
        echo("</tr>");
    } echo "</table>";
?>
</div>
</body>
</html>

```

### Лістинг Б.5 - Connect.php

```

<?php
$db = mysqli_connect("localhost","root","") or die ("Неможливе підключення
до MySQL");
mysqli_select_db($db, "mydb") or die ("Неможливо відкрити таблицю з
даними");
$session = новий SNMP(SNMP::VERSION_1, "127.0.0.1", "public");
$ifType = $session->walk(".iso.org.dod.internet.mgmt.mib-
2.host.hrSWInstalled", TRUE);
mysqli_query($db,"SET NAMES 'ANSI'");
$Name='';
function remove_snmp_types( $in_str )
{
    $tmp_str = ltrim($in_str, "STRING: ");
    return str_replace ( ' ','', $tmp_str);
}
$str = '';
for ( $i=1; $i <= (count($ifType)-2)/5; $i++)
{
    $Name=remove_snmp_types( $ifType['3.1.2.'.$i] );
    if (strstr($Name, "Microsoft") == true)
    $str = 'INSERT INTO soft (Name, Comment, SoftType, TypePatern, Version,
LicenseINFO_ID)
        VALUES ("'.$Name.'"," ", "Application", "SubString",1,1)';
    else
    $str = 'INSERT INTO soft (Name, Comment, SoftType, TypePatern, Version,
LicenseINFO_ID)
        VALUES ("'.$Name.'"," ", "Application", "FullString",1,2)';
    $result = mysqli_real_query($db,$str) або die ( mysqli_error($db));
}
if ($result == TRUE){
    echo "Інформацію занесено до бази даних";
}else{
    echo "Інформація не занесена до бази даних";
}
?>

```

### Лістинг Б.6 - Add\_computer.php

```

<?php

```

```

        session_start();
        header("Content-Type: text/html; charset=windows-1251");
    ?>
<html>
<head>
    <link rel="stylesheet" type="text/css" href="style.css">
    <script type="text/javascript" src="Javas.js"></script>
    <meta http-equiv="Content-Type" content="text/html; charset=windows-
1251" />
    <Title>Додати новий комп'ютер</Title>
    <style>
    # fieldset2 {
        height:93%;
        width:93%;
    }
    <meta charset="UTF-8">
    </style>
</head>
<body>
<div id="main">
<fieldset style="font-variant:small-caps" id="fieldset2">
<legend>Додати новий комп'ютер</legend>
    <div class="fieldset2">
        Ім'я комп'ютера<Br>
        <input type="text" required="required" min=1 maxlength=100
name="Name"><br>
        Відділення<Br>
        <?php
            $db = mysqli_connect("localhost","root","") or die
("Неможливе підключення до MySQL");
            mysqli_select_db($db, "mydb") or die ("Неможливо
відкрити таблицю з даними");
            $res="select ID від division";
            $result = mysqli_real_query($db,$res) або die (
mysqli_error($db));
            echo "<form>";
            echo "<select name='s1'>";
            while (($row=mysqli_fetch_array($result))!=null)
            {
                echo "<option>".$row[0]."</option>";
            }
            echo "</select>";
            echo "</form>";
            ?>
        <br>
        <input type="submit" value='^обаBumb" пате="Додати"></лпри^
        </div>
    </fieldset>
</div>
<?php
    if (isset($_POST["Додати"])){
        $Com=$_POST["Name"];
        $Div=$_POST["Div"];
        $db = mysqli_connect("localhost","root","") or die ("Неможливе
підключення до MySQL");
        mysqli_select_db($db, "mydb") or die ("Неможливо відкрити таблицю з
даними");
        $str = 'Input computer into incom (Comment, Division_ID)
VALUES ("'.$Com.'", "'.$Div.'")';
        $result = mysqli_real_query($db,$str) або die ( mysqli_error($db));
        if ($result==true)
            echo "Запит виконаний";
        else

```

```

        echo "ПОМИЛКА";
    }
?>
</body>
</html>

```

### Лістинг Б.7 - Dell\_computer.php

```

<?php
    session_start();
    header("Content-Type: text/html; charset=windows-1251"); ?> <html> <head>
    <link rel="stylesheet" type="text/css" href="style.css">
    <script type="text/javascript" src="Javas.js"></script>
    <meta http-equiv="Content-Type" content="text/html; charset=windows-
1251" />
    <title>Видалити комп'ютер</title>
    <style>
    # fieldset2 {
        height:93%;
        width:93%;
    }
    <meta charset="UTF-8">
    </style>
    </head>
    <body>
    <div id="main">
    <fieldset style="font-variant:small-caps" id="fieldset2">
    <legend>Видалити комп'ютер</legend>
    <div class="fieldset2">
        Ім'я комп'ютера<Br>
        <input type="text" required="required" min=1 maxlength=100
name="Name"><br>
        Відділення<Br>
        <input type="text" required="required" min=1 name="div"><br>
        <br>
        <input type="submit" value="Видалити" name="Видалити"
onClick='self.close()'></input>
    </div>
    </fieldset>
    </div>
    <?php
        if (isset($_POST["Видалити"])){
            $Com=$_POST["Name"];
            $Div=$_POST["div"];
            $db = mysqli_connect("localhost","root","") or die ("Неможливе
підключення до MySQL");
            mysqli_select_db($db, "mydb") or die ("Неможливо відкрити таблицю з
даними");
            $str = 'Delete from computer where Comment = '.$Com.'
                and Division_ID = '.$Div.'';
            $result = mysqli_real_query($db,$str) або die ( mysqli_error($db));
            if ($result==true)
                echo "Запит виконаний";
            else
                echo "ПОМИЛКА";
        }
    ?>
    </body>
</html>

```

### Лістинг Б.8 - Upgrade\_template.php

```

<?php
    session_start();

```

```

header("Content-Type: text/html; charset=windows-1251"); ?> <html> <head>
<link rel="stylesheet" type="text/css" href="style.css">
<script type="text/javascript" src="Javas.js"></script>
<meta http-equiv="Content-Type" content="text/html; charset=windows-
1251" />
<title>Додати новий комп'ютер</title>
<style>
# fieldset2 {
height:93%;
width:93%;
}
<meta charset="UTF-8">
</style>
</head>
<body>
<div id="main">
<fieldset style="font-variant:small-caps" id="fieldset2">
<legend>Інформація про шаблон</legend>
<div class="fieldset2">
Ім'я шаблону<br>
<input type="text" required="required" min=1 maxlength=100
name="Name"></input>
<input type="submit" value="Знайти" name="Додати"></input><br>
<?php
if (isset($_POST["Додати"])){
$Nam=$_POST["Name"];
$db = mysqli_connect("localhost","root","") or die
("Неможливе підключення до MySQL");
mysqli_select_db($db, "mydb") or die ("Неможливо відкрити
таблицю з даними");

$res = mysqli_query($db, 'SELECT Name as "Ім'я шаблону", Comment as
"Коментар", Programm as "Програми", DateUninstall as "Дата видалення",
DateInstall as "Дата установки" від template where Name='. $ Nam. ');
echo "<table>";
echo "<caption align='center'> Шаблони </caption>";
$row = mysqli_fetch_assoc ($ res); foreach ($row as $key => $value) { echo
("<th> $key </th>");
}
echo("<tr id = '0'>");
foreach ($row as $key => $value) { if (is numeric($value)) { echo("<td>
$value</td>"); }else{ echo ("<td>$value</td>");
}
}
echo("</tr>");
$id=1;
while($row = mysqli_fetch_assoc($res)){ echo("<tr id=".$id.">");
foreach ($row as $key => $value) { if (is numeric($value)) { echo("<td>
$value</td>"); }else{ echo ("<td>$value</td>");
}
}
$id++;
echo("</tr>");
} echo "</table>";
}
?>
</div>
</fieldset>
</div>
</body>
</html>

```

## Лістинг Б.9 - Style.css

```
#main {
    position: absolute; width : 100%;
    height: 100%;
    background: radial-gradient(at top, #FFFFFF, #A7CECC); } #labe {
    position: absolute; text-align: left; height: 100%;
    width: 20%; margin-left: 80%;
}
.nav {
    width: 320px;
    min-width: 320px;
    height: 100%;
    position: fixed;
    top: 0;
    bottom: 0;
    margin: 0;
    left: -320px;
    padding: 15px 20px;
    -webkit-transition: left 0.3s;
    -Moz-transition: left 0.3s; transition: left 0.3s;
    background: #16a085; z-index: 2000;
}
.nav-toggle { position: absolute; left: 320px;
    top: 1em;
    padding: 0.5em;
    background: inherit; color: #dadada;
    cursor: pointer; font-size: 1.2em; line-height: 1; z-index: 2001;
    -webkit-transition: color .25s ease-in-out;
    -Moz-transition: color .25s ease-in-out; transition: color .25s ease-in-out;
}
.nav-toggle:after { content: '\2630';
    text-decoration: none;
}
.nav-toggle:hover { color: #f4f4f4;
}
[id='nav-toggle'] { position: absolute; display: none;
}
[id='nav-toggle']:checked ~ .nav > .nav-toggle {
    left: auto;
    right: 2px;
    top: 1em;
}
[id='nav-toggle']:checked ~ .nav {
    left: 0;
    box-shadow: 4px 0px 20px 0px rgba (0,0,0, 0.5);
    - moz-box-shadow: 4px 0px 20px 0px rgba (0,0,0, 0.5);
    - webkit-box-shadow: 4px 0px 20px 0px rgba (0,0,0, 0.5);
    - verflow-y: auto;
}
[id='nav-toggle']:checked ~ main > article {
    - webkit-transform: translateX(320px);
    - moz-transform: translateX(320px);
    transform: translateX(320px);
}
[id='nav-toggle']:checked ~ .nav > .nav-toggle:after { content: '\2715';
}
body {
    -webkit-animation: bugfix infinite 1s;
}
@-webkit-keyframes bugfix { to {
    padding: 0;
}
```



```

    }
}
@media screen and (min-width: 320px) {
    html, body { margin: 0;
        overflow-x: hidden;
    }
}
@media screen and (max-width: 320px) {
    html, body {
        margin: 0;
        overflow-x: hidden;
    }
    .nav {
        width: 100%;
        box-shadow: none
    }
}
.nav h2 {
    width: 90%;
    padding: 0;
    margin: 10px 0;
    text-align: center;
    text-shadow: rgba(255, 255, 255,.1) -1px -1px 1px, rgba(0, 0, 0,.5)
1px 1px
1px;
    font-size: 1.3em;
    line-height: 1.3em;
    opacity: 0;
    transform: scale(0.1, 0.1);
    - ms-transform: scale(0.1, 0.1);
    - moz-transform: scale(0.1, 0.1);
    - webkit-transform: scale(0.1, 0.1);
    - ransform-origin: 0% 0%;
    - ms-transform-origin: 0% 0%;
    - moz-transform-origin: 0% 0%;
    - webkit-transform-origin: 0% 0%;
    transition: opacity 0.8s, transform0.8s;
    - ms-transition: opacity 0.8s, -ms-transform 0.8s;
    - moz-transition: opacity 0.8s, -moz-transform 0.8s;
    - webkit-transition: opacity 0.8s, -webkit-transform 0.8s;
}
.nav h2 a {
    color: #dadada;
    text-decoration: none; text-transform: uppercase;
}
[id='nav-toggle']:checked ~ .nav h2 { opacity: 1;
    transform: scale(1, 1);
    - ms-transform: scale(1, 1);
    - moz-transform: scale(1, 1);
    - webkit-transform: scale(1, 1);
}
.nav > ul { display: block; margin: 0;
    padding: 0;
    list-style: none;
}
.nav > ul > li { line-height: 2.5; opacity: 0;
    - webkit-transform: translateX(-50%);
    - moz-transform: translateX(-50%);
    - ms-transform: translateX(-50%);
    transform: translateX(-50%);
    - webkit-transition: opacity .5s .1s,-webkit-transform .5s.1s;
    - moz-transition: opacity .5s .1s,-moz-transform .5s.1s;
    - ms-transition: opacity .5s .1s,-ms-transform .5s.1s;
}

```

```

        transition: opacity .5s .1s,transform.5s.1s;
    }
[id='nav-toggle']:checked ~ .nav > ul > li {
    opacity: 1;
    - webkit-transform: translateX(0);
    - moz-transform: translateX(0);
    - ms-transform: translateX(0); transform: translateX(0);
}
.nav > ul > li:nth-child(2) { -webkit-transition: opacity .5s .2s,-webkit-
transform .5s.2s;
    transition: opacity .5s .2s, transform .5s.2s;
}
.nav > ul > li:nth-child(3) { -webkit-transition: opacity .5s .3s,-webkit-
transform .5s.3s;
    transition: opacity .5s .3s, transform .5s.3s;
}
.nav > ul > li:nth-child(4) { -webkit-transition: opacity .5s .4s,-webkit-
transform .5s.4s;
    transition: opacity .5s .4s, transform .5s.4s;
}
.nav > ul > li:nth-child(5) {
    - webkit-transition: opacity .5s .5s,-webkit-transform .5s.5s;
    transition: opacity .5s .5s, transform .5s.5s;
}
.nav > ul > li:nth-child(6) {
    - webkit-transition: opacity .5s .6s,-webkit-transform .5s.6s;
    transition: opacity .5s .6s, transform .5s.6s;
}
.nav > ul > li:nth-child(7) {
    - webkit-transition: opacity .5s .7s,-webkit-transform .5s.7s;
    transition: opacity .5s .7s, transform .5s.7s;
}
.nav > ul > li > a {
    display: inline-block;
    position: relative;
    padding: 0;
    font-family: 'Open Sans', sans-serif;
    font-weight: 300;
    font-size: 1.2em;
    color: #dadada;
    width: 100%;
    text-decoration: none;
    /* плавний перехід */
    - webkit-transition: color .5s ease, padding .5s ease;
    - moz-transition: color .5s ease, padding .5s ease;
    transition: color .5s ease, padding .5s ease;
}
.nav > ul > li > a:hover,
.nav > ul > li > a:focus {
    color: white;
    padding-left: 15px;
}
.nav > ul > li > a:before {
    content: '';
    display: block;
    position: absolute;
    right: 0;
    bottom: 0;
    height: 1px;
    width: 100%;
    - webkit-transition: width 0s ease;
    transition: width 0s ease;
}
}

```

```

.nav > ul > li > a:after {
  content: '';
  display: block;
  position: absolute;
  left: 0;
  bottom: 0;
  height: 1px;
  width: 100%;
  background: #3bc1a0;
  - webkit-transition: width .5s ease;
  transition: width .5s ease;
}
.nav > ul > li > a:hover:before {
  width: 0%;
  background: #3bc1a0;
  - webkit-transition: width .5s ease;
  transition: width .5s ease;
}
.nav > ul > li > a:hover:after { width: 0%;
  background: transparent;
  - webkit-transition: width 0s ease; transition: width 0s ease;
}
.mask-content { display: block; position: fixed;
  top: 0;
  left: 0;
  z-index: 1000;
  width: 100%;
  height: 100%;
  background-color: rgba(0, 0, 0,0.4);
  visibility: hidden;
  opacity: 0;
}
[id='nav-toggle']:checked ~ .mask-content { visibility: visible;
  opacity: 1;
  - webkit-transition: opacity .5s, visibility .5s; transition: opacity
  .5s, visibility .5s;
}
#body {
  background:radial-gradient(at top, #FFFFFF, #A7CECC);
  width:100%;
  height:100%;
  left : 0;
  top: 0;
  position: absolute; }
#info {
  top: 2%;
  left: 45%;
  position: absolute;
  width: 220px;
  height: 190px;
  background-color: white;
  border: 2px solid #c1ffd8;
  margin: 0 auto;
  text-align: center;
}
#result {
  text-align: center; font-size: x-large; font-weight: 700; }
.words
{ color: black;
}
#button1 {
  width: 200px;
  height:40px;
}

```

```

        border: 1px solid;
    }
    table {
        width: 100%;
        font-family: "Lucida Sans Unicode", "Lucida Grande", Sans-Serif; border-
collapse: collapse;
        color: #686461;
    }
    caption {
        padding: 10px;
        color: white;
        background: #8FD4C1;
        font-size: 18px;
        text-align: left; font-weight: bold;
    }
    th {
        border-bottom: 3px solid #B9B29F;
        padding: 10px;
        text-align: left;
    }
    td { padding: 10px;
    }
    tr:nth-child(odd) { background: white;
    }
    tr:nth-child(even) { background: #E8E6D1;
    }
    #информация
    {
margin-left: 2.1%;
border:1px solid black;
left : 2.1%;
background-color: white;
    }
    #but
    {
        left : 2.1%;
        margin-left: 2.1%;
    }
    #Scaning
    {
margin-left: 5.1%;
    }
    a.button7 {
        font-weight: 700;
        color: white;
        text-decoration: none;
        padding: .8em 1em calc(.8em + 3px);
        border-radius: 3px;
        background: rgb(64,199,129);
        box-shadow: 0 -3px rgb(53,167,110) inset;
        transition: 0.2s;
        margin-left: 2.2%;
    }
    a.button7:hover { background: rgb(53, 167, 110);}
    a.button7:active {
        background: rgb(33,147,90);
        box-shadow: 0 3px rgb(33,147,90) inset;
    }
    .fieldset2{
margin-left: 2.2%;
width:30%;
height:40%;
    }

```

```

#turn {
  border:1px solid black;
  width:25%;
  height:5%;
}
#openWind{ margin-left: 2.2%;
}

```

## Лістинг Б.10 - Java.js

```

function createXmlHttpRequestObject()
{ // функція створює та повертає об'єкт типу AJAX -запит
  var xmlHttp;
  try{ xmlHttp=new XMLHttpRequest();
  }
  catch (e) {try {
      xmlHttp=new ActiveXObject("MSXML2.XMLHTTP"); } catch (e)
    {try {
      xmlHttp=new ActiveXObject("Microsoft.XMLHTTP");
    }
    catch(e){} } }
  if(!xmlHttp){
    alert("Не вдалося створити об'єкт XMLHttpRequest
XMLHttpRequest"); }
  return xmlHttp;
}
function Init() {
  // створення об'єкта до виконання AJAX -запроса
  // і призначення оброблювача відповіді
  var xmlHttp = createXmlHttpRequestObject();
  xmlHttp.onreadystatechange = function(){ Receive(xmlHttp);
  } Send(xmlHttp);
}
function Receive(xmlHttp) {
  // обробник відгуку web -сервера, призначається функцією Init
  // працює відповідно до завдання (read.jpg)
  if (xmlHttp.readyState != 4) return;
  var response = getResponseText(xmlHttp); if (response == 1) {
    window.location.href = "Menu.php";
  } else { alert ("Невірні ім'я та/або пароль");
  } }
function getResponseText(xmlHttp)
{
  // Перевірка готовності результату
  // Повертає текст відгуку сервера на запит
  var responseText = false;
  console.log(xmlHttp.responseXML );
  if (xmlHttp.status == 200) {
    return xmlHttp.responseText ;
  }
  else {
    if (xmlHttp.status == 404) {
      alert("Requested URL не існує"); }
    else {
      alert("Error: status code is " + xmlHttp.status );
    }
  }
  return responseText;
}
function Send(xmlHttp) {
  // Функція посилає AJAX -запит. Параметри передаються методом POST
  // обчислюємо рядок параметрів
  // посилаємо заголовок, який передає серверу тип повідомлення
  var log=document.getElementById("login").value;

```

```

var pas=document.getElementById("password").value;
xmlHttp.open("POST", "Login.php", true);
xmlHttp.setRequestHeader("Content-type", "application/x-www-form-
urlencoded");
var requestStr = "login=" + log + "&password=" + pas;
xmlHttp.send(requestStr);
}

```

## Лістинг Б.11 - Add\_template.php

```

<?php
    session_start();
    header("Content-Type: text/html; charset=windows-1251");
?>
<html>
<head>
    <link rel="stylesheet" type="text/css" href="style.css">
    <script type="text/javascript" src="Javas.js"></script>
    <meta http-equiv="Content-Type" content="text/html; charset=windows-
1251"/>
    <title>Додати новий шаблон</title>
    <style>
    # fieldset2 {
        height:93%;
        width:93%;
    }
    <meta charset="UTF-8">
    </style>
</head>
<body>
<div id="main">
<fieldset style="font-variant:small-caps" id="fieldset2">
<legend>Створити шаблон</legend>
    <div class="fieldset2"> Ім'я шаблону^^
        <input type="text" required="required" min=1 maxlength=100
name="Тем"><br>
        Коментар<Br>
        <input type="text" min=1 maxlength=100 name="Com"><br>
        Програми<Br>
        <select size="10" multiple name="hero[]">
        <?php
            $db = mysqli_connect("localhost","root","") or die
("Неможливе підключення до MySQL");
            mysqli_select_db($db, "mydb") or die ("Неможливо
відкрити
таблицю з даними");
            $res="select Name from soft";
            $result = mysqli_real_query($db,$res) або die (
mysqli_error($db));
            while (($row=mysqli_fetch_array($result))!=null) {
                echo "<option>".$row[0]."</option>"; }
        ?>
        </select>
        <br>
        Комп'ютер<Br>
        <select name="computer">
        <?php
            $db = mysqli_connect("localhost","root","") or die
("Неможливе підключення до MySQL");
            mysqli_select_db($db, "mydb") or die ("Неможливо
відкрити
таблицю з даними");
            $res="select Comment from computer";
            $result = mysqli_real_query($db,$res) або die (

```

```

mysqli_error($db));
        while (($row=mysql_fetch_array($result))!=null) {
            echo "<option>".$row[0]."</option>"; }
        ?>
    </select>
    <br>
    <br>
    <input type="submit" value='^обаBUTb" пате="Додати"
onClick='self.close()'></input>
</div>
</fieldset>
</div>
<?php
    if (isset($POST["Додатиb"])){
        $Tem=$_POST["Tem"];
        $Com=$_POST["Com"];
        $Her=$_POST["hero[]"];
        $Computer=$_POST["computer"];
        $db = mysqli connect("localhost","root","") or die ("Неможливе
підключення до MySQL");
        mysqli select db($db, "mydb") or die ("Неможливо відкрити таблицю з
даними"); $today = date("Ymd");
        $computer='SELECT Computer_ID FROM template where computer.Comment='.$Com.'
            and Computer_ID=computer.ID';
        $soft='SELECT SOFT_ID from template where soft.Name ='.$Her.' and
            SOFT_ID=Soft.ID';
        $str = ' template INSERT INTO (Name, Comment, Programm, DateUninstall,
DateInstal, Version, Status, Computer_ID, SOFT_ID)
            VALUES
('.$Tem.',"'.$Com.',"'.$Her.',"'.$today.',"','',1,0,"'.$computer.
',"'$soft.'') ;
        $result = mysqli_real_query($db,$str) або die ( mysqli_error($db));
        if ($result==true)
            echo "Запит виконаний";
        else echo "ПОМИЛКА";
    }
    ?>
</body>
</html>

```

### Лістинг Б.12 - Dell\_Template.php

```

<?php
    session_start();
    header("Content-Type: text/html; charset=windows-1251");
    ?>
<html>
<head>
    <link rel="stylesheet" type="text/css" href="style.css">
    <script type="text/javascript" src="Javas.js"></script>
    <meta http-equiv="Content-Type" content="text/html; charset=windows-1251"
/>
    <Title>Видалити шаблон</Title>
    <style>
    #fieldset2 {
        height:93%;
        width:93%;
    }
    <meta charset="UTF-8">
    </style>
</head>
<body>
<div id = "main">
<fieldset style="font-variant:small-caps" id="fieldset2">

```

```

<legend>Видалити шаблон</legend>
  <div class="fieldset2">
    Ім'я шаблону<Br>
      <input type="text" required="required" min=1 maxlength=100
name="Name"><br>
      <br>
      <input type="submit" value="Видалити" name="Видалити"
onClick='self.close()'></input>
    </div>
  </fieldset>
</div>
<?php
  if (isset($_POST["Видалити"])){
    $Tem=$_POST["Name"];
    $db = mysqli_connect("localhost","root","") or die ("Неможливе
підключення до MySQL");
    mysqli_select_db($db, "mydb") or die ("Неможливо відкрити таблицю з
даними");
    $str = 'Delete from template where Comment = '.$Com.'';
    $result = mysqli_real_query($db,$str) або die ( mysqli_error($db));
    if ($result==true)
      echo "Запит виконаний";
    else echo "ПОМИЛКА";
  }
?>
</body>

```