

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: *Комп'ютеризована система контролю та обліку
навчальних занять*

Виконала: студентка IV курсу, групи СІс-44
спеціальності 123 «Комп'ютерна інженерія»

(шифр і назва спеціальності)

(підпис)

Українець У.В.

(прізвище та ініціали)

Керівник

(підпис)

Тиш Є.В.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Луцик Н.С.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Осухівська Г.М.

(прізвище та ініціали)

Рецензент

(підпис)

Загородна Н.В.

(прізвище та ініціали)

Тернопіль
2021

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)

« »

2021 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня бакалавр
(назва освітнього ступеня)

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

студенту Українець Уляні Володимирівній
(прізвище, ім'я, по батькові)

1. Тема роботи Комп'ютеризована система контролю та обліку навчальних занять

Керівник роботи Тиш Євгенія Володимирівна, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «10» лютого 2021 року № 4.7-97

2. Термін подання студентом завершеної роботи 27.06.2021 р.

3. Вихідні дані до роботи Типи занять, інформація про вчителів, дані про студентів і групи технології розробки комп'ютеризованих систем

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

Вступ. 1. Аналіз вимог до комп'ютеризованої системи та організаційної структури закладів освіти. 2. Проектування мережевої інфраструктури та програмного забезпечення комп'ютеризованої системи 3. Розробка програмного забезпечення комп'ютеризованої системи контролю та обліку навчальних занять. 4. Безпека життєдіяльності, основи охорони праці. Висновки

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

1. Фізична топологія комп'ютерної мережі (1 поверх).

2. Фізична топологія комп'ютерної мережі (2 поверх).

3. Діаграма прецедентів при проектуванні комп'ютеризованої системи.

4. Архітектура програмного забезпечення комп'ютеризованої системи.

5. ER-діаграма бази даних.

6. Алгоритм роботи програмного забезпечення комп'ютеризованої 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., проф. каф. МТ</i>		

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	<i>Розробка та аналіз вимог технічного завдання</i>	<i>10.02-21.02.2021</i>	
2	<i>Аналіз організаційної структури закладу освіти</i>	<i>21.02-07.03.2021</i>	
3	<i>Проектування мережевої інфраструктури комп'ютеризованої системи</i>	<i>08.03-20.03.2021</i>	
4	<i>Проектування схеми бази даних та архітектури програмного забезпечення</i>	<i>20.03-26.03.2021</i>	
5	<i>Реалізація програмного забезпечення комп'ютерної системи</i>	<i>27.03-10.04.2021</i>	
6	<i>Розробка інструкцій з налаштування параметрів комп'ютеризованої системи контролю та обліку навчальних занять</i>	<i>10.04-04.05.2021</i>	
7	<i>Безпека життєдіяльності, основи охорони праці</i>	<i>04.05-20.05.2021</i>	
8	<i>Оформлення кваліфікаційної роботи</i>	<i>27.05-10.06.2021</i>	
9	<i>Попередній захист кваліфікаційної роботи</i>	<i>10.06-20.06.2021</i>	
10	<i>Захист кваліфікаційної роботи</i>	<i>21.06-27.06.2021</i>	

Студент

(підпис)

Українець Уляна Володимирівна

(прізвище та ініціали)

Керівник роботи

(підпис)

Тиш Євгенія Володимирівна

(прізвище та ініціали)

АНОТАЦІЯ

Комп'ютеризована система контролю та обліку навчальних занять//
Кваліфікаційна робота на здобуття освітнього ступеня бакалавр // Українець
Уляна Володимирівна // ТНТУ, спеціальність 123 «Комп'ютерна інженерія»//
Тернопіль, 2021 // с.– 65, рис. – 38 , табл. – 10, аркушів А1 – 6, бібліогр. – 19.

Ключові слова: навчання, заняття, облік, контроль, система.

У кваліфікаційній роботі розроблено проект комп'ютеризованої системи контролю та обліку навчальних занять до складу якої входять мережева інфраструктура закладу освіти та програмне забезпечення управління процесом контролю та обліку занять.

Мережева інфраструктура побудована на основі фізичної і логічної топології комп'ютерної мережі у вигляді розширеної зірки. Забезпечення зв'язку між робочими станціями і сервером бази даних реалізовано за допомогою активного комутаційного обладнання фірми D-Link. У проекті комп'ютеризованої системи розроблено схему комутаційних з'єднань вузлів комп'ютерної мережі, IP-адресну схему, що дозволяє повністю забезпечити мережеву інфраструктуру для управління процесом контролю та обліку навчальних занять.

Керування процесом контролю та обліку навчальних занять забезпечено на рівні програмного забезпечення, розроблено на основі архітектури клієнт-сервер засобами мови програмування С#. База даних побудована на основі реляційного підходу та з використанням системи керування базами даних MS SQL Server.

ABSTRACT

Computer-aided system for academic studies control and records // Bachelor's thesis // Ukrainets Uliana Volodymyrivna // TNTU, speciality 123 «Computer engineering»// Ternopil, 2021 // p.– 65 , fig. – 38 , tab. – 10, posters A1 – 6, ref. – 19.

Keywords: learning, class, monitoring, accounting, system.

In the qualification work, a project of a computerized system of control and accounting of classes was developed, which includes the network infrastructure of the educational institution and software for managing the process of control and accounting of classes.

The network infrastructure is based on the physical and logical topology of a computer network in the form of an extended star. Communication between workstations and the database server is implemented using active D-Link switching equipment. In the project of the computerized system the scheme of switching connections of nodes of a computer network, the IP-address scheme allowing to provide completely a network infrastructure for management of process of the control and the account of training sessions is developed.

Management of the process of control and accounting of training sessions is provided at the software level, developed on the basis of client-server architecture by means of C # programming language. The database is based on a relational approach and using the MS SQL Server database management system.

ЗМІСТ

ПЕРЕЛІК ОСНОВНИХ УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ І СКОРОЧЕНЬ	8
ВСТУП	9
РОЗДІЛ 1 АНАЛІЗ ВИМОГ ДО КОМП'ЮТЕРИЗОВАНОЇ СИСТЕМИ ТА ОРГАНІЗАЦІЙНОЇ СТРУКТУРИ ЗАКЛАДІВ ОСВІТИ	10
1.1 Аналіз вимог щодо проектування комп'ютеризованої системи контролю та обліку навчальних занять	10
1.2 Аналіз типової організаційної структури закладів освіти	18
РОЗДІЛ 2 ПРОЕКТУВАННЯ МЕРЕЖЕВОЇ ІНФРАСТРУКТУРИ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРИЗОВАНОЇ СИСТЕМИ	20
2.1 Проектування та налаштування мережевої інфраструктури комп'ютеризованої системи контролю та обліку навчальних занять	20
2.2 Виявлення сутностей при проектуванні системи контролю та обліку навчальних занять	28
2.3 Проектування схеми бази даних комп'ютеризованої системи обліку та контролю навчальних занять	30
2.4 Розробка алгоритму роботи функціонування комп'ютеризованої системи контролю та обліку навчальних занять	35
2.5 Обґрунтування апаратного забезпечення комп'ютеризованої системи контролю та обліку навчальних занять	38
РОЗДІЛ 3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРИЗОВАНОЇ СИСТЕМИ КОНТРОЛЮ ТА ОБЛІКУ НАВЧАЛЬНИХ ЗАНЯТЬ	42

					КС КРБ 123.181.00.00 ПЗ			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>Лім.</i>	<i>Арк.</i>	<i>Аркуші</i>
Розроб.		Українець У.В.			<i>Комп'ютеризована система контролю та обліку навчальних занять</i>		6	
Перевір.		Тиш Є.В.				<i>ТНТУ, каф. КС, гр. СІс-44</i>		
Реценз.								
Н. Контр.		Луцик Н.С.						
Затверд.		Осухівська Г.М.						

3.1	Реалізація засобів доступу до даних.....	42
3.2	Розробка і тестування програмного забезпечення комп'ютеризованої системи контролю та обліку занять	47
3.3	Організація доступу до джерел даних із використанням технології ADO.NET, СКБД SQL Server та Visual Studio.....	55
РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ		59
4.1	Суть та зміст управління охороною праці	59
4.2	Підбирання оптимальних параметрів мікроклімату на робочих місцях з ПК	63
ВИСНОВКИ		66
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....		67
Додаток А. Технічне завдання		
Додаток Б. Фізична топологія комп'ютерної мережі закладу освіти		
Додаток В. Лістинг програмного додатку		
Додаток Г. Скрипт формування бази даних		

					КС КРБ 123.181.00.00 ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ОСНОВНИХ УМОВНИХ ПОЗНАЧЕНЬ,
СИМВОЛІВ І СКОРОЧЕНЬ

БД	База даних
КС	Комп'ютерна система
КМ	Комп'ютерна мережа
ПЗ	Програмне забезпечення

					<i>КС КРБ 123.181.00.00 ПЗ</i>	Арк.
						8
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

ВСТУП

Інтенсивне впровадження інформаційних технологій у навчальний процес дозволяє підвищити як якість знань здобувачів освіти, так і посилити контроль та облік навчальних занять. Сучасний розвиток платформ дистанційного та онлайн навчання дає змогу учням і студентам значно розширити свій кругозір, здобути нові знання і вміння, які часто безпосередньо не пов'язані з прямою спеціальністю та майбутньою професією. Серед таких платформ варто відмітити Coursera, Edex та ряд інших приватних курсів різного спрямування. Однак, хоч курси і викладають провідні спеціалісти у своїй галузі, все ж студентам перших курсів, а особливо учням, складно себе самодисциплінувати. Тому традиційний процес навчання і надалі відіграє важливу роль у формуванні вмінь та навиків майбутніх студентів та фахівців.

У сучасних умовах пандемії, застосування інформаційних технологій при навчанні майбутнього покоління використовується дуже інтенсивно, однак інструментальні засоби все ж не є досконалими, оскільки не в змозі так швидко адаптуватись під конкретні умови та процеси навчального процесу. Так, наприклад, розробки компанії Microsoft (продукт Teams) і Google (Google Meets) хоч і дозволяють забезпечувати процес онлайн навчання, однак в них відсутні засоби для обліку та фіксації успішності студентів. Платформи електронного навчання Atutor і Moodle хоч і містять інтегровані журнали відвідування та успішності студентів, все ж їх практичне використання доволі не зручне та не завжди зрозуміле кінцевому користувачу. Варто відмітити, що сьогодні в Україні на початковому етапі впроваджується система «Єдина школа», яка орієнтована на підтримку навчального процесу з врахуванням особливостей навчального процесу. У ній інтегровано функціональність електронного журналу вчителя, однак система все ще перебуває у процесі тестування та модернізації. Тому актуальною задачею при провадженні освітнього процесу у школах, училищах і коледжах є розробка і впровадження локальної комп'ютеризованої системи контролю та обліку навчальних занять.

					КС КРБ 123.181.00.00 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1 АНАЛІЗ ВИМОГ ДО КОМП'ЮТЕРИЗОВАНОЇ СИСТЕМИ ТА ОРГАНІЗАЦІЙНОЇ СТРУКТУРИ ЗАКЛАДІВ ОСВІТИ

1.1 Аналіз вимог щодо проектування комп'ютеризованої системи контролю та обліку навчальних занять

Основне призначення проектованої комп'ютеризованої системи полягає у забезпеченні процесів обліку та контролю за навчальними заняттями, зокрема, відвідуваності занять учнями або студентами, їх успішності та здатності формування розкладу протягом визначеного періоду часу.

При проектуванні комп'ютеризованої системи контролю та обліку навчальних занять необхідно дослідити та/або спроектувати існуючу апаратну і комунікаційну інфраструктуру, спроектувати схему бази даних і клієнтське програмне забезпечення для зручного використання програмного забезпечення системи.

Інформаційна інфраструктура повинна базуватись на організаційній структурі закладу освіти і як наслідок локальна комп'ютерна мережа має включати необхідні апаратні пристрої для забезпечення функцій комунікації між клієнтськими станціями і сервером баз даних.

Програмне забезпечення комп'ютеризованої системи має задовольняти основним функціям щодо контролю успішності учнів, а база даних повинна містити довідники щодо груп/класів, студентів/учнів та викладачів навчальних занять.

Основним призначенням системи контролю та обліку навчальних занять є фіксація проведених оффлайн навчань, що дозволить централізовано моніторити стан проведення занять та успішність учнів чи студентів.

					КС КРБ 123.181.00.00 ПЗ			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		Українець У.В.			<i>Аналіз вимог до комп'ютеризованої системи та організаційної структури закладів освіти</i>	<i>Лім.</i>	<i>Арк.</i>	<i>Аркуші</i>
<i>Перевір.</i>		Тиш Є.В.					10	
<i>Реценз.</i>						<i>ТНТУ, каф. КС, гр. СІс-44</i>		
<i>Н. Контр.</i>		Луцик Н.С.						
<i>Затверд.</i>		Осухівська Г.М.						

Реалізація комп'ютеризованої системи підвищить ефективність управління процесами надання освітніх послуг та дозволить приймати рішення для оптимізації та підвищення якості освіти.

Мета створення комп'ютеризованої системи контролю та обліку навчальних занять полягає в автоматизації відповідних процесів для підвищення якості навчального процесу, а також на основі наявних даних прийняття управлінських рішень.

Досягнення мети створення системи можливе шляхом розв'язку наступних задач:

- аналіз організаційної структури закладу освіти;
- на основі особливостей приміщень закладу освіти та фізичного розташування обладнання розробка проекту фізичної топології мережі;
- проектування логічної топології мережі;
- розробка схеми комутації вузлів мережі та IP-адресної схеми;
- обґрунтування вибору апаратного забезпечення комп'ютеризованої системи;
- проектування схеми бази даних для зберігання довідників та автоматизації процесів контролю та обліку навчальних занять;
- реалізація клієнтського програмного забезпечення для роботи з базою даних.

До основних задач і функцій комп'ютеризованої системи контролю та обліку навчальних занять належать забезпечення комунікації між вузлами системи з використанням комп'ютерної мережі та управління процесами забезпечення навчального процесу з використанням прикладного програмного додатку. Сукупно компоненти комп'ютеризованої системи формують складову частину інформаційної інфраструктури закладу освіти при провадженні освітньої діяльності в режимі очного навчання.

Основні задачі, які повинна вирішити комп'ютерна мережа полягають в наступному:

					КС КРБ 123.181.00.00 ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

- забезпечення зв'язку між комп'ютерними класами, учительською кімнатою та дирекцією закладу освіти;
- автоматизація обліку інформації, необхідної учителям;
- організація доступу до мережі Інтернет;
- встановлення та налаштування сервера баз даних;
- забезпечення процесу розмежування прав доступу та контроль над ними при зверненні до спільних ресурсів.

Функціями комп'ютерної мережі є автоматизація процесу навчання шляхом об'єднання комп'ютерів в єдиний інформаційний простір і спільного використання ресурсів.

Функції прикладного програмного додатку полягають у:

- формуванні довідників про базові сутності автоматизації процесів обліку та контролю навчальних занять;
- забезпеченні здатності фіксації проведення навчальних занять;
- управління даними щодо відвідуваності занять;
- забезпеченні можливості оцінювання знань здобувачів освіти;
- формуванні журналів успішності учнів/студентів.

В цілому, вимоги до комп'ютеризованої системи обліку та контролю можна поділити на дві основні частини: вимоги до комп'ютерної мережі та вимоги до прикладного програмного забезпечення управління відповідними процесами.

Вимогами до комп'ютерної мережі закладу освіти є:

- надійність роботи апаратних засобів;
- продуктивність апаратної засобів;
- захищеність обладнання;
- захищеність доступу до ресурсів;
- паралельний доступ до ресурсів;
- розмежування прав доступу;
- часова ефективність;
- ефективність використання ресурсів інформаційної системи.

					КС КРБ 123.181.00.00 ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

До вимог, які в цілому висуваються до прикладного програмного забезпечення комп'ютеризованої системи належать:

- забезпечення функцій вводу, оновлення та видалення інформації про навчальні заняття;
- можливість формування інформаційних довідників про вчителів, групи, заняття, їх види і т.п;
- підтримка реляційної схеми бази даних та відповідної сигнатури;
- клієнт-серверна архітектура клієнтського додатку;
- підтримка рольової політики аутентифікації користувача при використанні програмного забезпечення;
- можливість одночасного доступу до ресурсів бази даних;
- визначена продуктивність надання результатів пошуку та виконання операцій при обліку та контролі навчальних занять.

Вимогами, які висуваються до структури та функціонування комунікаційної інфраструктури комп'ютеризованої системи обліку та контролю навчальних занять належать:

1. Захищеність:

- наявність засобів для обмеженого фізичного доступу до комутаційного обладнання;
- наявність коробів та інших засобів для обмеження доступу до каналів зв'язку;
- програмна захищеність активних компонентів комп'ютерної мережі;

2. Надійність:

- безвідмовність функціонування мережі не менше, ніж 20000 год.;
- резервування каналів живлення активного та пасивного апаратного забезпечення;
- наявність засобів захисту від коливання напруги;
- визначена стабільність функціонування апаратних пристроїв, що входять до складу комп'ютерної мережі;
- відповідність вимогам щодо надійності каналів передачі даних;

					КС КРБ 123.181.00.00 ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

– наявність механізмів авторизації користувачів;

3. Продуктивність:

– швидкість обміну пакетами на рівні 100 Мб/с;

– максимальний розмір оброблюваних даних – 250000 МБ;

4. Відновлюваність та резервування:

– здатність до відновлення при виникненні аварійних ситуацій до 12 год.;

– наявність резервних каналів передачі даних;

Структура комп'ютерної мережі включає в себе:

– структуру, що забезпечує зв'язок між комп'ютерами в межах окремих класів;

– структуру функціонування системи відносно мережі Internet;

– організацію зв'язку між дирекцією та іншими одиницями, що забезпечують навчальний процес.

До функціональних вимог та структури прикладного програмного забезпечення комп'ютеризованої системи належать:

1. Функціональність:

– формування довідників про типи навчальних занять і назви предметів;

– створення довідників студентів, учителів і класних кімнат (приміщень);

– можливість введення, редагування та оновлення інформації про проведені заняття;

– здатність відображати успішність учнів/студентів;

– наявність засобів для впорядкування даних за відповідними полями бази даних;

– наявність механізмів управління авторизованим доступом;

– можливість створення звітів успішності здобувачів освіти;

– облік навантаження викладачів/учителів;

2. Продуктивність:

– час реакції на дії користувача до 1 с;

– час відповіді системи до 1 с;

					КС КРБ 123.181.00.00 ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

- оптимальне використання ресурсів RAM;
- оптимальність використання ресурсів жорсткого диску;
- прозорість та доступність використання функціональності програмного забезпечення;
- налаштування політики прав доступу до системи обліку та контролю навчальних занять;
- гнучкість та масштабованість структури програмного забезпечення комп'ютеризованої системи;

3. Надійність:

- завершеність функціональної та нефункціональної складової програмного забезпечення;
- стійкість роботи при виникненні нештатних ситуацій;
- можливість відновлення та резервного збереження даних;
- наявність додаткових механізмів захисту інформації;

4. Зручність використання:

- використання однотипних елементів управління і оформлення користувацького інтерфейсу;
- документованість підтримки програмного забезпечення;
- забезпечення зрозумілості функціонального призначення елементів керування;
- відповідність результатів очікуванням користувачів;
- простота і зручність використання.

В загальному випадку, структура комп'ютеризованої системи обліку та контролю навчальних занять повинна відображати організаційний зв'язок закладу освіти, процеси моніторингу та обліку проведених навчальних занять.

Зв'язок між компонентами комп'ютеризованої системи організовується за допомогою передачі даних по кабельній інфраструктурі комп'ютерної мережі. Вимоги до організації комп'ютерної мережі та її компонентів описано у відповідних стандартах ДСТУ, а спосіб їх використання не повинен суперечити вимогам з техніки безпеки та експлуатації. Налаштування параметрів мережі

					КС КРБ 123.181.00.00 ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

здійснюється на рівні операційних систем та програмного забезпечення активного комутаційного обладнання.

Діагностика комп'ютеризованої системи обліку та контролю навчальних занять відбувається згідно регламенту та відповідного розкладу, і передбачає перевірку параметрів працездатності компонентів локальної комп'ютерної мережі та прикладного програмного забезпечення.

Перспективами розвитку комп'ютеризованої системи обліку та контролю навчальних занять є масштабування комп'ютерної мережі шляхом додавання більшої кількості робочих станцій та інтеграції програмного забезпечення з іншими системами підтримки та автоматизації навчального процесу у закладах освіти.

Модернізація системи обліку та контролю навчальних занять можлива у випадку міграції desktop програмного забезпечення на web-орієнтований програмний додаток, що передбачає повне відтворення схеми бази даних і самих даних, або безпосередньо використання того ж самого СКБД. При модернізації повинна бути збережена логіка виконання операцій керування процесами обліку та контролю.

Надійність комп'ютеризованої системи контролю та обліку навчальних занять в контексті захищеності повинна забезпечуватись засобами і механізмами на рівні операційних систем сервера, робочих станцій та комутаційного обладнання комп'ютерної мережі. Доступ до бази даних повинен використовувати функціональність авторизаційних механізмів системи керування базами даних.

Система повинна бути захищена від фізичних чи механічних пошкоджень на рівні апаратного забезпечення, шляхом обмеження доступу до коробів (кабельних), комутаційних розеток, комутаторів, комутаційних шаф і т.д.

Основними вимогами і функціями комп'ютеризованої системи обліку та контролю навчальних занять є:

– забезпечення комунікаційної інфраструктури при взаємодії клієнтського програмного забезпечення з системою керування базами даних;

					КС КРБ 123.181.00.00 ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

- адекватна реакція на дії користувачів;
- можливість забезпечення функціональності, що відображає суть процесів обліку та контролю за проведеними заняттями;
- визначена продуктивність виконання операцій;
- контрольований та авторизований доступ до інформаційних ресурсів;
- зручність використання та обслуговування компонентів комп'ютеризованої системи;
- наявність засобів резервного формування копій бази даних.

Апаратне забезпечення комп'ютеризованої системи контролю та обліку навчальних занять включає наступні компоненти:

1. Активне комутаційне обладнання:

- D-Link DES-1016P – 4 шт.
- D-Link DES 1005 P – 2 шт.
- D-Link DGS-1210-10P/F – 1 шт.

2. Сервер баз даних:

– процесор з тактовою частотою не менше 2,1 ГГц з 8 ядрами та 16 потоками;

- розмір оперативної пам'яті – не менше 32 ГБ;
- об'єм жорсткого диску - не менше 8 Тб.

3. Робочі станції:

- процесор з тактовою частотою не менше 3,2 ГГц;
- об'єм оперативної пам'яті - не менше 8 ГБ;
- об'єм жорсткого диску – не менше 200 Гб.

Операційна система сервера бази даних повинна підтримувати систему керування базами даних MS SQL Server та забезпечувати зручність адміністрування. Можливе застосування як ОС сімейства Windows Server, так і Unix-подібних.

Операційна система робочих станцій рекомендована до використання – Windows 10.

					КС КРБ 123.181.00.00 ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

1.2 Аналіз типової організаційної структури закладів освіти

Провівши аналіз технічного завдання, необхідно більш детально розглянути організаційну структуру закладів освіти для визначення інформаційних потоків, а також основних видів інформації, яка буде циркулювати у локальній комп'ютерній мережі.

Типова організаційна структура шкіл має структуру, яка наведена на рис. 1.1.

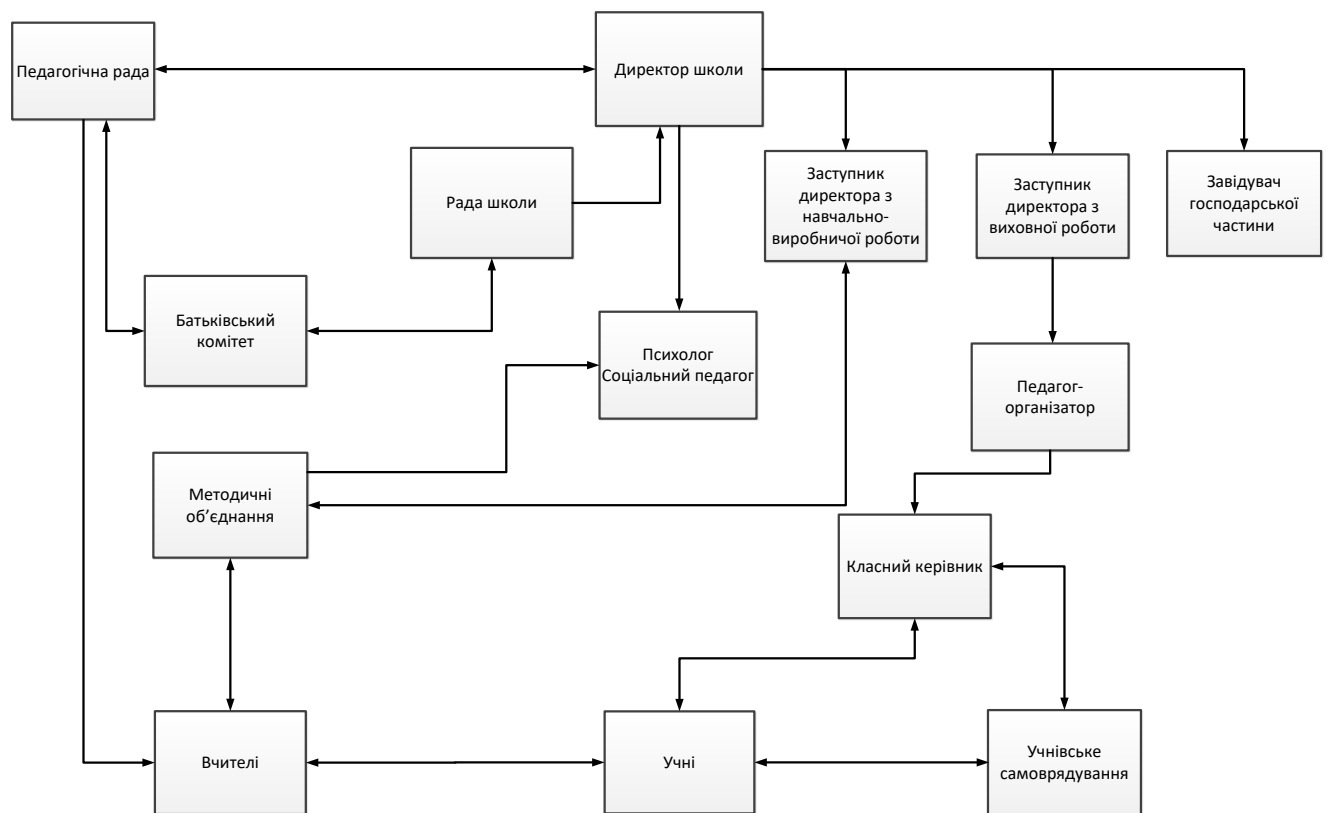


Рисунок 1.1 – Типова організаційна структура закладу освіти

Виходячи з організаційної структури закладу освіти коротко проведемо аналіз діяльності кожного з учасників навчального процесу та визначимо кількість наявних робочих станцій, серверів та іншого обладнання, яке входить до складу мережі.

До структури навчального закладу входять:

					КС КРБ 123.181.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

1 Директор. Для ефективної роботи директора, необхідно встановити один комп'ютер у його кабінеті та один комп'ютер у приймальні для секретаря. Для з'єднання з іншими підрозділами необхідно використати комутатор мінімум на 4 порти.

2 Учительська. Для підвищення ефективності роботи учителів, автоматизованої підготовки матеріалів уроків в учительській необхідно організувати хоча б чотири робочих місця. Відповідно для об'єднання цих комп'ютерів в єдиний інформаційний простір потрібно вибрати комутатор на 5 або 8 портів.

3 Мультимедійні класи. У мультимедійних класах знаходиться по одному комп'ютеру, які з'єднано із рідкокристалічними телевізорами. Доцільності встановлювати у них комутатори немає, оскільки поряд знаходяться комп'ютерні класи з більшою кількістю ПК, де наявність комутатора є обов'язковою.

4 Комп'ютерні класи. Для забезпечення ефективного навчання учнів та підвищення кваліфікації викладачів, у закладі освіти необхідно створити ще хоча б 2 класи (наявних є 2 класи). Два класи знаходяться на першому поверсі, а два інших на другому. У всіх класах планується встановити по 9 комп'ютерів згідно вимог охорони праці користувачів ПК (на одне робоче місце 6 м²). Кількість комутаторів та портів буде визначатись фізичною топологією мережі.

Для розробки проекту комп'ютеризованої системи контролю та обліку навчальних занять потрібно необхідно провести аналіз плану приміщення, визначити фізичні вимоги до розташування обладнання та проаналізувати вимоги щодо типу організації і прокладання мережі. Окрім цього, потрібно визначити вимоги до прикладного програмного забезпечення, розробити алгоритми його функціонування для забезпечення ефективності процесу керування моніторингом та обліком навчальних занять.

					КС КРБ 123.181.00.00 ПЗ	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2 ПРОЕКТУВАННЯ МЕРЕЖЕВОЇ ІНФРАСТРУКТУРИ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРИЗОВАНОЇ СИСТЕМИ

2.1 Проектування та налаштування мережевої інфраструктури комп'ютеризованої системи контролю та обліку навчальних занять

Загальна структура мережевої інфраструктури комп'ютеризованої системи контролю та обліку навчальних занять на логічному рівні повинна відповідати організаційній структурі підприємства, в даному випадку організаційній структурі закладу освіти. Кожен із структурних підрозділів, у якому наявні робочі станції є сегментами мережі, які необхідно об'єднати в єдиний інформаційний простір за допомогою комутуючого обладнання. У табл. 2.1 наведено розподіл кількості робочих станцій відносно приміщення з автоматизованими робочими місцями.

Таблиця 2.1 – Структура комп'ютерної мережі закладу освіти

№ приміщення	Кількість комп'ютерів	Призначення
4	1	Мультимедійний клас (учительський комп'ютер- 1)
5	9	Комп'ютерний клас (Учнівські комп'ютери – 8, учительський комп'ютер- 1)
6	9	Комп'ютерний клас (Учнівські комп'ютери – 8, учительський комп'ютер- 1)

					КС КРБ 123.181.00.00 ПЗ		
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>			
<i>Розроб.</i>		Українець У.В.			<i>Лім.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Перевір.</i>		Тиш Є.В.				20	
<i>Реценз.</i>					ТНТУ, каф. КС, гр. СІс-44		
<i>Н. Контр.</i>		Луцик Н.С.					
<i>Затверд.</i>		Осухівська Г.М.					
					<i>Проектування мережевої інфраструктури та програмного забезпечення комп'ютеризованої системи</i>		

№ приміщення	Кількість комп'ютерів	Призначення
7	1	Мультимедійний клас (учительський комп'ютер- 1)
10	1	Секретар директора
11	1	Директор школи
12	1	Мультимедійний клас (учительський комп'ютер- 1)
13	9	Комп'ютерний клас (Учнівські комп'ютери – 8, учительський комп'ютер- 1)
14	9	Комп'ютерний клас (Учнівські комп'ютери – 8, учительський комп'ютер- 1)
15	4	Учительська
16	1	Мультимедійний клас (учительський комп'ютер- 1)
17	1	FTP сервер

Для забезпечення швидкості передачі даних на рівні 100 Мб/с необхідно вибрати технологію передачі даних. Оскільки, безпроводна передача даних у закладах освіти, особливо у школах, не рекомендується, тому потрібно здійснювати вибір лише технологій передачі даних на основі кабельних систем. Доцільним є використання технології Fast Ethernet на основі виті пари категорії 5е.

Комутація каналів необхідна для під'єднання робочих станцій до комутаторів через певні інтерфейси, що дозволило б реалізувати обмін інформаційними ресурсами між користувачами різних груп та різних рівнів доступу. Оскільки, у більшості закладів освіти розподілу на підмережі та VLAN

					КС КРБ 123.181.00.00 ПЗ	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

немає (не є доцільним), наведемо загальне розбиття за портами кожного з комутаторів. У таблиці 2.2 наведено структуру мережі закладу освіти.

Таблиця 2.2 – Структура комп'ютерної мережі

Комутаційний пристрій	Інтерфейс	Розетка/інтерфейс
S_1	FE0/1	ws_1
	FE0/2	ws_2
	FE0/3	S_5
S_2	FE0/1	S_5
	FE0/2	ws_3
	FE0/3	ws_4
	FE0/4	ws_5
	FE0/5	ws_6
	FE0/6	ws_7
	FE0/7	ws_8
	FE0/8	ws_9
	FE0/9	ws_10
	FE0/10	ws_11
	FE0/11	ws_12
S_3	FE0/1	S_5
	FE0/2	ws_13
	FE0/3	ws_14
	FE0/4	ws_15
	FE0/5	ws_16
	FE0/6	ws_17
	FE0/7	ws_18
	FE0/8	ws_19
	FE0/9	ws_20
	FE0/10	ws_21

Комутаційний пристрій	Інтерфейс	Розетка/інтерфейс
S_4	FE0/1	S_5
	FE0/2	ws_22
	FE0/3	ws_23
	FE0/4	ws_24
	FE0/5	ws_25
	FE0/6	ws_26
S_5	FE0/1	S_1
	FE0/2	S_2
	FE0/3	S_3
	FE0/4	S_4
	FE0/5	S_6
	FE0/6	S_7
	FE0/7	Server
	FE0/8	Internet
S_6	FE0/1	S_5
	FE0/2	ws_27
	FE0/3	ws_28
	FE0/4	ws_29
	FE0/5	ws_30
	FE0/6	ws_31
	FE0/7	ws_32
	FE0/8	ws_33
	FE0/9	ws_34
	FE0/10	ws_35
	FE0/11	ws_36

Комутаційний пристрій	Інтерфейс	Розетка/інтерфейс
S_7	FE0/1	S_5
	FE0/2	ws_37
	FE0/3	ws_38
	FE0/4	ws_39
	FE0/5	ws_40
	FE0/6	ws_41
	FE0/7	ws_42
	FE0/8	ws_43
	FE0/9	ws_44
	FE0/10	ws_45
	FE0/11	ws_46

Практично усі комутатори, які наявні у локальній комп'ютерній мережі є некерованими, окрім центрального комутатора S_5. Комутатор S_5 є комутатором другого рівня і налаштування основних його параметрів наведено далі у цьому розділі. У табл. 2.3 наведено логічну IP-адресацію робочих станцій та сервера, які наявні у закладі освіти

Таблиця 2.3 – IP-адресація робочих станцій і сервера

Користувачі	№ приміщення	Діапазон назв компонентів мережі	Діапазон IP- адрес	Маска мережі
Учні	5	ws_28	192.168.1.28	255.255.255.0
		
	ws_35	192.168.1.36		
	6	ws_37	192.168.1.37	
.....			
		ws_45	192.168.1.45	255.255.255.0

Користувачі	№ приміщення	Діапазон назв компонентів комп'ютерної мережі	Діапазон IP- адрес	Маска мережі
Учні	13	ws_4	192.168.1.4	255.255.255.0
		
	ws_11	192.168.1.11		
	14	ws_13	192.168.1.13	
.....			
Учителі	4	ws_27	192.168.1.27	255.255.255.0
		5	ws_36	192.168.1.36
	6	ws_45	192.168.1.45	255.255.255.0
	7	ws_46	192.168.1.46	255.255.255.0
	10	ws_1	192.168.1.252	255.255.255.0
	11	ws_2	192.168.1.2	255.255.255.0
	12	ws_3	192.168.1.3	255.255.255.0
	13	ws_12	192.168.1.12	255.255.255.0
	14	ws_21	192.168.1.21	255.255.255.0
	15	ws_22	192.168.1.22	255.255.255.0
		
	16	ws_25	192.168.1.25	255.255.255.0
		ws_26	192.168.1.26	
	17	FTP сервер	192.168.1.253	255.255.255.0
	17	Router	192.168.1.1	255.255.255.0
Dynamic IP adress				

Фізична топологія комп'ютерної мережі при проектуванні комп'ютеризованої системи контролю та обліку навчальних занять показана у додатку Б.

Визначивши схему розташування апаратних пристроїв комп'ютерної мережі та IP-адресну схему потрібно реалізувати схему з'єднання активного комутаційного обладнання (рис. 2.1) і логічну топологію мережі (рис. 2.2).

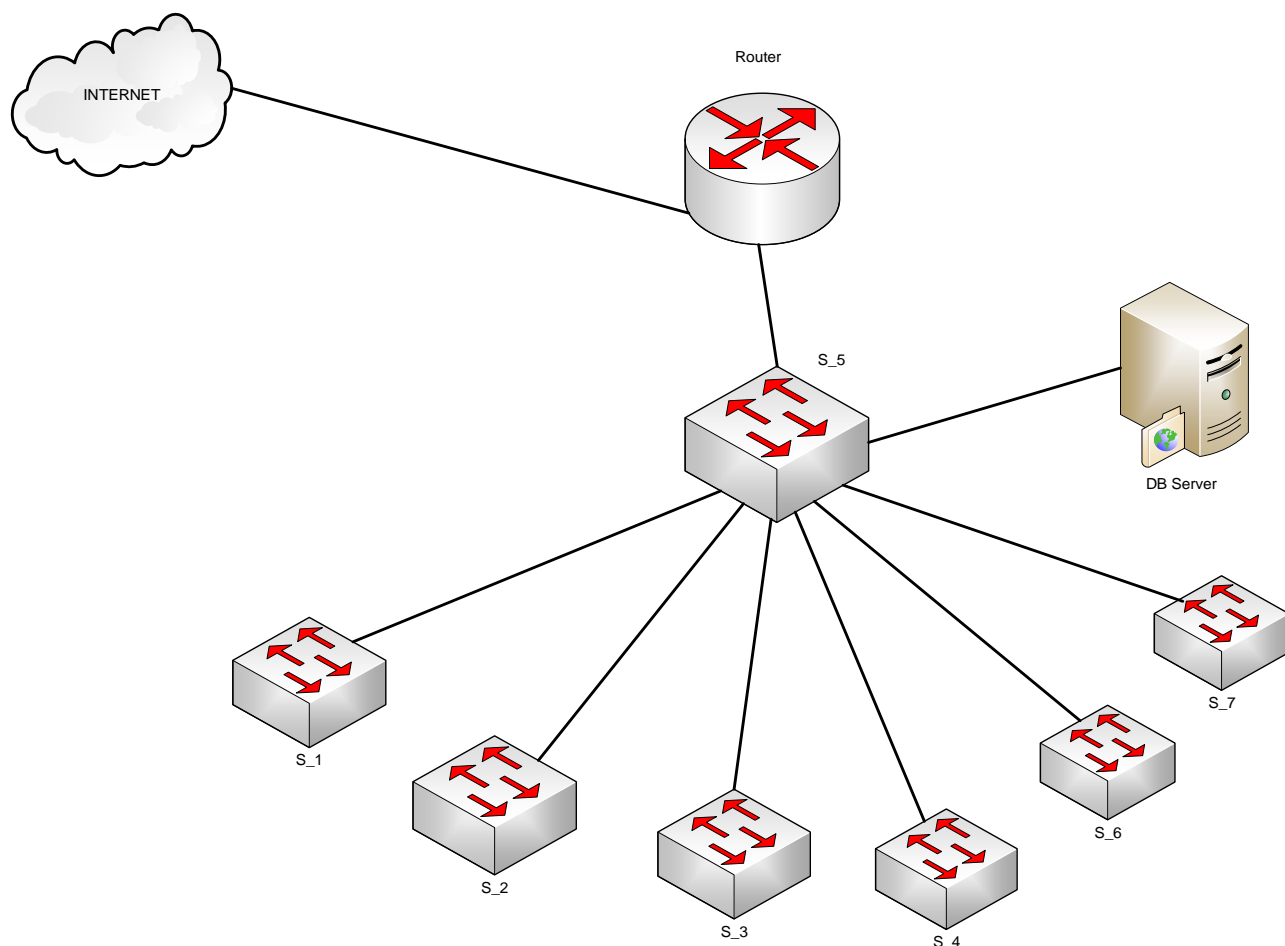


Рисунок 2.1 – Логічна топологія мережевої інфраструктури

В якості активного комутаційного обладнання обрано комутатори фірми D-Link, що характеризуються високою надійністю і задовольняють вимоги, наведені у технічному завданні. Практично усі комутатори є некерованими, окрім одного маршрутизатора 2 рівня. Характеристики обладнання та їх зовнішній вигляд приведено у п.п. 2.5.

Змн.	Арк.	№ докум.	Підпис	Дата

КС КРБ 123.181.00.00 ПЗ

Арк.

26

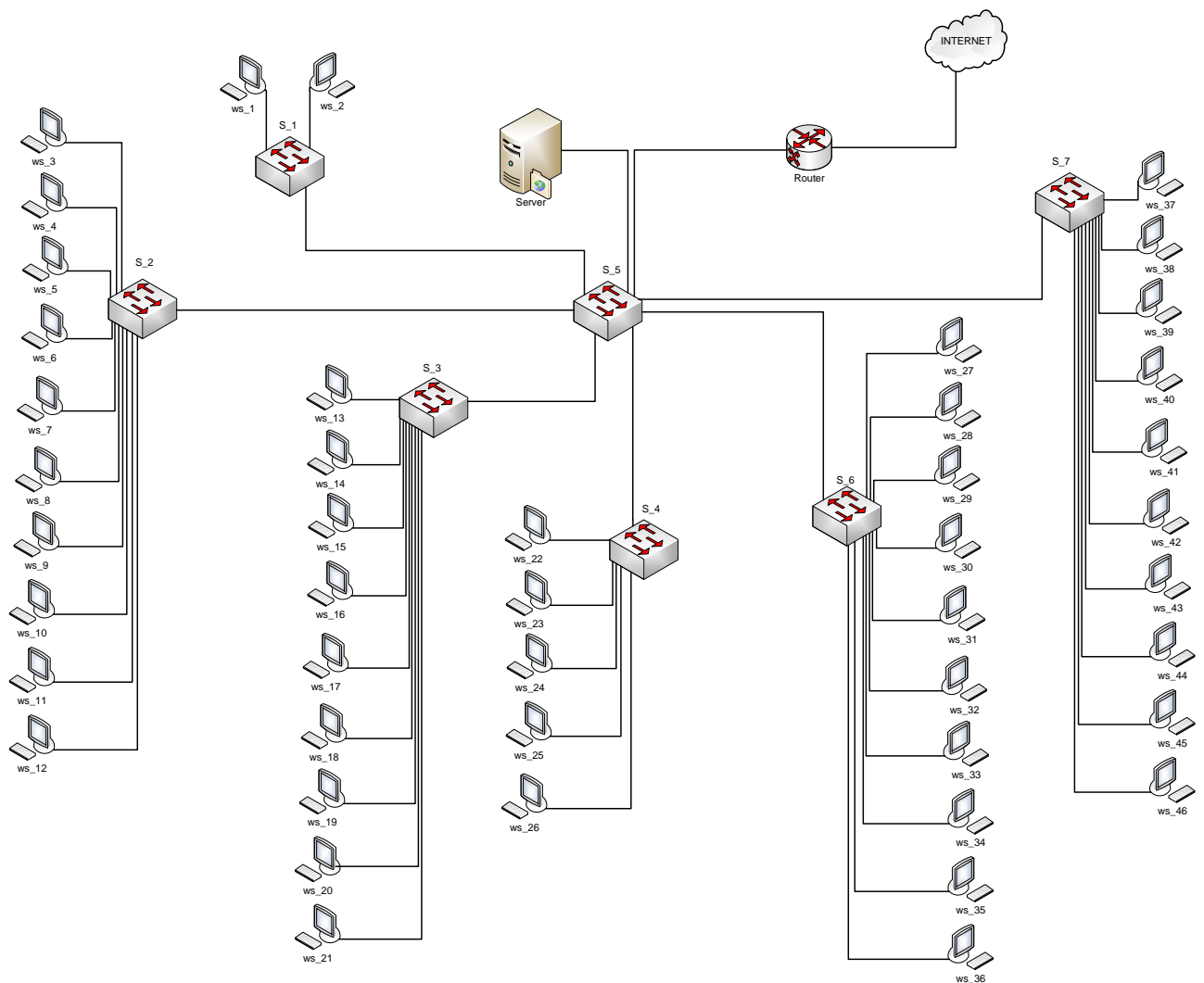


Рисунок 2.2 – Логічна топологія мережевої інфраструктури

Для ефективної роботи активного комутаційного обладнання, а саме 5-ти та 8-ми портових комутаторів фірми D-Link необхідно провести їх налаштування. Загальний алгоритм налаштування наступний:

1) створити обліковий запис адміністратора та включити шифрування паролів в cfg - файлі наступними командами:

```
admin#create account admin <user_name>
admin#enable password encryption
```

2) створити назву комутатора, для прикладу switch_1:

```
admin# config command_prompt switch_1
```

3) присвоїти IP-адресу комутатору:

```
switch_1:admin#config ipif System ipaddress 192.168.1.249/24
```

4) задати часовий пояс:

```
switch_1:admin#config time_zone operator + hour 3 min 0
```

Змн.	Арк.	№ докум.	Підпис	Дата

КС КРБ 123.181.00.00 ПЗ

Арк.

27

5) задати час, в мережі є доступний NTP сервер:

```
switch_1:admin#enable sntp switch_1:admin#config sntp primary  
192.168.1.3 secondary 192.168.1.4 poll-interval 3600
```

де 192.168.1.3, 192.168.1.4 - це IP-адреса NTP сервера

6) Виключити web-інтерфейс:

```
admin#disable web
```

7) включити ssh:

```
switch_1:admin#enable ssh
```

8) виключити telnet:

```
switch_1:admin#disable telnet
```

9) задати gateway по замовчуванню:

```
switch_1:admin#create iproute default 192.168.1.1
```

10) налаштувати доступ до комутатора тільки з визначени IP-адрес:

```
switch_1:admin#create trusted_host 192.168.1.4
```

11) включити захист від кільцювання:

```
switch_1:admin#enable
```

```
loopdetect switch_1:admin#config
```

```
loopdetect ports 1-8 state enabled switch_1:admin#config
```

```
loopdetect recover_timer 60 interval 10 mode port-based
```

12) зберегти базові налаштування:

```
switch_1:admin#save
```

Наведені вище фрагменти налаштування комутаторів D-Link є базовими, повний набір команд конкретного комутатора наведено в інструкції з експлуатації і визначається потребами користувача.

2.2 Виявлення сутностей при проектуванні системи контролю та обліку навчальних занять

На основі детального аналізу діяльності викладача було виділено сутності предметної області, які важливі в контексті контролю та обліку навчальних занять, виявлено їх атрибути та з'ясовано зв'язки між ними. Визначені сутності,

					КС КРБ 123.181.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		28

їх атрибути та характеристики цих атрибутів представлено в табличному вигляді (табл. 2.4 – 2.10).

Таблиця 2.4 – Сутність «Дисципліна» та її атрибути

Назва атрибута	Тип атрибута	Примітка
#ID_Дисципліна	Цілочисельний	Первісний ключ
Назва	Стрічковий	

Таблиця 2.5 – Сутність «Група» та її атрибути

Назва атрибута	Тип атрибута	Примітка
#ID_Група	Цілочисельний	Первісний ключ
Назва групи	Цілочисельний	

Таблиця 2.6 – Сутність «Студент» та її атрибути

Назва атрибута	Тип атрибута	Примітка
#ID_Студент	Цілочисельний	Первісний ключ
#ID_Група	Цілочисельний	Зовнішній ключ
ПІБ	Цілочисельний	

Таблиця 2.7 – Сутність «Викладач» та її атрибути

Назва атрибута	Тип атрибута	Примітка
#ID_Викладач	Цілочисельний	Первісний ключ
ПІБ	Цілочисельний	

Таблиця 2.8 – Сутність предметної області «Вид заняття» та її атрибути

Назва атрибута	Тип атрибута	Примітка
#ID_Вид_заняття	Цілочисельний	Первісний ключ
Назва	Цілочисельний	

Таблиця 2.9 – Сутність предметної області «Заняття» та її атрибути

Назва атрибута	Тип атрибута	Примітка
#ID_Заняття	Цілочисельний	Первісний ключ
#ID_Викладач	Цілочисельний	Зовнішній ключ
#ID_Дисципліна	Цілочисельний	Зовнішній ключ
#ID_Вид_заняття	Цілочисельний	Зовнішній ключ
#ID_Група	Цілочисельний	Зовнішній ключ
Дата	Дата	
Тема	Стрічковий	

Таблиця 2.10 – Сутність предметної області «Бали» та її атрибути

Назва атрибута	Тип атрибута	Примітка
#ID_Бал	Цілочисельний	Первісний ключ
#ID_Студент	Цілочисельний	Зовнішній ключ
Бал	Цілочисельний	

Між наведеними сутностями предметної області існують відповідні зв'язки, що повинні підтримуватись через зовнішні ключі в фізичній реалізації схеми бази даних. Так наприклад сутність «Бал» пов'язана із сутністю «Студент» через однойменну властивість. Підтримка цього зв'язку в фізичній реалізації бази даних повинна включати первинний ключ відношення «Студент» та зовнішній ключ для таблиці «Бал». Отримані дані є вихідною інформацією для проектування та розробки структури бази даних.

2.3 Проектування схеми бази даних комп'ютеризованої системи обліку та контролю навчальних занять

Проаналізувавши сутності предметної області та зв'язки між ними було отримано реляційні відношення та сформовано на їх основі схему бази даних, структуру якої подано у таблицях та зображено на ER-діаграмі.

У результаті аналізу предметної області, з'ясування її сутностей та зв'язків між ними здійснено фізичну реалізацію бази даних. Сутності предметної області було відображено на таблиці бази даних, атрибути – на стовпці таблиць, а зв'язки між ними – на відповідні зовнішні ключі.

Сутності групи, студентів, викладачів, дисциплін та видів занять спроектовані як словники із виділеним унікальним ідентифікатором для первинного ключа та текстовим полем для атрибутів (наприклад прізвище та ініціали студента). Крім того реляція «Студент» та «Група» пов'язані між собою через зовнішній ключ «один-до-багатьох». На рис. 2.3 – 2.7 показано таблиці, які є довідниками, або по-іншому словниками базиданих


Groups	
	id_group
	name

Рисунок 2.3 – Структура таблиці «Groups»


Students	
	id_students
	id_group
	FI

Рисунок 2.4– Структура таблиці «Students»


Teachers	
	id_teacher
	FIO

Рисунок 2.5 – Структура таблиці «Teachers»


Disciplines	
	id_discipline
	discipline

Рисунок 2.6 – Структура таблиці «Disciplines»

Vid_lessons	
	id_vid_lesson
	vid_lesson

Рисунок 2.7 – Структура таблиці «Vid_lessons»

Сутність «Заняття» було відображено на таблицю Lessons (рис. 2.8), що включає набір ключів які зв'язують її із іншими таблицями – Teachers, Disciplines, Vid_lessons, Groups. Зовнішні ключі таблиці зв'язані із протилежними первісними ключами таблиць зв'язком «один-до-багатьох». Крім того таблиця містить поле date типу datetime (дата та час) і тема стрічкового типу. Використання зовнішніх ключів допомагає нормалізувати структуру збереження та розподіленості даних, підвищує масштабованість бази даних.


Lessons	
	id_lesson
	id_teacher
	id_discipline
	id_vid_lesson
	id_group
	date
	tema

Рисунок 2.8– Структура таблиці «Lessons»

Сутність «Бали» було розподілено на дві таблиці «Bally» (рис. 2.9) та «SuperBalls» (рис. 2.10): в одну з яких виділено суперсутність, що підтримує зв'язок оцінок із заняттями через зовнішній ключ.


Bally	
	id_bally
	id_students
	ball
	id_super

Рисунок 2.9 – Структура таблиці «Bally»


SuperBalls	
	Id
	IdLesson

Рисунок 2.10 – Структура таблиці «SuperBalls»

Для підтримки масштабованості та нормалізації структури даних в базі даних використовуються зв'язані таблиці TeacherDiscipline (рис. 2.11), яка підтримує зв'язок між викладачами та предметами, які вони викладаються і таблиця GroupDisTeach (рис. 2.12), що відповідає за зв'язок між парами «Викладач-Предмет» та групою для якої читається даний предмет.


TeacherDiscipline	
	Id
	Id_Teacher
	Id_Discipline

Рисунок 2.11 – Структура таблиці «TeacherDiscipline»


GroupDisTeach	
	Id
	IdGroup
	IdTeachDis

Рисунок 2.12 – Структура таблиці «GroupDisTeach»

Розроблена структура бази даних орієнтована на відповідність стандарту SQL, нормалізації даних, підтримки масштабованості та забезпечення коректного представлення предметної області.

На основі спроектованої бази даних одержано її структуру у вигляді ER-діаграми, як показано на рис. 2.13.

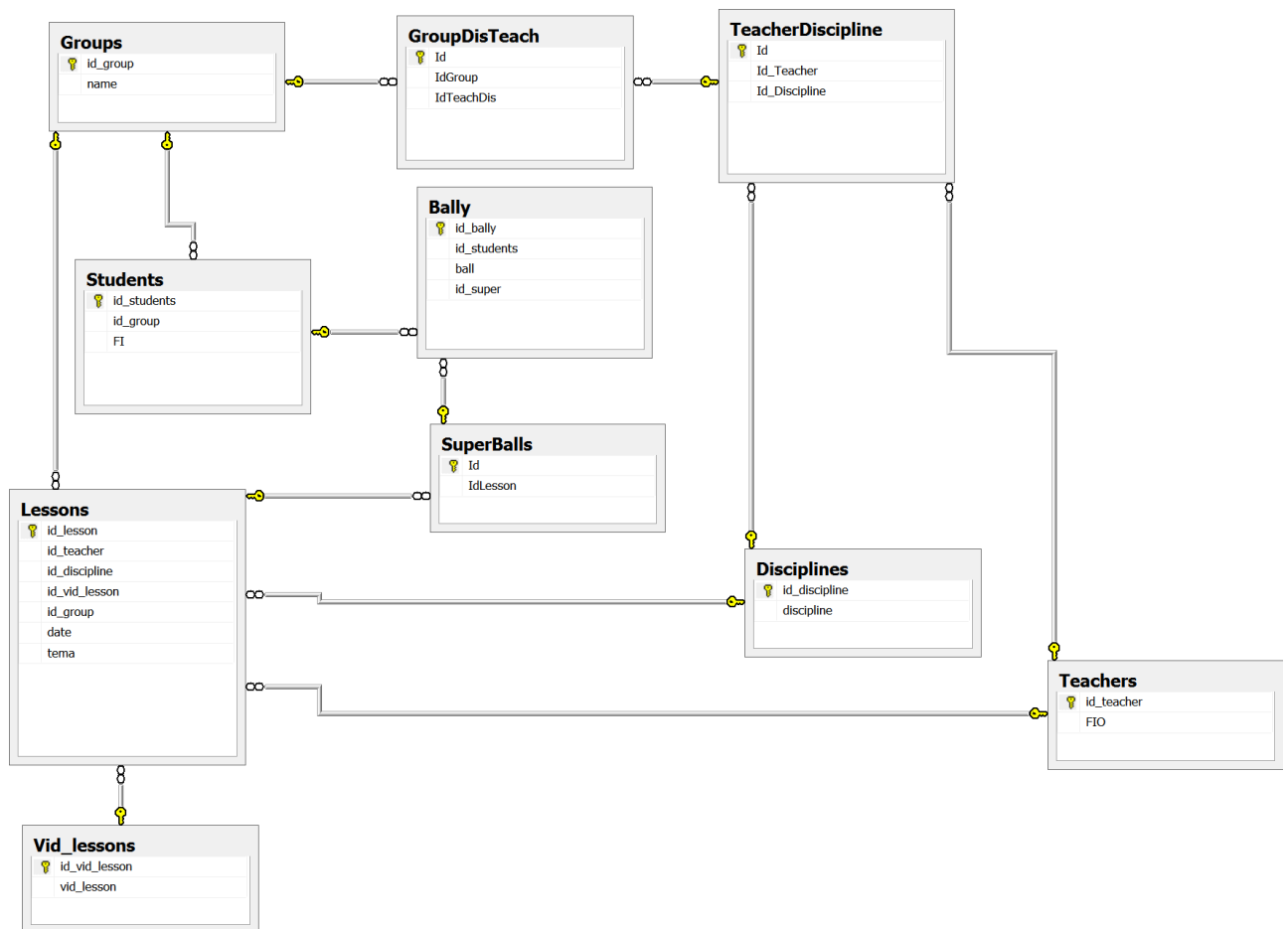


Рисунок 2.13 – ER-діаграма бази даних

База даних є нормалізованою, що забезпечує відсутність аномалій додавання, видалення та оновлення інформації про заняття, їх контроль та облік.

2.4 Розробка алгоритму роботи функціонування комп'ютеризованої системи контролю та обліку навчальних занять

Згідно вимог технічного завдання та на основі аналізу прикладних задач предметної області було з'ясовано вимоги до функціональності, що повинен забезпечувати прикладний додаток контролю та обліку навчальних занять, а саме:

- внесення списку студентів;
- внесення списку викладачів;
- внесення списків дисциплін та видів занять;
- редагування зв'язків між даними в довідниках;
- журнал для внесення та редагування даних про заняття;

- механізми виставлення відміток про успішність студентів на заняттях;
- планувальник занять із внесенням розкладу занять і темами, що розглядаються;

На основі аналізу розроблених вимог до функціональних можливостей програми розроблено модульну структуру програмного засобу (рис. 2.14), що забезпечуватиме виконання прикладних задач та вимог предметної області.

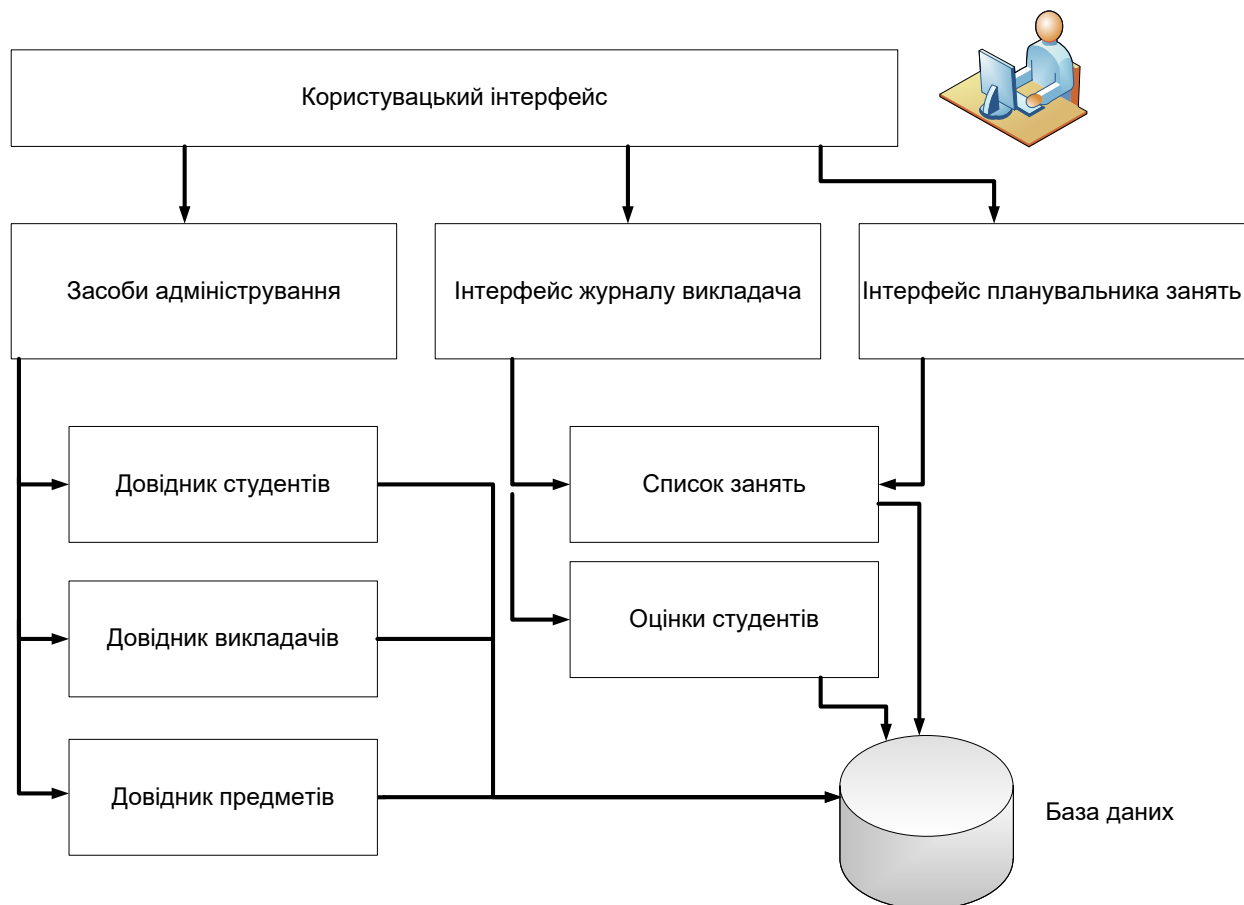


Рисунок 2.14 – Структура модулів програмної реалізації комп'ютеризованої системи контролю та обліку навчальних занять

Для реалізації визначених функціональних вимог та виконання прикладних задач, які повинні підтримуватись системою контролю та обліку навчальних занять розроблено алгоритм роботи програмного додатку.

Алгоритм роботи спроектовано із врахуванням мультизадачності користувацького віконного інтерфейсу, можливості виконання нелінійних послідовностей операцій та зв'язаності модулів програми.

Робота програмного засобу орієнтована на централізовану взаємодію із репозиторієм даних на основі СКБД, а алгоритм її роботи показаний на рис. 2.15.

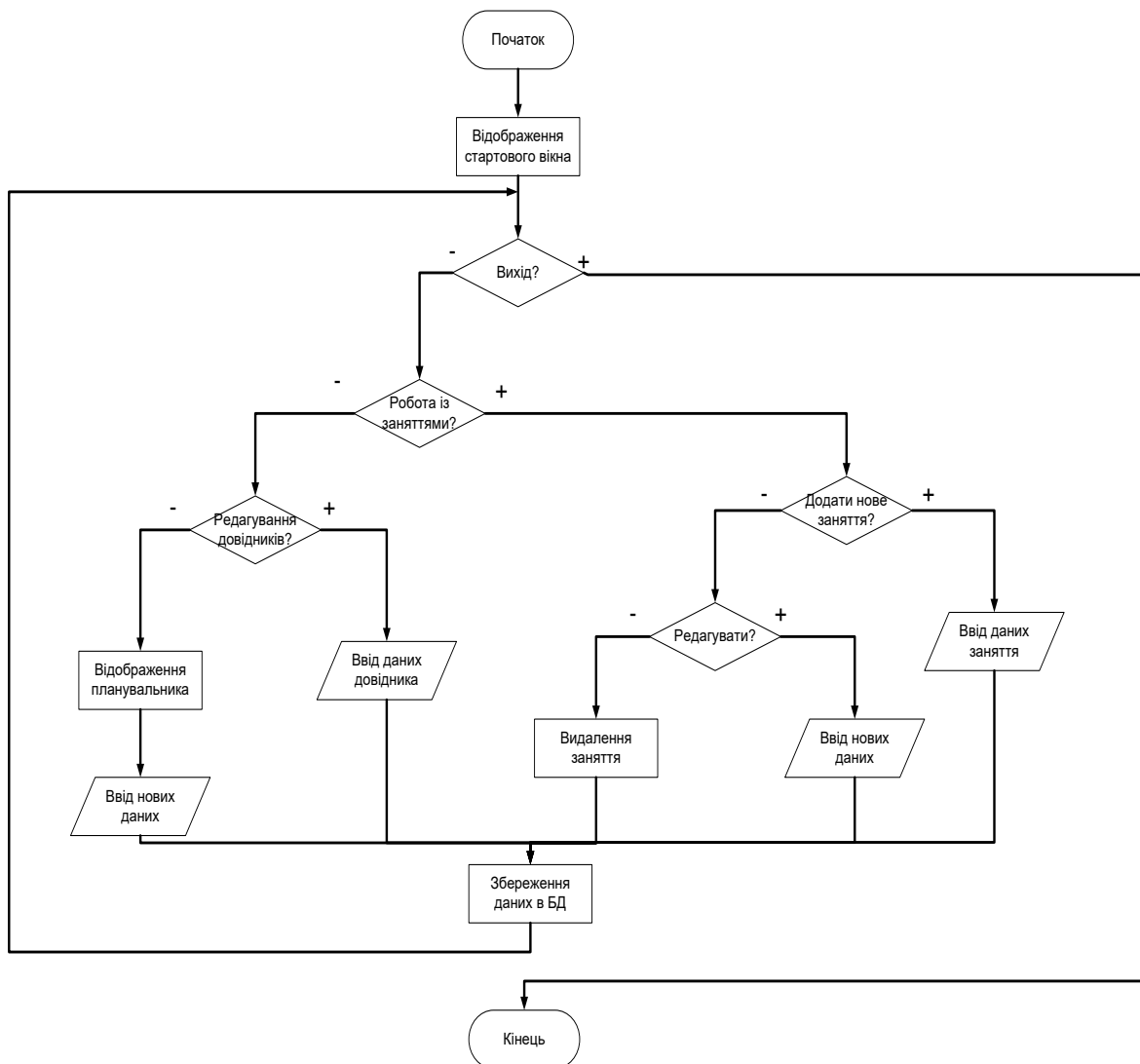


Рисунок 2.15– Алгоритм роботи програмної реалізації електронного журналу викладача

Розробка модульної архітектури та алгоритму роботи дозволяє розбити процеси та прикладні задачі на блоки операцій, які реалізуються з допомогою програмних конструкцій (класів, структур, методів). Виконання цього етапу також дозволяє з'ясувати зв'язки між модулями та компонентами системи, розподілити функціональність між різними рівнями – користувацьким інтерфейсом, програмним кодом та сховищем даних.

2.5 Обґрунтування апаратного забезпечення комп'ютеризованої системи контролю та обліку навчальних занять

Мережеву інфраструктуру комп'ютеризованої системи контролю та обліку навчальних занять запропоновано реалізувати за допомогою комутаторів фірми D-Link. Провівши аналіз технічних характеристик і вартості комутаційного обладнання обрано наступне обладнання:

- два некерованих комутатори D-Link 1005 P (рис. 2.16);
- чотири некерованих комутатори DES-1016P з 16 портами 10/100Base-TX1 (рис. 2.17);
- один WebSmart комутатор 2 рівня з 8 портами 10/100Base-T + 2 комбо-портами 10/100/1000 BASE-T/SFP (рис. 2.18).
- один комутатор з 8 портами DGS-1008P.



Рисунок 2.16 – Комутатор D-Link 1005 P



Рисунок 2.17 – Комутатор DES-1016P з 16 портами 10/100Base-TX1



Рисунок 2.18 – WebSmart комутатор 2 рівня

При виборі сервера баз даних для зберігання та маніпулювання даними при контролі та обліку навчальних занять необхідно врахувати наступні аспекти:

- тип жорстких дисків;
- тип процесора;
- тип оперативної пам'яті.

Рекомендується використовувати жорсткі диски з максимальною швидкістю доступу. Чим вище вимоги до продуктивності сервера, тим більше операцій вводу-виводу він виконує в одиницю часу (стандартна одиниця - IOPS, або кількість таких операцій за 1 секунду).

У даний час ще використовуються застарілі механічні диски типу SATA і SAS, в яких швидкість роботи безпосередньо визначається частотою обертання шпинделя. Якщо в першому випадку вона становить 7 200 об/хв., то у другому – вже 10 000 - 15 000 об/хв.

Однак найшвидшими є SSD-диски. Якщо взяти високопродуктивний SAS-диск і порівняти його з SSD, то при однаковому обсязі і вартості в першому випадку одержуємо близько 150 IOPS, а в другому - у десятки разів більше. Відповідно, використовувати SSD-диски раціональніше для отримання максимальної продуктивності.

Роботу сучасного сервера баз даних неможливо уявити без RAID-масиву - технології віртуалізації даних, яка має на меті об'єднання декількох фізичних дисків в один віртуальний. Оптимальний вибір для СКБД - дзеркальний дисковий масив RAID 10. До його переваг належать:

- прискорення роботи сервера (за рахунок збільшення IOPS), так як всі дані будуть записуватися паралельно на кожен фізичний носій;
- підвищення відмовостійкості – дані будуть продубльовані та інформація збережеться у разі виходу з ладу одного фізичного носія.

Недолік у такого рішення полягає в тому, що сумарний обсяг віртуального диска в RAID-масиві буде дорівнює обсягу одного фізичного накопичувача. Наприклад, якщо використовувалися два SSD-диска по 500 Гб, обсяг масиву RAID 10 також складе 500 Гб.

Ще один популярний варіант об'єднання дисків - технологія RAID 0. У цьому випадку обсяги фізичних дисків підсумовуються, але при пошкодженні будь-якого з них частина даних буде втрачена, так як дублювання не виконувалося. Головний плюс такого рішення – збільшення загального обсягу віртуального простору. При об'єднанні двох SSD-дисків по 500 Гб розмір масиву RAID 0 складає 1000 Гб. Однак для реалізації комп'ютеризованої системи обліку та контролю навчальних занять рекомендується обрати масив RAID 10 як оптимальному по швидкості роботи і надійності зберігання даних.

Відносно оперативної пам'яті, то тут діє принцип – чим більше, тим краще. Більше RAM означає, що швидше буде опрацьовуватись інформація. При зверненні до БД, збереженої на сервері, дані будуть кешуватися в оперативній пам'яті. Найбільша продуктивність досягається в тому випадку, коли її обсяги і дискового простору однакові.

Параметри процесора вибираються індивідуально в залежності від розміру БД і з урахуванням її потенційного збільшення. Основним параметром є кількість ядер, яке може досягати декількох десятків. Серед інших критеріїв можна додатково враховувати тактову частоту шини, технологію виготовлення, бренд виробника.

При реалізації комп'ютеризованої системи контролю та обліку навчальних занять запропоновано конфігурацію сервера HP ProLiant DL560 Gen8 (5 SFF), який показано на рис. 2.19.

					КС КРБ 123.181.00.00 ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 2.19 – Сервер баз даних HP ProLiant DL560 Gen8 (5 SFF)

Основним технічними характеристиками сервера, представленого на рис. 2.19 є:

- процесор – 4*E5-4650, 2.70-3.30 ГГц, Eight Core, 20MB, 130W, 1600;
- RAM: 32 ГБ DDR3 (4 x 8 GB 1866MHz);
- RAID-контролер: HP Smart Array P420i (Advanced Pack) + FBWC 1 GB + конденсатор;
- мережевий модуль змінний: 4 * 1GBE;
- блок живлення: 2 * 1200 W.

Таким чином, обґрунтовано технічні характеристики апаратного забезпечення комп'ютеризованої системи контролю та обліку навчальних занять.

					КС КРБ 123.181.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		41

**РОЗДІЛ 3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ
КОМП'ЮТЕРИЗОВАНОЇ СИСТЕМИ КОНТРОЛЮ ТА ОБЛІКУ
НАВЧАЛЬНИХ ЗАНЯТЬ**

3.1 Реалізація засобів доступу до даних

Специфіка програмних систем, що орієнтовані на роботу із централізованих сховищем даних полягає у необхідності проектування та розробки рівня доступу до даних. Рівень доступу до даних формується із компонентів для доступу до СКБД, засобів маніпуляції даними всередині бази даних та технології відображення реляційних даних та структур на об'єктно-орієнтовані сутності.

Процеси з'єднання із базами даних та утримання підключення є надзвичайно ресурсоемкими, а специфічні маніпуляції з даними на стороні сервера – малоефективними в зв'язку із обмеженістю декларативної мови SQL. Для скорочення витрат ресурсів, підвищення ефективності обробки даних, вирішення задач специфічної обробки та представлення технологія ADO.NET пропонує концепцію «відключеного режиму».

Принципи цієї концепції полягають в тому, що дані із джерела вибираються при короткочасному і швидкому підключенні, формують відповідні структури та заповнюють специфічну конструкцію в клієнтській програмі.

Конструкція, що заповнюється відображає необхідну мінімальну структуру об'єктів та взаємозв'язків всередині бази даних, таких як таблиці, первинні та зовнішні ключі, обмеження, збережені процедури. Далі з'єднання із джерелом даних припиняється та всі маніпуляції проводяться в автономному режимі.

					КС КРБ 123.181.00.00 ПЗ			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		<i>Українець У.В.</i>			<i>Розробка програмного забезпечення комп'ютеризованої системи контролю та обліку навчальних занять</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Аркуші</i>
<i>Перевір.</i>		<i>Тиш Є.В.</i>					42	
<i>Реценз.</i>						<i>ТНТУ, каф. КС, гр. СІс-44</i>		
<i>Н. Контр.</i>		<i>Луцик Н.С.</i>						
<i>Затверд.</i>		<i>Осухівська Г.М.</i>						

Коли виникає необхідність оновлення даних на стороні клієнта чи їх збереження в базу даних відбувається підключення до джерела даних і синхронізація записів.

Автономна модель роботи із базою даних базується на використанні наборів спеціалізованих класів DataSet, та їх підмножин DataTable, DataRow, DataColumn. Ці класи спроектовані таким чином щоб надавати можливість формування представлення реляційної схеми практично будь-якого сховища даних, що підтримує реляційну модель та може бути підключене через провайдера даних.

При розробці комп'ютеризованої системи контролю та обліку навчальних занять для доступу до даних реалізовано бізнес-клас, що забезпечує зв'язну ланку між репозиторієм даних та користувацьким інтерфейсом. Цей бізнес-клас названо DataLayer і показано на рис. 3.1.

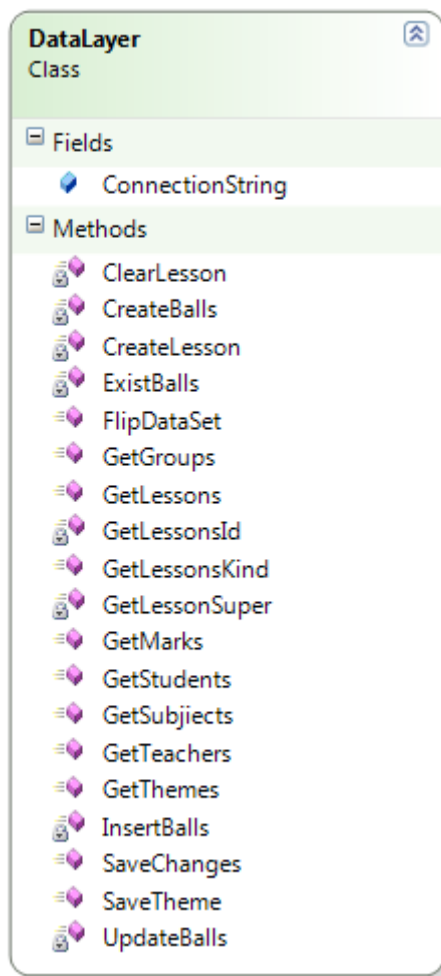


Рисунок 3.1 – Клас DataLayer та його методи

Основним призначенням класу `DataLayer` є вибірка даних із репозиторіїв та їх відображення в елементах візуального інтерфейсу, приймання оновлених даних та збереження у базу даних.

Наприклад, метод `GetTeachers` призначений для вибірки записів про викладачів із бази даних та формування словника «ключ-значення» для ідентифікаторів та прізвищ викладачів, як показано у лістингу 3.1.

Лістинг 3.1 – реалізація методу `GetTeachers ()`

```
public static Dictionary<int,string> GetTeachers()  
{  
    SqlConnection conn=new  
    SqlConnection(ConnectionString);  
    Dictionary<int, string> teachers = new  
    Dictionary<int, string>();  
    using (conn)  
    {  
        try  
        {  
            SqlCommand comm = new SqlCommand("SELECT *  
FROM Teachers", conn);  
            comm.CommandType = CommandType.Text;  
            conn.Open();  
            SqlDataReader reader = comm.ExecuteReader();  
            if (reader != null)  
            {  
                while (reader.Read())  
                {  
                    int Id = reader.GetInt32(0);  
                    string FIO = reader.GetString(1);  
                    teachers.Add(Id, FIO);  
                }  
            }  
        }  
        catch  
        {  
            teachers.Clear();  
            teachers.Add(0, "немає викладачів");  
        }  
        finally  
        {  
            conn.Close();  
        }  
    }  
    return teachers;  
}
```

Змн.	Арк.	№ докум.	Підпис	Дата

КС КРБ 123.181.00.00 ПЗ

Арк.

44

Інкапсуляція з'єднань в окремі атомарні методи та упаковка значень полів таблиць в об'єкти що зберігаються в пам'яті (наприклад, Dictionary чи DataSet) має на меті зниження апаратних навантажень та підвищення масштабованості програмних систем. Крім того, автономна модель ADO.NET дозволяє працювати з даними як із об'єктами, проводити різноманітні маніпуляції із структурою даних не впливаючи на джерело вихідних значень.

Наприклад, наступний код методу класу DataLayer демонструє проєкцію рядків таблиці на стовпці з метою покращення сприйняття кінцевим користувачем та показана у вигляді лістингу 3.2.

Лістинг 3.2 – метод FlipDataSet ()

```
public static DataSet FlipDataSet(DataSet my_DataSet)
{
    DataSet ds = new DataSet();
    foreach (DataTable dt in my_DataSet.Tables)
    {
        DataTable table = new DataTable();
        for (int i = 0; i <= dt.Rows.Count; i++)
        {
            table.Columns.Add(Convert.ToString(i));
        }
        DataRow r=null;
        for (int k = 0; k < dt.Columns.Count; k++)
        {
            r = table.NewRow();
            r[0] = dt.Columns[k].ToString();
            for (int j = 1; j <= dt.Rows.Count; j++)
                r[j] = dt.Rows[j - 1][k];
        }
        table.Rows.Add(r);
        ds.Tables.Add(table);
    }
    return ds;
}
```

Крім того, для підтримки функціонування довідників в прикладному додатку створено модель наборів таблиць та об'єднано їх в DataSet з допомогою автоматизованих засобів проєктування середовища Visual Studio. Оскільки модель DataSet повністю відображає схему бази даних з допомогою створення нових користувацьких типів об'єктів реляційної моделі її називають типізованим

					КС КРБ 123.181.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		45

набором даних. Типізовані набори даних базуються на використанні спеціалізованих класів платформи .NET, що утворюють зв'язану структуру схожу на схеми в реляційних сховищах даних.

Кореневим елементом цієї ієрархії є клас DataSet, що представляє собою об'єкт відображення певної частини чи цілої схеми сховища даних. Інші елементи входять в нього як типізовані колекції та є його властивостями.

Типізовані DataSet (рис. 3.2) будуються на основі представлення схеми бази даних як певного набору DataTable, зв'язків між ними та методів вибірки, оновлення, видалення та вставки. Методи маніпуляції можуть бути представлені запитамі, автогенерованими чи користувацькими збереженими процедурами.

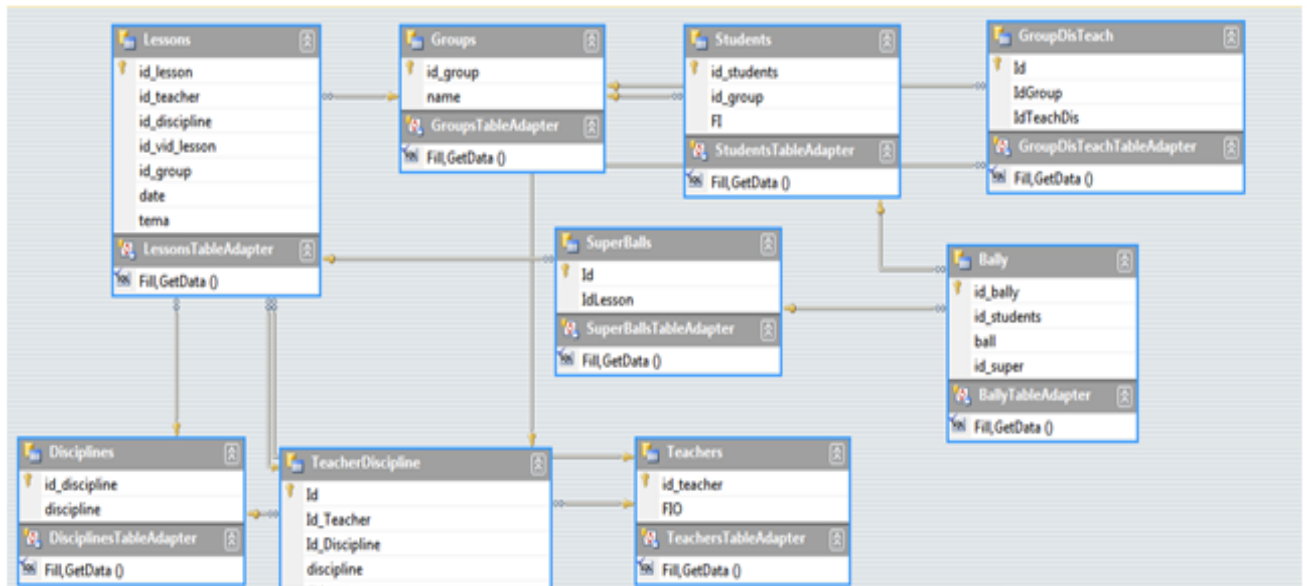


Рисунок 3.2 – Вигляд XSD-схеми типізованого набору DataSet

Згенерована на основі структури бази даних XSD схема типізованого набору DataSet дозволяє автоматизувати маніпуляції із даними, підвищити швидкодію та надійність обміну інформацією між прикладним додатком та базою даних.

Реалізація рівня доступу до даних на основі технології ADO.NET дозволила забезпечити високий рівень інтеграції даних в клієнтський інтерфейс, спростила програмне керування даними та одночасно дозволила зменшити навантаження на апаратні ресурси.

3.2 Розробка і тестування програмного забезпечення комп'ютеризованої системи контролю та обліку занять

Для розробки клієнтського графічного інтерфейсу програмного забезпечення системи контролю та обліку навчальних занять обрано технологію Windows Forms.

Важливою особливістю роботи додатків WinForms є орієнтованість на події та їх обробку. Наприклад є можливим реагувати програмно на натискання кнопки, що розташована в панелі навігатора та виконувати певний програмний код. Програмний код, який наведений у лістингу 3.3, демонструє обробку події натиснення на кнопку «Зберегти» у вікні керування даними викладачів.

Лістинг 3.3 – Метод реагування на натиснуту кнопку

```
private void saveToolStripButton_Click(object sender, EventArgs e)
{
    try
    {
        teachersBindingSource.EndEdit();
        teachersTableAdapter.Update(teacherJournalDS);
        teacherJournalDS.AcceptChanges();
        MessageBox.Show("Вдало збережено", "Результат операції", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    catch
    {
        MessageBox.Show("Помилка при збереженні", "Результат операції", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

					КС КРБ 123.181.00.00 ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

Виконання цієї частини коду має на меті збереження даних у таблицю бази даних. У разі успішного чи невдалого завершення операції збереження користувач отримає відповідне повідомлення.

Графічний користувацький інтерфейс комп'ютеризованої системи контролю та обліку навчальних занять розроблено відповідно до вимог функціональності, простоти у використанні та зручності для кінцевого користувача. Основні елементи стартового вікна (рис. 3.3) розділені на кілька логічних зон із допомогою групуючи контейнерів та вкладок.

Зверху стартової форми міститься група фільтрів, що призначені для вибору опцій відображення занять в електронному журналі. Ці фільтри включають випадні списки вибору викладача, групи та дисципліни. Варіанти вибору в списках фільтру відображаються згідно даних в довідниках та зв'язків між дисциплінами, викладачами і групами, що задані в базі даних.

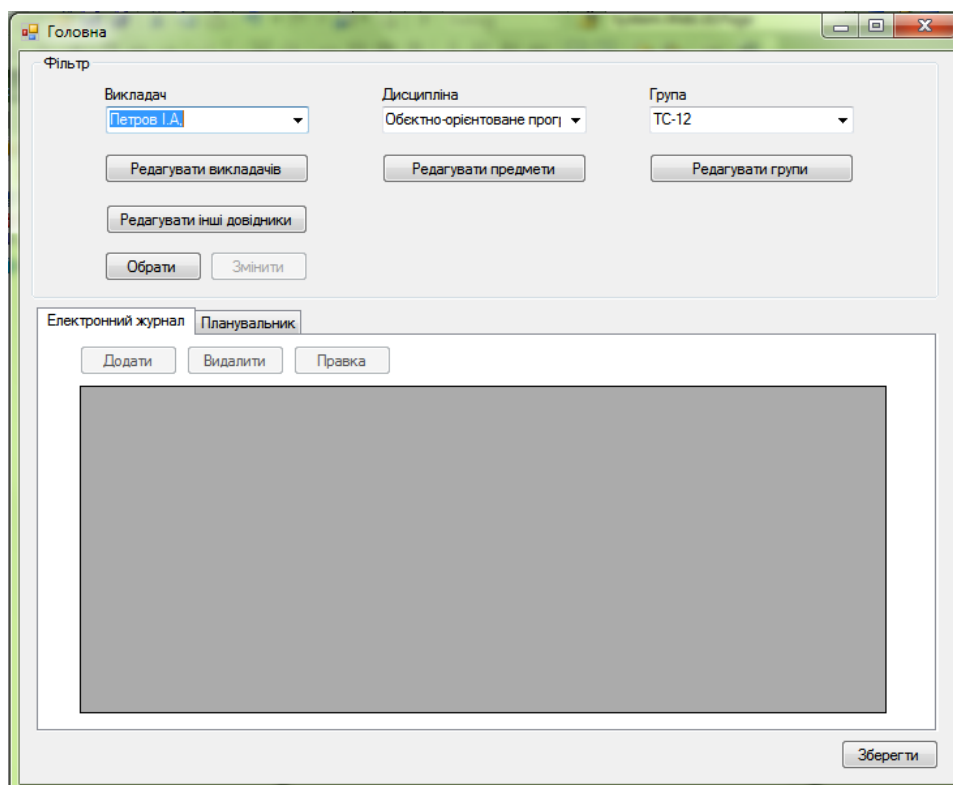


Рисунок 3.3 – Вигляд стартового вікна програми після запуску

Нижче групи фільтрів розташована група кнопок, що призначені для виклику адміністративних засобів електронного журналу викладача – довідників

Змн.	Арк.	№ докум.	Підпис	Дата

викладачів, дисциплін, груп. Одна із кнопок групує функції редагування інших записів – прив’язки предметів до викладачів, та визначення їх зв’язків із групами.

Форми адміністрування викладачів (рис. 3.4), предметів (рис. 3.5) та груп (рис. 3.6) мають схожий інтерфейс, що розроблений на основі компонентів відображення DataGridView, компоненту навігатора та полів для вводу інформації, як показано на рис. 3.4. Компонент навігатора також забезпечує виконання операцій створення, збереження чи видалення записів в базі даних.

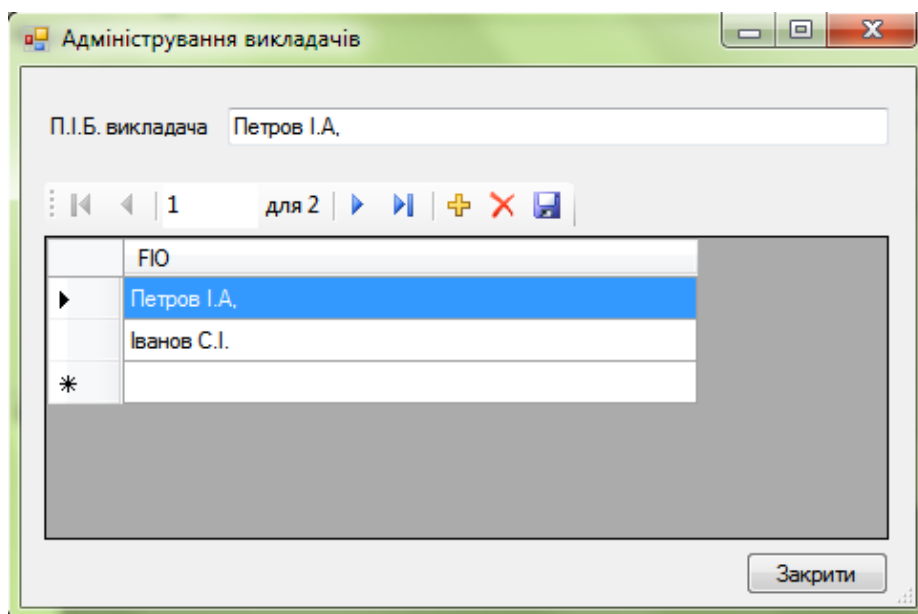


Рисунок 3.4 – Форма адміністрування записів викладачів

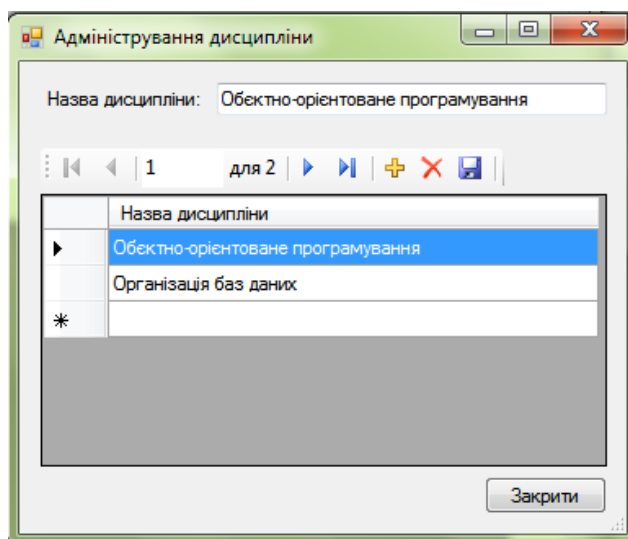


Рисунок 3.5 – Вигляд вікна редагування записів про дисципліни

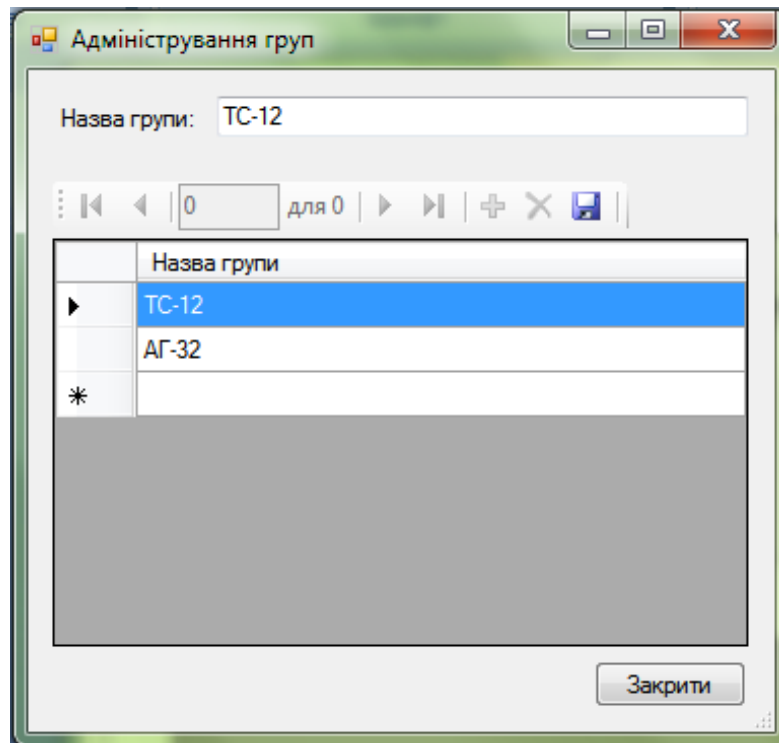


Рисунок 3.6 – Форма адміністрування груп

У випадку виклику редагування інших довідників, користувачам надається вибір варіантів довідників із випадного списку, як показано на рис. 3.7. Відокремлення групи зв'язаних довідників має на меті підвищити масштабованість системи та дозволити подальше розширення списку довідників при потребі.

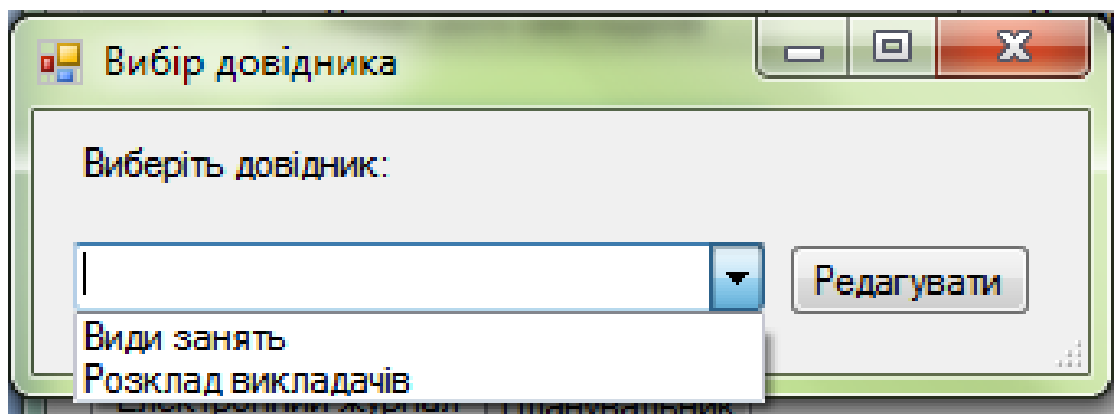


Рисунок 3.7 – Вибір довідника для редагування

Приклад форми редагування виду занять приведено на рис. 3.8.

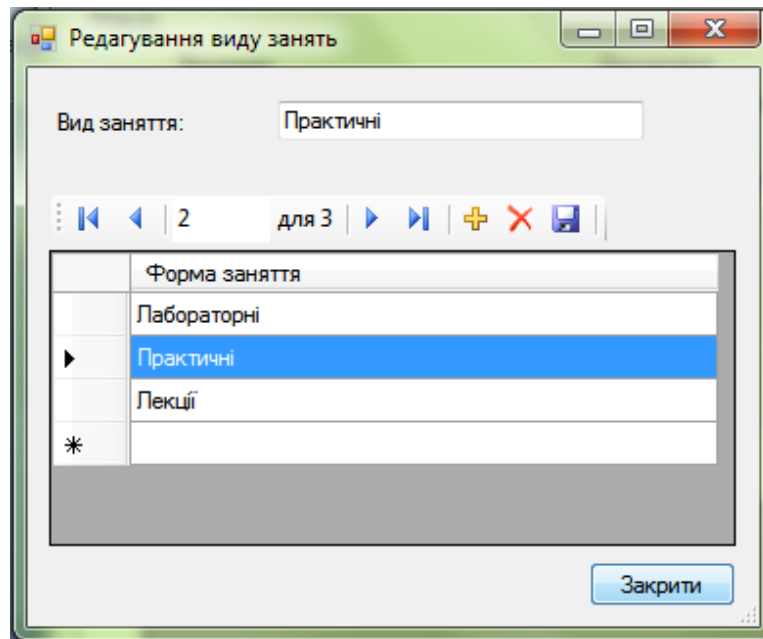


Рисунок 3.8 – Вигляд форми редагування видів занять

Форма керування розкладом викладача (рис. 3.9) дозволяє зв'язувати записи викладачів та відповідні предмети, що вони викладають.

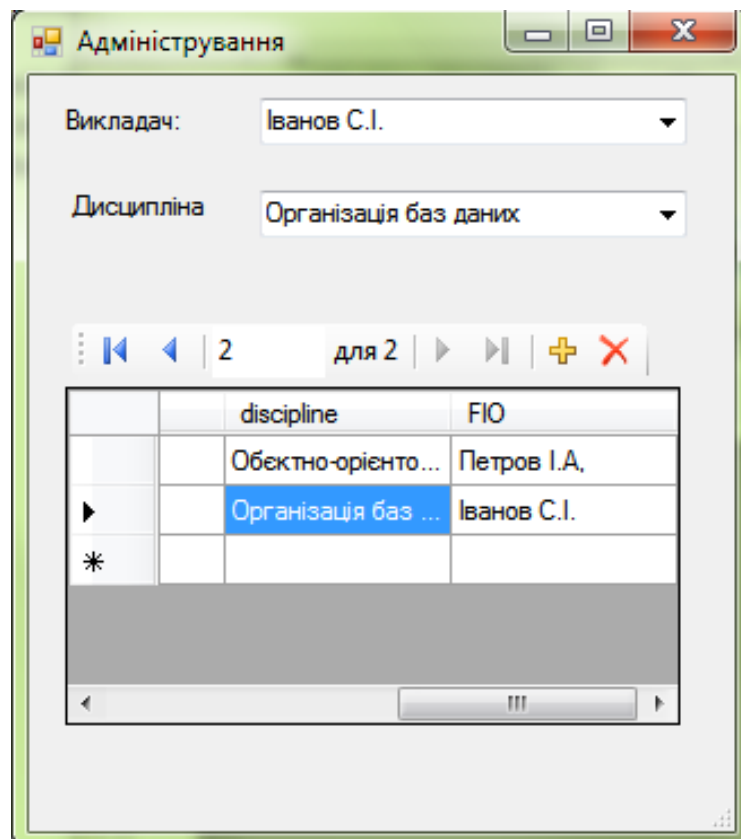


Рисунок 3.9– Вигляд форми адміністрування розкладу викладачів

Основним компонентами форми, показаної на рис. 3.9, є табличний компонент відображення, навігатор із засобами маніпулювання даними та випадні списки для вибору викладачів та предметів.

При виборі відповідних фільтрів на головній формі та натискання кнопки «Обрати» табличний компонент вкладки «Електронний журнал» відобразить список студентів обраної групи та заняття по предмету, як показано на рис. 3.10. Табличний компонент дозволяє виставляти оцінку студента за заняття, або відмітку про відвідування – наприклад літеру «н» усім відсутнім.

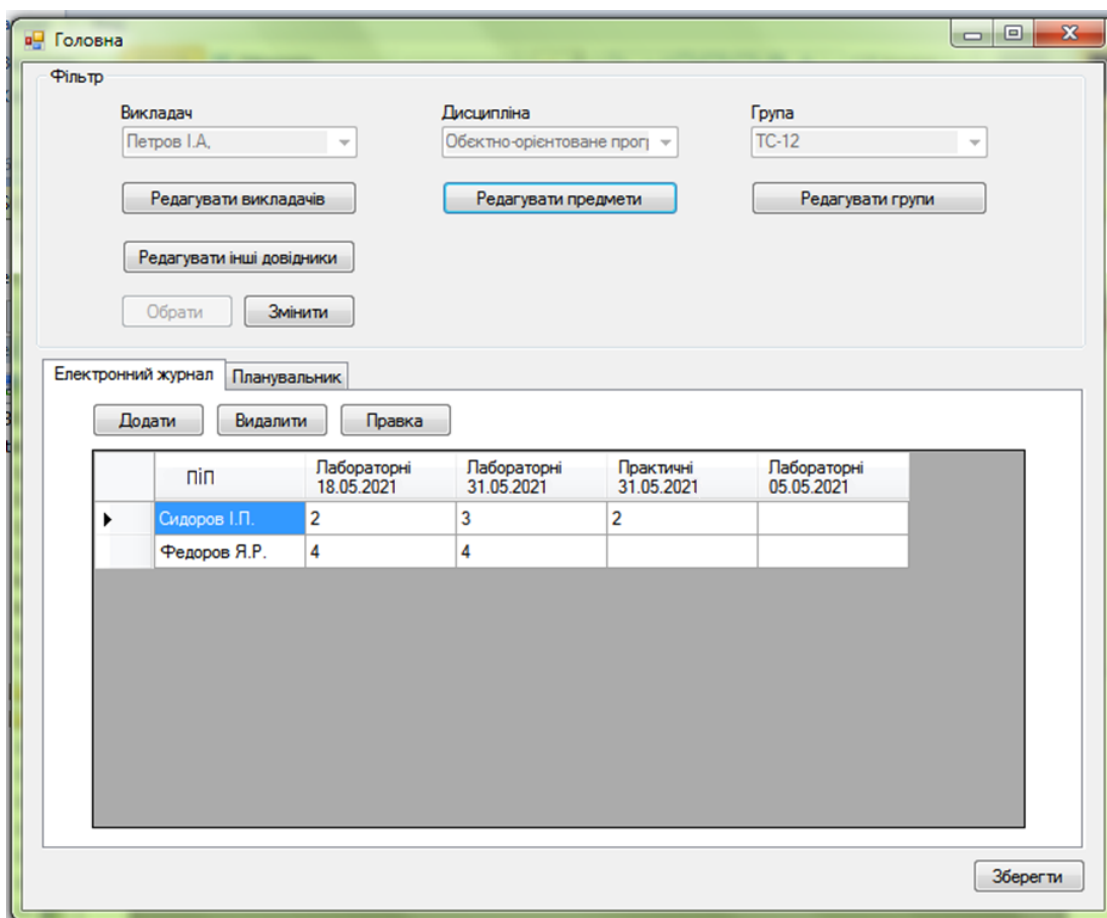


Рисунок 3.10 – Вигляд електронного журналу викладача

У верхній частині вкладки «Електронний журнал» містяться кнопки редагування записів журналу – додавання, видалення та зміни записів. При натисканні кнопки «Додати» (рис. 3.11) користувач може в діалоговому вікні «Додавання заняття» обрати вид та дату проведення заняття.

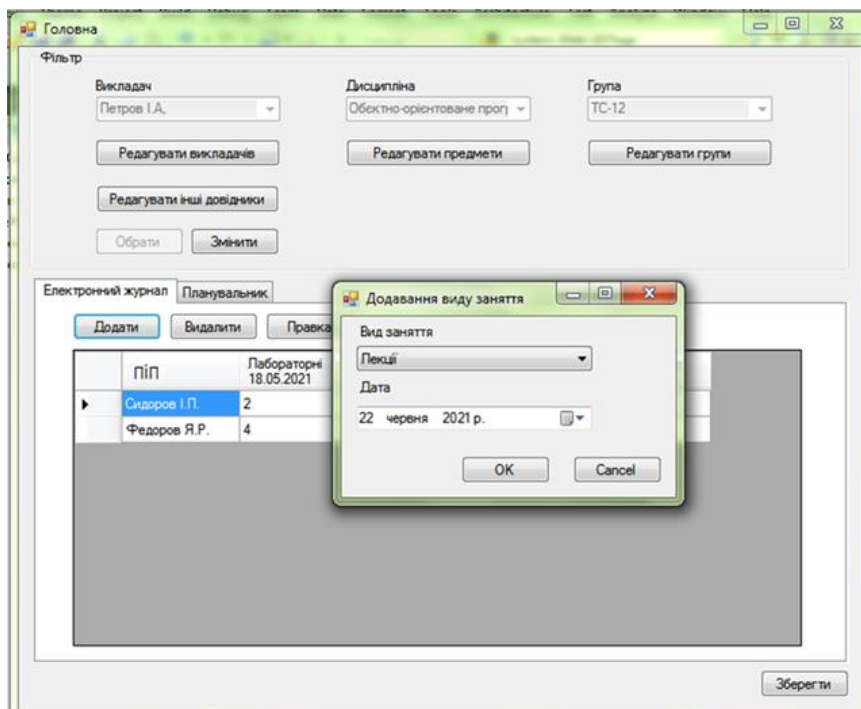


Рисунок 3.11 – Додавання нового заняття

При виборі кнопки «Правка» (рис. 3.12) користувач отримає можливість змінити дані про вид заняття та дату проведення у модальному діалоговому вікні. При натисканні кнопки «Видалити» користувач має можливість видалити відповідний запис із списку занять.

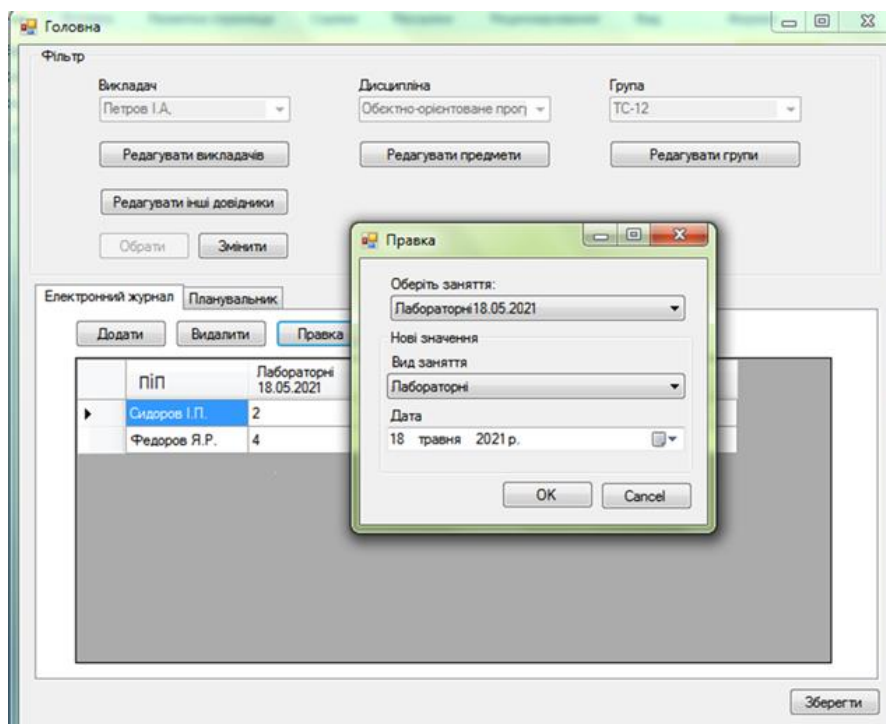


Рисунок 3.12 – Правка даних заняття

Змн.	Арк.	№ докум.	Підпис	Дата

Планувальник занять дозволяє викладачу вести зручний облік тем, що розглядаються на заняттях (рис. 3.13). Список занять формується відповідно до введеного в електронному журналі розкладу.

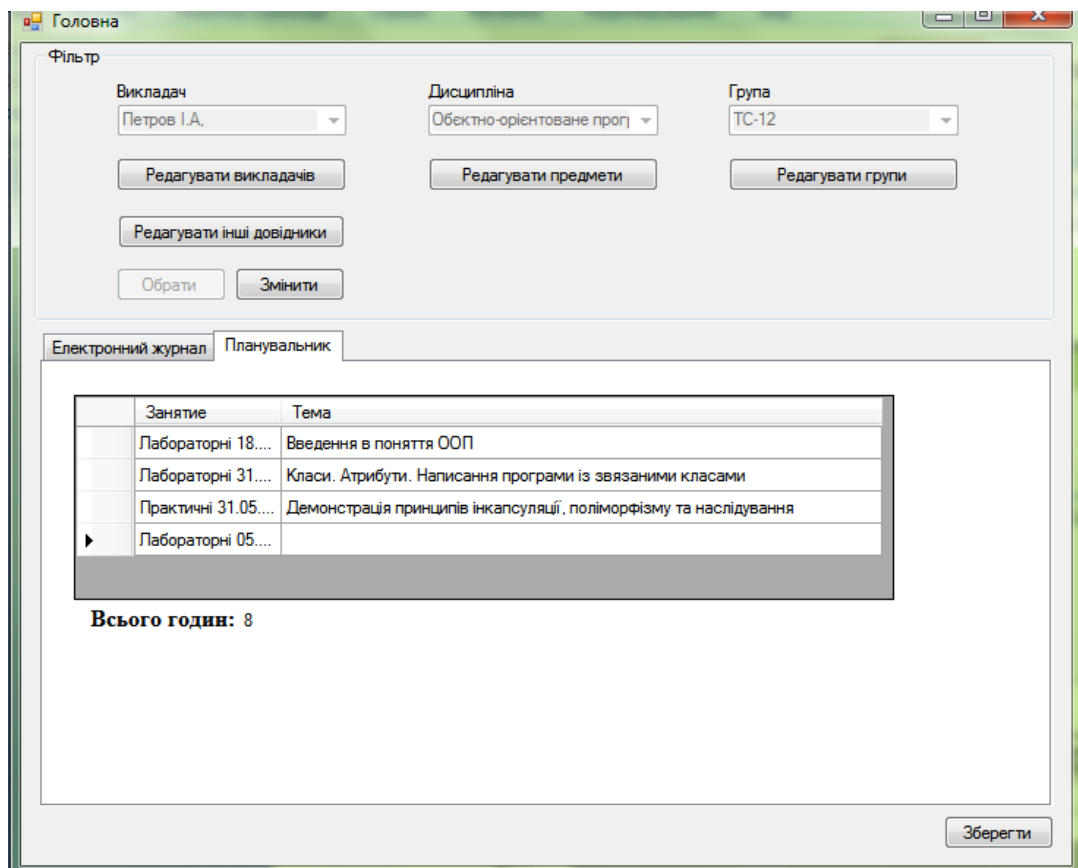


Рисунок 3.13 – Вигляд планувальника

Розроблений користувацький інтерфейс комп'ютеризованої системи контролю та обліку навчальних занять дозволяє виконувати редагування та внесення даних в довідниках, проводити облік оцінок та відміток про відвідування занять, дозволяє вносити теми занять. Основною задачею, яку вирішено за рахунок розробки прикладного додатку комп'ютеризованої системи є систематизація та автоматизація процесу обліку успішності студентів, централізований збір та збереження даних про процес навчання та контролю знань.

Користувацький прикладний інтерфейс є невід'ємною складовою програмного засобу та забезпечує виконання та підтримку прикладних задач, які виникають при проведенні навчального процесу.

3.3 Організація доступу до джерел даних із використанням технології ADO.NET, СКБД SQL Server та Visual Studio

Для організації доступу до віддаленого сервера баз даних із використанням технології ADO.NET необхідно виконати наступні етапи:

- налаштування СКБД SQL Server для віддаленого доступу із розмежуванням прав;
- створення з'єднання із допомогою інструменту «Оглядач серверів» середовища Visual Studio;
- створення типізованого набору DataSet із використанням XSD-схем.

Налаштування СКБД SQL Server для віддаленого доступу полягає у налаштуванні доступу до FILESTREAM та СКБД. Для налаштування віддаленого доступу до FILESTREAM необхідно використати візуальну оболонку Sql Server Configuration Manager та встановити опції віддаленого доступу, як показано на рис. 3.14.

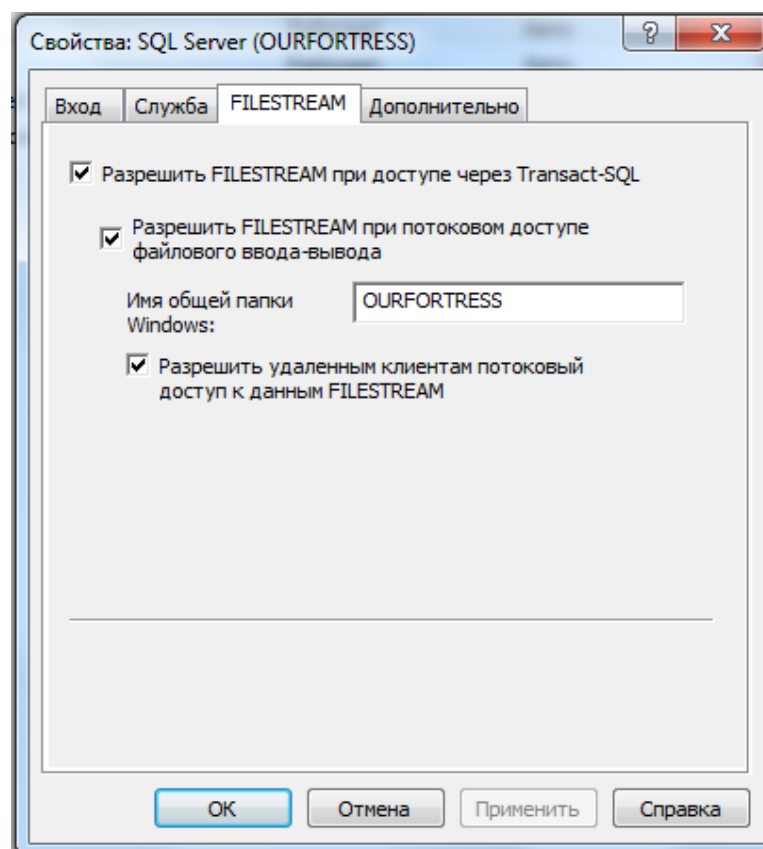


Рисунок 3.14 – Налаштування віддаленого доступу до даних FILESTREAM

Для встановлення віддаленого доступу до СКБД SQL Server необхідно дозволити використання підключень через стек протоколів TCP/IP. Це налаштування виконується з використанням візуальної оболонки майстра конфігурації в розділі «Мережева конфігурація SQL Server» (рис. 3.15).

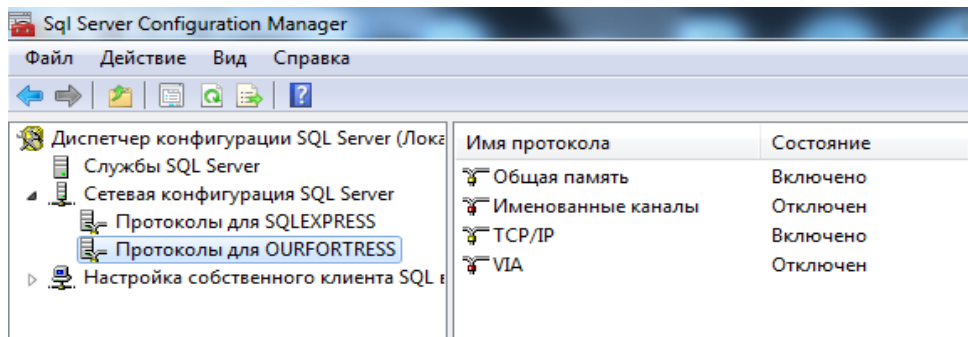


Рисунок 3.15 – Налаштування дозволу на використання стеку протоколів TCP/IP для доступу до СКБД

Дозвіл віддаленого доступу до сервера СКБД налаштовується з допомогою візуальних засобів SQL Server Management Studio у розділі Connections налаштувань сервера (рис. 3.16).

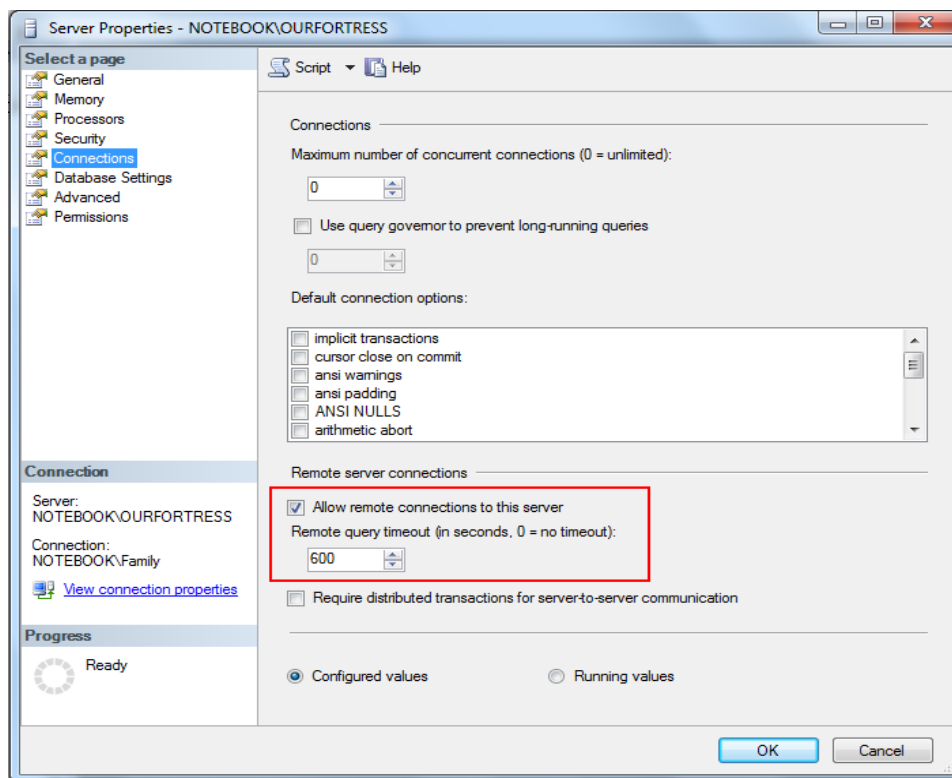


Рисунок 3.16 – Налаштування віддаленого доступу до сервера баз даних

Змн.	Арк.	№ докум.	Підпис	Дата

Підключення до бази даних виконується з допомогою засобів модуля «Оглядач серверів» інтегрованого середовища розробки Visual Studio.

Запустити «Оглядач серверів» можна за допомогою виконання пунктів меню «Вигляд»-«Оглядач серверів». Додавання з'єднання виконується за допомогою пункту контекстного меню розділу «Data connections»-«Add connection...».

В діалоговому вікні майстра підключень необхідно вказати ім'я сервера бази даних, вид аутентифікації, користувача та пароль, ім'я цільової бази даних (рис. 3.17).

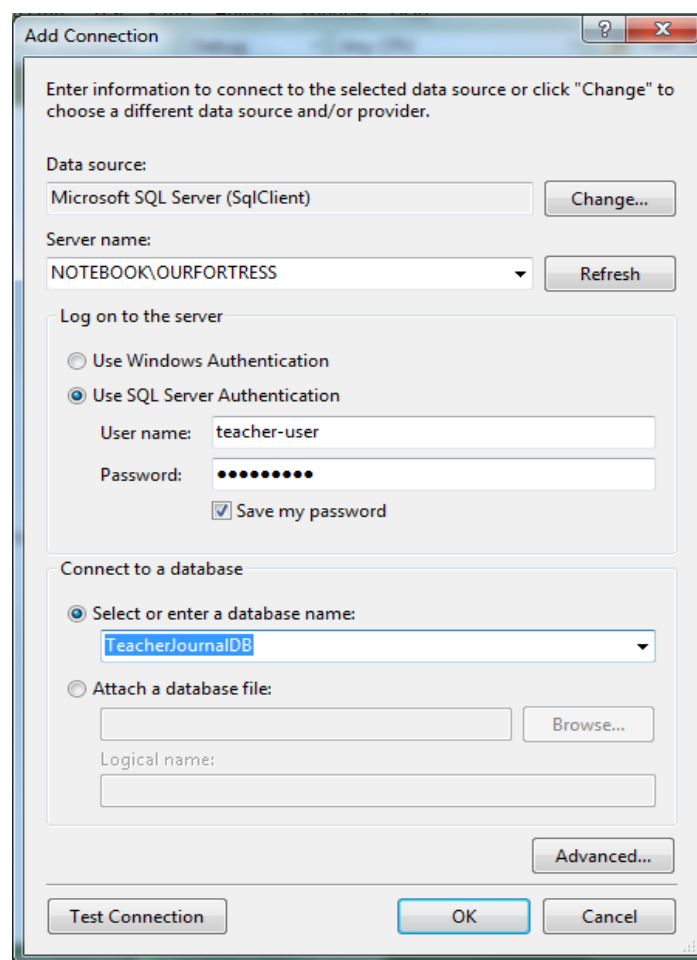


Рисунок 3.17 – Налаштування підключення до бази даних

Підключення до бази даних з'явиться у вікні оглядача серверів, стане можливою навігація по дереві об'єктів цільової бази даних та доступ до вмісту таблиць.

Для створення типізованого набору даних необхідно додати XSD – схему в ієрархію проекту з допомогою команди контекстного меню «Add» - «New item...» - «Data set». Генерація XSD – схеми виконується у візуальному редакторі за допомогою техніки Drag & Drop, тобто перетягуванням відповідних таблиць із вікна оглядача серверів. Результат створеного типізованого набору показано на рис. 3.18.

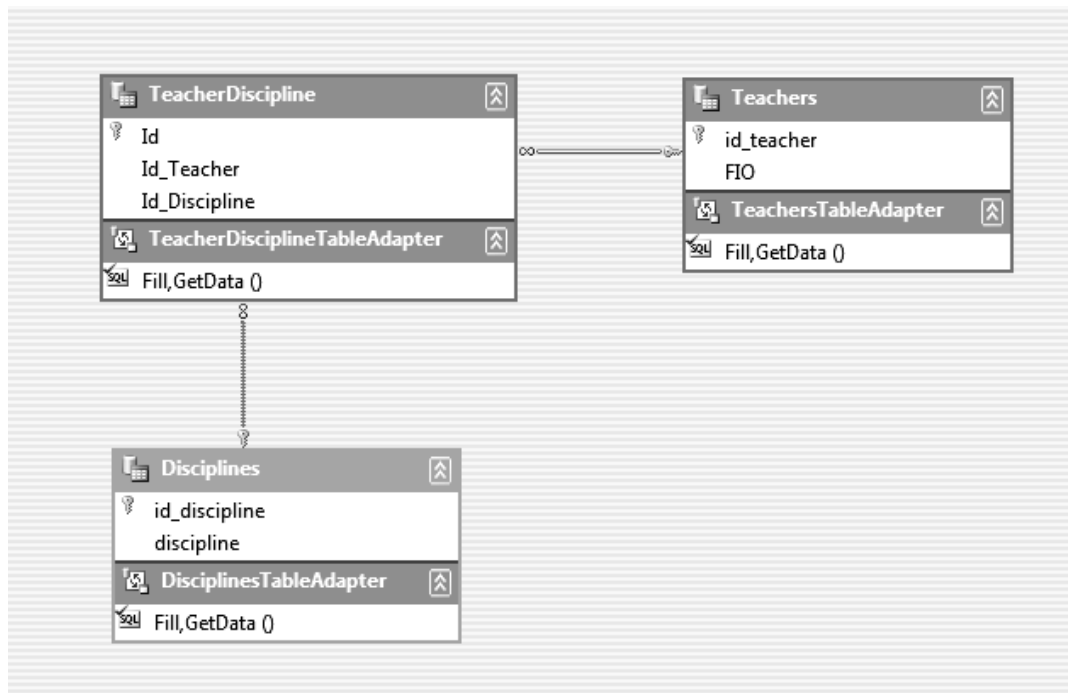


Рисунок 3.18 – Вигляд типізованих наборів даних у візуальному редакторі

Після заповнення XSD – схеми необхідними таблицями засоби автоматизованого проектування виконують генерацію відповідних схем XML та об’єктів платформи .NET. Універсальна структура типізованих наборів даних надає широкі можливості підвищення автономності та масштабованості логіки доступу до сховища даних в програмних системах.

Отже, використання засобів платформи ADO.NET та типізованих наборів даних дає значні переваги при формуванні рівнів доступу до даних при виконанні розробки програмних засобів планування та організації часу, таких як система контролю та обліку навчальних занять.

РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Суть та зміст управління охороною праці

Основними завданнями управління охороною праці є:

1) опрацювання заходів щодо здійснення державної політики з охорони праці на регіональному та галузевому рівнях;

2) підготовка, прийняття та реалізація заходів, спрямованих на забезпечення:

- належних, безпечних і здорових умов праці;
- утримання в належному стані виробничого устаткування, будівель і споруд, інженерних мереж, безпечного ведення технологічних процесів;
- необхідних засобів індивідуального захисту для працівників;
- організації і проведення навчання працівників з питань охорони праці;
- пропаганди охорони праці;
- обліку, аналізу та оцінки стану умов і безпеки праці; -професійного добору працівників окремих спеціальностей;
- страхування працівників від нещасного випадку на виробництві та профзахворювань;

3) організаційно-методичне керівництво на регіональному та галузевому рівнях;

4) стимулювання інтеграції управління охороною праці в єдину систему загального управління організацією виробництва;

5) широке впровадження позитивного досвіду у сфері охорони праці.

Основні функції управління охороною праці:

- організація та координація робіт у галузі охорони праці;
- облік, аналіз та оцінка показників стану умов та безпеки праці;

					КС КРБ 123.181.00.00 ПЗ			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		Українець У.В.			<i>Безпека життєдіяльності, основи охорони праці</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Аркуші</i>
<i>Перевірив.</i>		Тиш Є.В.					59	
<i>Консульт.</i>		Пилипець М.І.				<i>ТНТУ, каф. КС, гр. СІс-44</i>		
<i>Н. Контр.</i>		Луцик Н.С.						
<i>Затверд.</i>		Осухівська Г.М.						

- планування та фінансування робіт;
- контроль за дотриманням вимог нормативно-правових актів з питань охорони праці.

Нормативно-правове забезпечення управління охороною праці має вдосконалюватися у таких напрямках:

- необхідно продовжити перебудову чинної нормативно-правової бази з охорони праці з урахуванням сучасних умов, вимог законодавства України, міжнародних або європейських норм;
- після прийняття нової редакції Закону України "Про охорону праці" слід переглянути відповідні нормативно-правові акти;
- проаналізувати стан нормативно-правової бази, визначити пріоритети щодо черговості перегляду нормативно-правових актів з охорони праці;
- необхідно забезпечити розробку та реалізацію в кожній галузі перспективних і поточних планів нормотворчої діяльності та опрацювання проектів ДНАОП на рівні сучасних вимог;
- на допомогу суб'єктам малого й середнього бізнесу ННДІОП опрацьовує довідково-методичні матеріали з питань охорони праці.

Першочерговим у системі управління охороною праці є забезпечення органів державного управління охорони праці та служб охорони праці підприємств, установ, організацій кваліфікованими фахівцями з охорони праці.

Належна кваліфікація й обізнаність усіх працівників із питань охорони праці є запобіжником ризику отримати виробничу травму чи професійне захворювання. Тому у процесі реформування управління охороною праці одним із найбільш пріоритетних напрямів є підвищення рівня знань працівників із цих питань, що має забезпечуватися у закладах освіти і безперервно шляхом навчання працівників у процесі їх трудової діяльності.

Для підвищення рівня знань фахівців із питань охорони праці необхідно:

- опрацювати проект положення "Про підготовку, перепідготовку та підвищення кваліфікації працівників системи Держнаглядохоронпраці";

					КС КРБ 123.181.00.00 ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

– розробити й реалізувати комплекс заходів щодо визначення пріоритетних питань при підготовці та підвищенні кваліфікації державних інспекторів з охорони праці з урахуванням наукових досліджень, досягнень та технічних рішень щодо створення безпечних умов праці в галузях виробництва.

Інформаційне забезпечення в галузі охорони праці має здійснюватися органами управління охороною праці на всіх рівнях і потребує вдосконалення шляхом визначення та поширення міжнародного й вітчизняного досвіду щодо пропаганди безпечних методів і засобів праці, вирішення інших актуальних питань у цій сфері із залученням сучасних інформаційних технологій, ЗМІ, оперативного розповсюдження посібників, пам'яток, методик, листівок відповідного спрямування.

ННДІОП має забезпечити збирання, обробку й доведення до кожного підприємства незалежно від сфери управління (галузевого чи регіонального рівня) інформації з питань управління та нагляду за охороною праці.

Для зниження ризиків, пов'язаних із виробничим устаткуванням, технологічними процесами, будівлями й спорудами, необхідно:

– переглянути нормативну базу, що регламентує безпечність виробничого устаткування, технологічних процесів, будівель і споруд, привести її у відповідність до вимог директив Європейського Союзу;

– удосконалити порядок проведення експертизи устаткування, технологічних процесів, будівель і споруд на їх відповідність вимогам безпеки з урахуванням міжнародних та європейських норм;

– ужити заходів щодо виведення з експлуатації (поетапно) морально застарілого і фізично зношеного виробничого устаткування, будівель, споруд тощо.

Враховуючи те, що протягом останніх років організація виробництва засобів індивідуального захисту (ЗІЗ) в Україні не дає очікуваних результатів, необхідно докорінно переглянути підхід до вирішення цієї проблеми, використовуючи досвід Білорусі, Литви, Латвії, Росії. Для цього слід упровадити на території України ЗІЗ, які вже отримали відповідний міжнародний сертифікат,

					КС КРБ 123.181.00.00 ПЗ	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

виробників ЗІЗ у державі зорієнтувати виключно на ті, впровадження у виробництво яких є економічно вигідним. Доцільно вивчити питання щодо заснування в Україні спільних з іноземними представництвами підприємств із виробництва таких ЗІЗ, які б відповідали вимогам європейських норм і мали відповідні міжнародні сертифікати.

Для вирішення питань, пов'язаних із обліком, аналізом та оцінкою стану умов та безпеки праці, слід:

- опрацювати (удосконалити) і забезпечити впровадження єдиної державної статистичної звітності щодо обліку, аналізу та оцінки стану безпеки й умов праці;

- законодавчо врегулювати звітність щодо обліку, аналізу та оцінки стану безпеки й умов праці підприємств недержавної форми власності;

- надати матеріальну підтримку ННДІОП шляхом включення до державного бюджету витрат, пов'язаних із проведенням обґрунтованого аналізу стану охорони праці, наглядової діяльності та їх взаємозв'язку, опрацюванням періодичних аналітичних матеріалів щодо стану охорони праці в Україні.

Планування робіт з охорони праці має здійснюватися з урахуванням результатів аналізу й оцінки стану охорони праці, визначення пріоритетних напрямів діяльності.

Фінансування робіт з охорони праці. Необхідно створити належне правове підґрунтя і забезпечити фінансування заходів з охорони праці на державному, галузевому і регіональному рівнях за рахунок коштів:

- Фонду соціального страхування від нещасних випадків, виділених на профілактику виробничого травматизму й профзахворювань;

- державного бюджету і місцевих бюджетів - для часткового фінансування (разом із коштами Фонду соціального страхування від нещасних випадків) Національної, галузевих і регіональних програм поліпшення стану безпеки, гігієни праці та виробничого середовища або інших цільових програм з охорони праці, а також заходів з охорони праці, передбачених програмами соціально-економічного і культурного розвитку України та її адміністративно-

					КС КРБ 123.181.00.00 ПЗ	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

територіальних одиниць; при цьому кошти на охорону праці в державному й місцевих бюджетах виділяються окремими рядками;

- інших джерел фінансування, не заборонених законодавством.

Система контролю за витрачанням коштів, виділених на охорону праці на рівні підприємства, має бути вдосконалена таким чином, щоб забезпечити їх спрямування за цільовим призначенням відповідно до Переліку заходів та засобів з охорони праці, що затверджується Кабінетом Міністрів України.

4.2 Підбирання оптимальних параметрів мікроклімату на робочих місцях з ПК

Висока чи низька температура повітря в приміщенні з ПК негативно впливає на функціональний стан користувача. Недостатня вологість в приміщенні призводить до надмірного висихання слизових оболонок очей, носа, горла та до нагромадження зарядів статичної електрики, що утворюються в процесі роботи ПК.

Разом з тим недопустима вологість повітря більше 75%. На робочих місцях користувачів ПК параметри мікроклімату відповідають вимогам ДСН 3.3.6.042-99 “Санітарні норми мікроклімату виробничих приміщень”, ДСан Пін 3.3.2-007-98, ДНАОП 0.00-1.31-99.

Холодна пора року – період року, який характеризується середньодобовою температурою зовнішнього повітря, що дорівнює $+10^{\circ}\text{C}$ і нижче.

Тепла пора року – період року, який характеризується середньодобовою температурою зовнішнього середовища вище $+10^{\circ}\text{C}$.

До категорії 1а належать роботи, що виконуються сидячи і не потребують фізичного напруження, при яких витрата енергії дорівнює 105-140Вт (90-120 ккал/год.). У табл. 4.1 приведені норми мікроклімату для приміщень з ВДТ ЕОМ та ПЕОМ (ПК).

					КС КРБ 123.181.00.00 ПЗ	Арк.
						63
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 4.1 – Норми мікроклімату для приміщень з ВДТ ЕОМ та ПЕОМ

Пора року	Категорія робіт**	Температура повітря, °С	Відносна вологість повітря, %	Швидкість руху повітря, м/с
Холодна	легка-1а	22-24	60-40	0,1
легка-1б	21-23	60-40	0,1	
Тепла	легка-1а	23-25	60-40	0,1
легка-1б	22-24	60-40	0,2	

До категорії 1б належать роботи, що виконуються сидячи, стоячи або пов'язані з ходінням та супроводжуються деяким фізичним напруженням, при яких витрата енергії дорівнює 141-175 Вт (121-150 ккал/год.).

Повітря яким дихають користувачів ПК складається з позитивно та негативно заряджених часток-іонів. Давно відомо про цілющий вплив на людину негативних іонів. Вони підвищують рівень гаммаглобуліна в крові, що збільшує опір організму хвороботворним бактеріям, а також стимулює вироблення бетаендорфіна, що інколи називають “гормоном щастя”, який позитивно впливає на настрій, запобігаючи депресіям та підвищуючи працездатність.

При роботі з ВДТ і ПК іонний склад повітря на робочому місці користувача змінюється. Уже через 5 хвилин роботи кількість легких іонів знижується у 8 разів, а через 3 години є майже нульовою. Також зменшується концентрація середніх та важких негативно заряджених часток, а концентрація позитивних іонів зростає, що шкідливо впливає на здоров'я користувача комп'ютера [19].

Такий вплив проявляється в зниженні його працездатності (швидка втомлюваність, головний біль), в погіршенні короткочасової пам'яті, діяльності серцево-судинної системи, бронхо-легеневого апарату (збільшення частоти пульсу та дихання (із-за недостатнього поступлення кисню в кров)).

У приміщеннях з комп'ютерами забезпечується 3-кратний обмін повітря за годину. Адже під час розумової праці мозок людини споживає в 9-10 разів більше кисню, а ПК його забирає, виникає кисневе голодування.

Для забезпечення нормованого мікроклімату та рівня іонізації повітря на робочих місцях користувачів застосовують припливно-витяжну вентиляцію чи систему кондиціонування повітря, прилади зволоження та/або установки генерації негативних іонів (аероіонізатори). Знизити деіонізацію повітря в зоні дихання користувача ПК дозволяє також встановлений на монітор захисний заземлений екран.

На сьогоднішній день деякі компанії починають випускати монітори із влаштованими іонізаторами повітря. Компанія Samsung Electronics заявила, що почала продавати РК-монітори із вмонтованими іонізаторами повітря. При включенні іонізатора він починає випромінювати негативно заряджені іони, які перешкоджають електризації пилу та скупченню бактерій біля робочого місця користувача. Сучасні модифіковані аероіонізатори дозволяють одночасно очищувати і іонізувати повітря негативними іонами.

У великих приміщеннях використовують аероіонізатори типу люстри Чижевського.

Вимоги до вмісту шкідливих речовин у повітрі робочої зони, відповідно до ГОСТ 12.1.005-88 «Загальні санітарно-гігієнічні вимоги до повітря робочої зони», наступні:

- концентрація озону – не більше 0,1 мг/м³;
- вміст оксидів азоту – не більше 5 мг/м³;
- вміст пилу – не більше 4 мг/м³.

Таким чином, дотримання вимог щодо вмісту аеронів, застосування засобів іонізації повітря та знешкодження шкідливих речовин у повітрі дають змогу дотримуватись рекомендацій стандартів щодо мікроклімату у виробничих приміщеннях з використанням ПЕОМ і тим самим забезпечити оптимальні умови праці користувачів ПК.

					КС КРБ 123.181.00.00 ПЗ	Арк.
						65
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У даній кваліфікаційній роботі розроблено проект комп'ютеризованої системи контролю та обліку навчальних занять до складу якої входять мережева інфраструктура закладу освіти та програмне забезпечення управління процесом контролю та обліку занять.

Мережева інфраструктура побудована на основі фізичної і логічної топології комп'ютерної мережі у вигляді розширеної зірки. Забезпечення зв'язку між робочими станціями і сервером бази даних реалізовано за допомогою активного комутаційного обладнання фірми D-Link, що включає два некерованих комутатори D-Link 1005 P, чотири некерованих комутатори DES-1016P з 16 портами 10/100Base-TX1, один WebSmart комутатор 2 рівня з 8 портами 10/100Base-T + 2 комбо-портами 10/100/1000 BASE-T/SFP та один комутатор з 8 портами DGS-1008P.

У проекті комп'ютеризованої системи розроблено схему комутаційних з'єднань вузлів комп'ютерної мережі, IP-адресну схему, що дозволяє повністю забезпечити мережеву інфраструктуру для управління процесом контролю та обліку навчальних занять.

Керування процесом контролю та обліку навчальних занять забезпечено на рівні програмного забезпечення, розроблено на основі архітектури клієнт-сервер засобами мови програмування C#. База даних побудована на основі реляційного підходу та з використанням системи керування базами даних MS SQL Server.

Графічний інтерфейс користувача спроектовано за допомогою технології Windows Form, що створює зрозумілий та зручний у використанні інструмент контролю та обліку навчальних занять.

Окрім цього, у роботі описано принцип організації доступу до джерел даних із використанням технології ADO.NET, СКБД SQL Server та Visual Studio.

Комплекс мережевої інфраструктури та програмного забезпечення складають повноцінну комп'ютеризовану систему контролю та обліку навчальних занять.

					КС КРБ 123.181.00.00 ПЗ	Арк.
						66
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Закон України «Про повну загальну середню освіту». URL: <https://osvita.ua/legislation/law/2232/> (дата звернення 08.05.2021 р.)
2. Микитишин А.Г., Митник М.М., Стухляк П.Д., Пасічник В.В. Комп'ютерні мережі. Книга 1. Львів, «Магнолія 2006». 2013. 256 с.
3. Микитишин А.Г., Митник М.М., Стухляк П.Д., Пасічник В.В. Комп'ютерні мережі. Книга 2. Львів, «Магнолія 2006», 2014. 312 с.
4. Микитишин А.Г., Митник М.М., Стухляк П.Д. Телекомунікаційні системи та мережі. Тернопіль: Вид-во ТНТУ імені Івана Пулюя. 2016. 384 с.
5. Буров Є. Комп'ютерні мережі. 2-ге оновлене і доповн. Вид. Львів: Бак, 2003. – 584 с.
6. Воробієнко П.П., Нікітюк Л.А., Резніченко П.І. Телекомунікаційні та інформаційні мережі: Підручник для вищих навчальних закладів. К.: САММІТ-КНИГА, 2010. 640 с.
7. Таненбаум Э. Компьютерные сети. С-Пт. : Питер, 2013. 960 с.
8. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы: Ученик для вузов. 5-е изд. СПб.: Питер, 2016. 992 с.
9. Microsoft Visual Studio/ URL: https://uk.wikipedia.org/wiki/Microsoft_Visual_Studio. (дата звернення 01.05.2021 р.)
10. Learn all about biometrics and how to build a security system that uses your fingerprints as the key in this tutorial. URL: <https://maker.pro/raspberry-pi/projects/raspberry-pi-fingerprint-scanner-using-a-usb-to-serial-ttl-converter> (дата звернення 04.04.2021 р.)
11. SQL Syntax. URL: https://www.w3schools.com/sql/sql_syntax.asp дата звернення 19.04.2021 р.)
12. C# Tutorial. URL: <https://www.w3schools.com/cs/> (дата звернення 17.04.2021 р.)
13. Building a simple server client application using C#. URL: <https://codeabout.wordpress.com/2011/03/06/building-a-simple-server-client-application-using-c>. (дата звернення 27.05.2021 р.).

					КС КРБ 123.181.00.00 ПЗ	Арк.
						67
Змн.	Арк.	№ докум.	Підпис	Дата		

14. Code Cracker for C# URL: <https://visualstudiogallery.msdn.microsoft.com/ab588981-91a5-478c-8e65-74d0ff450862>nofollow. (дата звернення 10.05.2021 р.)

15. Автоматизация Code First и EF Power Tools. URL: <https://metanit.com/sharp/entityframework/2.3.php> (дата звернення 13.05.2021 р.)

16. Астелс Д., Миллер Г., Новак М. Практическое руководство по экстремальному программированию. М.: «Вильямс». 2012. 320с.

17. Гульятев А.К., Машин В.А. Проектирование и дизайн пользовательского интерфейса. С-Пб : "Корона-принт". 2000. 349 с.

18. НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями». Київ. 2018.

19. Бедрій Я. Основи охорони праці користувачів персональних комп'ютерів: навчальний посібник для студентів ВНЗ та інженерів-практиків. Навчальна книга-Богдан. 2014. 144 с.

					КС КРБ 123.181.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		68

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.

“ ___ ” _____ 2021 р

КОМП'ЮТЕРИЗОВАНА СИСТЕМА КОНТРОЛЮ ТА ОБЛІКУ

НАВЧАЛЬНИХ ЗАНЯТЬ

ТЕХНІЧНЕ ЗАВДАННЯ

на 13 листках

Вид робіт:

Кваліфікаційна робота

На здобуття освітнього ступеня «Бакалавр»

Спеціальність 123 «Комп'ютерна інженерія»

«УЗГОДЖЕНО»

Керівник кваліфікаційної роботи

_____ к.т.н., доц. Тиш Є.В.

« ___ » _____ 2021 р.

«ВИКОНАВЕЦЬ»

Студентка групи СІс-44

_____ Українець У.В.

« ___ » _____ 2021 р.

Тернопіль 2021

1 Загальні відомості

1.1 Повна назва та її умовне позначення

Повна назва теми кваліфікаційної роботи: «Комп'ютеризована система контролю та обліку навчальних занять».

Умовне позначення кваліфікаційної роботи: КС КРБ 123.181.00.00

1.2 Виконавець

Студентка групи СІс-44, факультету комп'ютерно-інформаційних систем і програмної інженерії, кафедри комп'ютерних систем та мереж, Тернопільського національного технічного університету імені Івана Пулюя, Українець Уляна Володимирівна.

1.3 Підстава для виконання роботи

Підставою для виконання кваліфікаційної роботи є наказ по університету (№ 4.7-97 від 10.02.2021 р.)

1.4 Планові терміни початку та завершення роботи

Плановий термін початку виконання кваліфікаційної роботи – 10.02.2021 р.

Плановий термін завершення виконання кваліфікаційної роботи – 23.06.2021 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 ІСО, ГОСТ, 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

Основне призначення проектованої комп'ютеризованої системи полягає у забезпеченні процесів обліку та контролю за навчальними заняттями, зокрема, відвідуваності занять учнями або студентами, їх успішності та здатності формування розкладу протягом визначеного періоду часу.

При проектуванні комп'ютеризованої системи контролю та обліку навчальних занять необхідно дослідити та/або спроектувати існуючу апаратну і комунікаційну інфраструктуру, спроектувати схему бази даних і клієнтське програмне забезпечення для зручного використання програмного забезпечення системи.

Інформаційна інфраструктура повинна базуватись на організаційній структурі закладу освіти і як наслідок локальна комп'ютерна мережа має включати необхідні апаратні пристрої для забезпечення функцій комунікації між клієнтськими станціями і сервером баз даних.

Програмне забезпечення комп'ютеризованої системи має задовольняти основним функціям щодо контролю успішності учнів, а база даних повинна містити довідники щодо груп/класів, студентів/учнів та викладачів навчальних занять.

Основним призначенням системи контролю та обліку навчальних занять є фіксація проведених офлайн навчань, що дозволить централізовано моніторити стан проведення занять та успішність учнів чи студентів.

Реалізація комп'ютеризованої системи підвищить ефективність управління процесами надання освітніх послуг та дозволить приймати рішення для оптимізації та підвищення якості освіти.

2.2 Мета створення системи

Мета створення комп'ютеризованої системи контролю та обліку навчальних занять полягає в автоматизації відповідних процесів для підвищення якості навчального процесу, а також на основі наявних даних прийняття управлінських рішень.

Досягнення мети створення системи можливе шляхом розв'язку наступних задач:

- аналіз організаційної структури закладу освіти;
- на основі особливостей приміщень закладу освіти та фізичного розташування обладнання розробка проекту фізичної топології мережі;
- проектування логічної топології мережі;
- розробка схеми комутації вузлів мережі та IP-адресної схеми;
- обґрунтування вибору апаратного забезпечення комп'ютеризованої системи;
- проектування схеми бази даних для зберігання довідників та автоматизації процесів контролю та обліку навчальних занять;
- реалізація клієнтського програмного забезпечення для роботи з базою даних.

2.3 Характеристика об'єкту

2.3.1 Основні задачі та функції об'єкту

До основних задач і функцій комп'ютеризованої системи контролю та обліку навчальних занять належать забезпечення комунікації між вузлами системи з використанням комп'ютерної мережі та управління процесами забезпечення навчального процесу з використанням прикладного програмного додатку. Сукупно компоненти комп'ютеризованої системи формують складову частину інформаційної інфраструктури закладу освіти при провадженні освітньої діяльності в режимі очного навчання.

Основні задачі, які повинна вирішити комп'ютерна мережа полягають в наступному:

- забезпечення зв'язку між комп'ютерними класами, учительською кімнатою та дирекцією закладу освіти;
- автоматизація обліку інформації, необхідної учителям;
- організація доступу до мережі Інтернет;
- встановлення та налаштування сервера баз даних;
- забезпечення процесу розмежування прав доступу та контроль над ними при зверненні до спільних ресурсів.

Функціями комп'ютерної мережі є автоматизація процесу навчання шляхом об'єднання комп'ютерів в єдиний інформаційний простір і спільного використання ресурсів.

Функції прикладного програмного додатку полягають у:

- формуванні довідників про базові сутності автоматизації процесів обліку та контролю навчальних занять;
- забезпеченні здатності фіксації проведення навчальних занять;
- управління даними щодо відвідуваності занять;
- забезпеченні можливості оцінювання знань здобувачів освіти;
- формуванні журналів успішності учнів/студентів.

3 Вимоги до системи

3.1 Вимоги до системи в цілому

В цілому, вимоги до комп'ютеризованої системи обліку та контролю можна поділити на дві основні частини: вимоги до комп'ютерної мережі та вимоги до прикладного програмного забезпечення управління відповідними процесами.

Вимогами до комп'ютерної мережі закладу освіти є:

- надійність роботи апаратних засобів;
- продуктивність апаратної засобів;
- захищеність обладнання;
- захищеність доступу до ресурсів;
- паралельний доступ до ресурсів;
- розмежування прав доступу;
- часова ефективність;
- ефективність використання ресурсів інформаційної системи.

До вимог, які в цілому висуваються до прикладного програмного забезпечення комп'ютеризованої системи належать:

- забезпечення функцій вводу, оновлення та видалення інформації про навчальні заняття;
- можливість формування інформаційних довідників про вчителів, групи, заняття, їх види і т.п;
- підтримка реляційної схеми бази даних та відповідної сигнатури;
- клієнт-серверна архітектура клієнтського додатку;
- підтримка рольової політики аутентифікації користувача при використанні програмного забезпечення;
- можливість одночасного доступу до ресурсів бази даних;
- визначена продуктивність надання результатів пошуку та виконання операцій при обліку та контролі навчальних занять.

3.1.1 Вимоги до структури та функціонування системи

Вимогами, які висуваються до структури та функціонування комунікаційної інфраструктури комп'ютеризованої системи обліку та контролю навчальних занять належать:

5. Захищеність:

– наявність засобів для обмеженого фізичного доступу до комутаційного обладнання;

– наявність коробів та інших засобів для обмеження доступу до каналів зв'язку;

– програмна захищеність активних компонентів комп'ютерної мережі;

6. Надійність:

– безвідмовність функціонування мережі не менше, ніж 20000 год.;

– резервування каналів живлення активного та пасивного апаратного забезпечення;

– наявність засобів захисту від коливання напруги;

– визначена стабільність функціонування апаратних пристроїв, що входять до складу комп'ютерної мережі;

– відповідність вимогам щодо надійності каналів передачі даних;

– наявність механізмів авторизації користувачів;

7. Продуктивність:

– швидкість обміну пакетами на рівні 100 Мб/с;

– максимальний розмір оброблюваних даних – 250000 МБ;

8. Відновлюваність та резервування:

– здатність до відновлення при виникненні аварійних ситуацій до 12 год.;

– наявність резервних каналів передачі даних;

Структура комп'ютерної мережі включає в себе:

– структуру, що забезпечує зв'язок між комп'ютерами в межах окремих класів;

– структуру функціонування системи відносно мережі Internet;

– організацію зв'язку між дирекцією та іншими одиницями, що забезпечують навчальний процес.

До функціональних вимог та структури прикладного програмного забезпечення комп'ютеризованої системи належать:

5. Функціональність:

– формування довідників про типи навчальних занять і назви предметів;
 – створення довідників студентів, учителів і класних кімнат (приміщень);
 – можливість введення, редагування та оновлення інформації про проведені заняття;

– здатність відображати успішність учнів/студентів;
 – наявність засобів для впорядкування даних за відповідними полями бази даних;

– наявність механізмів управління авторизованим доступом;
 – можливість створення звітів успішності здобувачів освіти;
 – облік навантаження викладачів/учителів;

6. Продуктивність:

– час реакції на дії користувача до 1 с;
 – час відповіді системи до 1 с;
 – оптимальне використання ресурсів RAM;
 – оптимальність використання ресурсів жорсткого диску;
 – прозорість та доступність використання функціональності програмного забезпечення;

– налаштування політики прав доступу до системи обліку та контролю навчальних занять;

– гнучкість та масштабованість структури програмного забезпечення комп'ютеризованої системи;

7. Надійність:

– завершеність функціональної та нефункціональної складової програмного забезпечення;

- стійкість роботи при виникненні нештатних ситуацій;
- можливість відновлення та резервного збереження даних;
- наявність додаткових механізмів захисту інформації;

8. Зручність використання:

- використання однотипних елементів управління і оформлення користувацького інтерфейсу;
- документованість підтримки програмного забезпечення;
- забезпечення зрозумілості функціонального призначення елементів керування;
- відповідність результатів очікуванням користувачів;
- простота і зручність використання.

В загальному випадку, структура комп'ютеризованої системи обліку та контролю навчальних занять повинна відображати організаційний зв'язок закладу освіти, процеси моніторингу та обліку проведених навчальних занять.

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

Зв'язок між компонентами комп'ютеризованої системи організовується за допомогою передачі даних по кабельній інфраструктурі комп'ютерної мережі. Вимоги до організації комп'ютерної мережі та її компонентів описано у відповідних стандартах ДСТУ, а спосіб їх використання не повинен суперечити вимогам з техніки безпеки та експлуатації. Налаштування параметрів мережі здійснюється на рівні операційних систем та програмного забезпечення активного комутаційного обладнання.

3.1.3 Вимоги по діагностуванню системи

Діагностика комп'ютеризованої системи обліку та контролю навчальних занять відбувається згідно регламенту та відповідного розкладу, і передбачає перевірку параметрів працездатності компонентів локальної комп'ютерної мережі та прикладного програмного забезпечення.

3.1.4 Перспективи розвитку, модернізація системи

Перспективами розвитку комп'ютеризованої системи обліку та контролю навчальних занять є масштабування комп'ютерної мережі шляхом додавання більшої кількості робочих станцій та інтеграції програмного забезпечення з іншими системами підтримки та автоматизації навчального процесу у закладах освіти.

Модернізація системи обліку та контролю навчальних занять можлива у випадку міграції desktop програмного забезпечення на web-орієнтований програмний додаток, що передбачає повне відтворення схеми бази даних і самих даних, або безпосередньо використання того ж самого СКБД. При модернізації повинна бути збережена логіка виконання операцій керування процесами обліку та контролю.

3.1.5 Вимоги до надійності системи

Надійність комп'ютеризованої системи контролю та обліку навчальних занять в контексті захищеності повинна забезпечуватись засобами і механізмами на рівні операційних систем сервера, робочих станцій та комутаційного обладнання комп'ютерної мережі. Доступ до бази даних повинен використовувати функціональність авторизаційних механізмів системи керування базами даних.

Система повинна бути захищена від фізичних чи механічних пошкоджень на рівні апаратного забезпечення, шляхом обмеження доступу до коробів (кабельних), комутаційних розеток, комутаторів, комутаційних шаф і т.д.

3.1.6 Вимоги до функцій та задач, які виконує система

Основними вимогами і функціями комп'ютеризованої системи обліку та контролю навчальних занять є:

- забезпечення комунікаційної інфраструктури при взаємодії клієнтського програмного забезпечення з системою керування базами даних;
- адекватна реакція на дії користувачів;
- можливість забезпечення функціональності, що відображає суть процесів обліку та контролю за проведеними заняттями;

- визначена продуктивність виконання операцій;
- контрольований та авторизований доступ до інформаційних ресурсів;
- зручність використання та обслуговування компонентів комп'ютеризованої системи;
- наявність засобів резервного формування копій бази даних.

3.1.7 Вимоги до апаратного забезпечення

Апаратне забезпечення комп'ютеризованої системи контролю та обліку навчальних занять включає наступні компоненти:

4. Активне комутаційне обладнання:

- D-Link DES-1016P – 4 шт.
- D-Link DES 1005 P – 2 шт.
- D-Link DGS-1210-10P/F – 1 шт.

5. Сервер баз даних:

- процесор з тактовою частотою не менше 2,1 ГГц з 8 ядрами та 16 потоками;
- розмір оперативної пам'яті – не менше 32 ГБ;
- об'єм жорсткого диску - не менше 8 Тб.

6. Робочі станції:

- процесор з тактовою частотою не менше 3,2 ГГц;
- об'єм оперативної пам'яті - не менше 8 ГБ;
- об'єм жорсткого диску – не менше 200 Гб.

3.1.8 Вимоги до програмного забезпечення

Операційна система сервера бази даних повинна підтримувати систему керування базами даних MS SQL Server та забезпечувати зручність адміністрування. Можливе застосування як ОС сімейства Windows Server, так і Unix-подібних.

Операційна система робочих станцій рекомендована до використання – Windows 10.

4 Вимоги до документації

Документація повинна відповідати вимогам ЄСКД та ДСТУ

Комплект документації повинен складатись з:

- пояснювальної записки;
- графічного матеріалу:
- 1. Фізична топологія комп'ютерної мережі (1 поверх).
- 2. Фізична топологія комп'ютерної мережі (2 поверх).
- 3. Діаграма прецедентів при проектуванні комп'ютеризованої системи.
- 4. Архітектура програмного забезпечення комп'ютеризованої системи.
- 5. ER-діаграма бази даних.
- 6. Алгоритм роботи програмного забезпечення комп'ютеризованої 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 та аналіз вимог технічного завдання	10.02-21.02.2021
2	Аналіз організаційної структури закладу освіти	21.02-07.03.2021
№ етапу	Назва етапу виконання кваліфікаційної роботи	Термін виконання
3	Проектування мережевої інфраструктури комп'ютеризованої системи	08.03-20.03.2021
4	Проектування схеми бази даних та архітектури програмного забезпечення	20.03-26.03.2021
5	Реалізація програмного забезпечення комп'ютерної системи	27.03-10.04.2021
6	Розробка інструкцій з налаштування параметрів комп'ютеризованої системи контролю та обліку навчальних занять	04.05-20.05.2021

Продовження таблиці 1

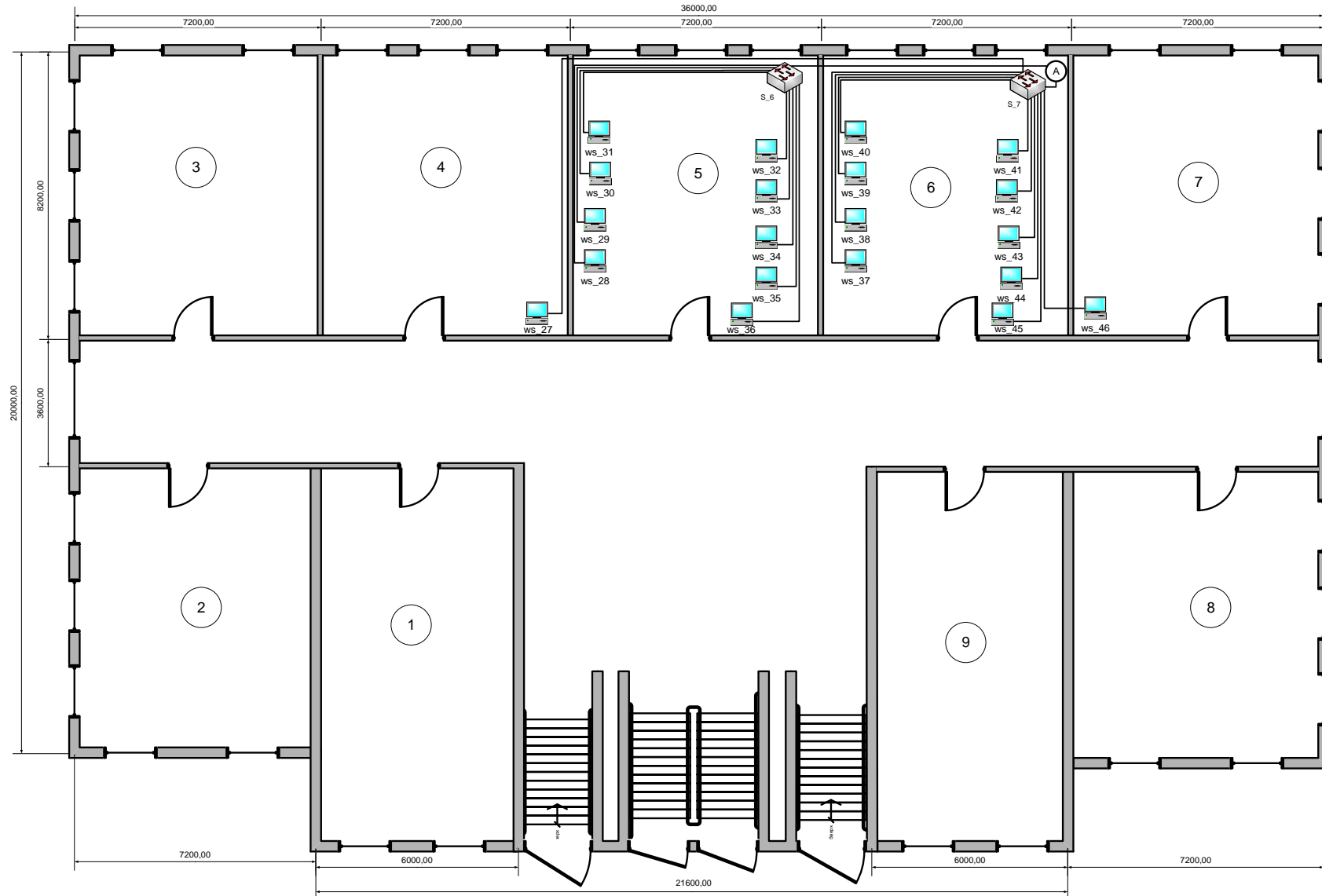
№ етапу	Назва етапу виконання кваліфікаційної роботи	Термін виконання
7	Безпека життєдіяльності, основи охорони праці	20.05-27.05.2021
8	Оформлення кваліфікаційної роботи	27.05-10.06.2021
9	Попередній захист кваліфікаційної роботи	10.06-20.06.2021
10	Захист кваліфікаційної роботи	21.06-27.06.2021

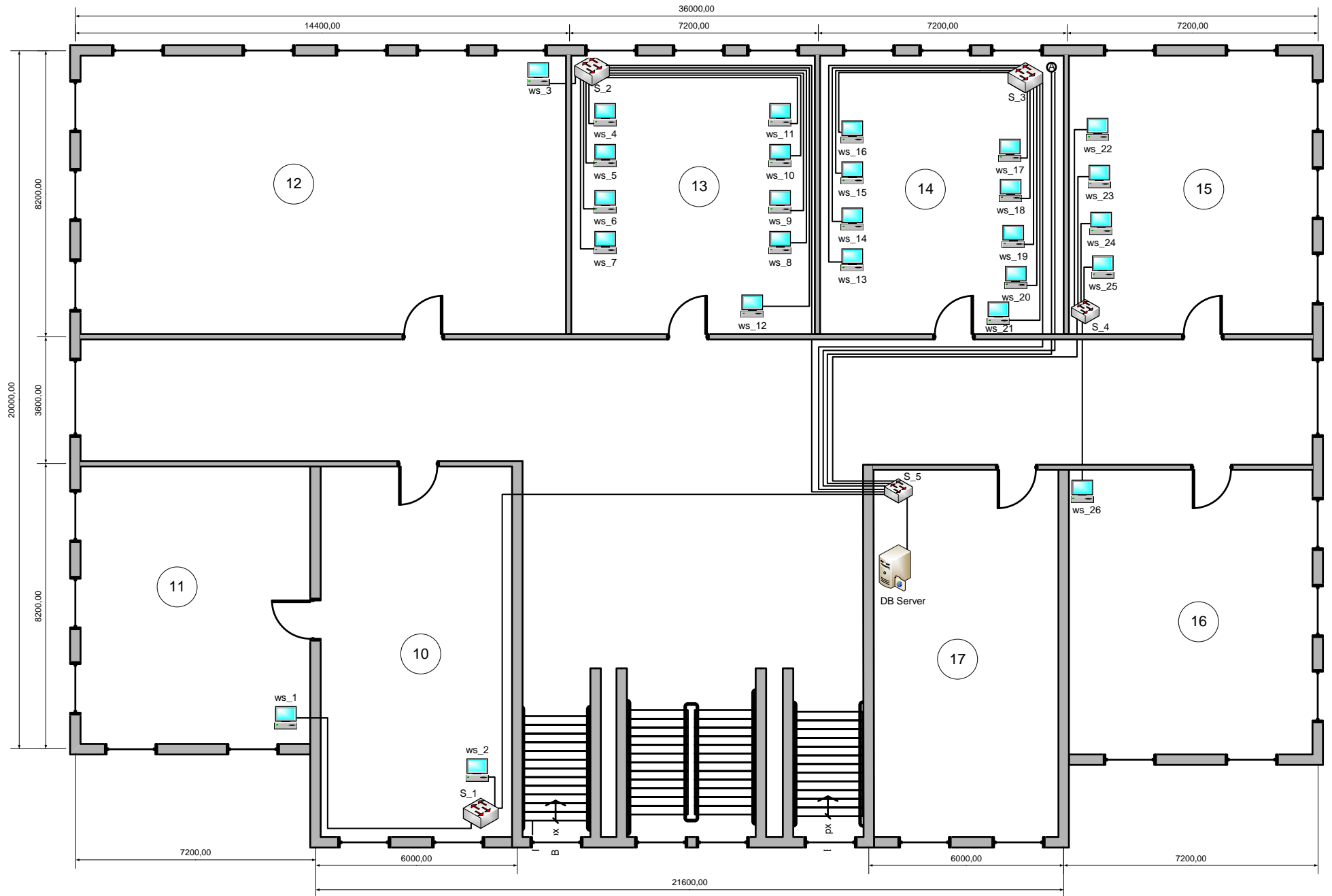
6 Додаткові умови виконання кваліфікаційної роботи

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

Додаток Б.

Фізична топологія комп'ютерної мережі закладу освіти





Додаток В.

Лістинг програмного додатку

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Windows.Forms;

namespace TeacherJournal
{
    static class Program
    {
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Form1());
        }
    }
}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace TeacherJournal
{
    public partial class GroupAdmin : Form
    {
        public GroupAdmin()
        {
            InitializeComponent();
        }

        private void GroupAdmin_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the
            'teacherJournalDS.Groups' table. You can move, or remove it,
            as needed.

            this.groupsTableAdapter.Fill(this.teacherJournalDS.Groups);
        }

        private void saveToolStripButton_Click(object sender,
        EventArgs e)
        {

```

```

        groupsBindingSource.EndEdit();
        groupsTableAdapter.Update(teacherJournalDS);
        teacherJournalDS.AcceptChanges();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        this.Close();
    }
}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace TeacherJournal
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            teachers = new Dictionary<int, string>();
            FillData();
            btNew.Enabled = false;
            btAdd.Enabled = false;
            btEdit.Enabled = false;
            btDelete.Enabled = false;
            tcData.Enabled = false;
        }

        private Dictionary<int, string> teachers;

        private Dictionary<int, string> subjects;

        private Dictionary<int, string> groups;

        private Dictionary<int, String> students;

        public void FillData()
        {
            teachers=DataLayer.GetTeachers();
            this.cbTeachers.DataSource = teachers.Values.ToList();
            subjects =
DataLayer.GetSubjects(teachers.Keys.First());
            cbDiscipline.DataSource = subjects.Values.ToList();
            groups = DataLayer.GetGroups(subjects.Keys.First());
            cbGroup.DataSource = groups.Values.ToList(); ;
        }
    }
}

```

```

private void comboBox2_SelectedIndexChanged(object sender,
EventArgs e)
{
    groups =
DataLayer.GetGroups (subjects.Where (item=>item.Value.Equals (cb
Discipline.SelectedValue)).First ().Key);
    if (groups.Count == 0)
    {
        groups.Add(0, "нет груп");
        btSelect.Enabled = false;
    }
    else
    {
        btSelect.Enabled = true;
    }
    cbGroup.DataSource = groups.Values.ToList(); ;
}

private void cbTeachers_SelectedIndexChanged(object
sender, EventArgs e)
{
    subjects = DataLayer.GetSubjects (teachers.Where (item
=> item.Value.Equals (cbTeachers.SelectedValue)).First ().Key);
    if (groups != null)
    {
        groups.Clear();
        groups.Add(0, "немає груп");
    }
    else
    {
        groups = new Dictionary<int, string>();
        groups.Add(0, "немає груп");
    }
    cbDiscipline.DataSource = subjects.Values.ToList();
}

private DataTable Journal;

public void AddLesson(string Kind, DateTime Date)
{
    if (Journal != null)
    {
        DataColumn newLesson = new
DataColumn(Kind.ToString() + "\n" + Date.ToShortDateString());
        newLesson.DataType = typeof(string);
        newLesson.DefaultValue = "";
        if
(Journal.Columns.Contains(newLesson.ColumnName))
        {
            MessageBox.Show("Таке заняття вже існує!",
"Увага! Дублювання записів", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
            return;
        }
    }
}

```

```

        Journal.Columns.Add(newLesson);
    }
}

public List<string> GetLessons()
{
    List<string> result = new List<string>();
    if (Journal != null)
    {
        foreach (DataColumn col in Journal.Columns)
        {
            if (col.ColumnName != "StudentName")
            {
                result.Add(col.ColumnName);
            }
        }
    }
    return result;
}

public void DeleteLesson(string name)
{
    if (Journal != null)
    {
        Journal.Columns.Remove(name);
    }
}

private DataTable Plans;

private void RefreshThemes()
{
    Themes = new Dictionary<int, string[]>();
    string[] ColNames = new string[Journal.Columns.Count -
1];
    for (int i = 1; i < Journal.Columns.Count; i++)
    {
        ColNames[i - 1] = Journal.Columns[i].ColumnName;
    }
    Themes = DataLayer.GetThemes(GetCurrentDisID(),
GetCurrentTeacherID(), GetCurrentGroup(), ColNames);
    Plans = new DataTable();

    DataColumn datac = new DataColumn("Занятие");
    Plans.Columns.Add(datac);
    DataColumn datac2 = new DataColumn("Тема");
    Plans.Columns.Add(datac2);
    foreach (string[] row in Themes.Values.ToList())
    {
        Plans.Rows.Add(row[0], row[1]);
    }
    dgvPlans.DataSource = Plans;
    dgvPlans.Columns[1].Width = dgvPlans.Width -
dgvPlans.Columns[0].Width - 50;
    dgvPlans.Columns[0].ReadOnly = true;
}

```



```

}

private void btSelect_Click(object sender, EventArgs e)
{
    students = DataLayer.GetStudents(groups.Where(item =>
item.Value.Equals(cbGroup.SelectedValue)).First().Key);
    Journal = new DataTable("Journal");
    Journal.Columns.Add(new DataColumn("StudentName"));
    Journal.Columns[0].Caption = "ФИО";
    Journal.Columns[0].DataType=typeof(string);
    Journal.Columns[0].ReadOnly=true;
    cbTeachers.Enabled = false;
    cbDiscipline.Enabled = false;
    cbGroup.Enabled = false;
    btSelect.Enabled = false;
    btNew.Enabled = true;
    btAdd.Enabled = true;
    btEdit.Enabled = true;
    tcData.Enabled = true;
    btDelete.Enabled = true;
    List<string> lessons =
DataLayer.GetLessons(GetCurrentTeacherID(), GetCurrentDisID(),
GetCurrentGroup());
    foreach (string lesson in lessons)
    {
        DataColumn dc = new DataColumn(lesson);
        dc.DataType = typeof(string);
        Journal.Columns.Add(dc);
    }

    foreach (String FIO in students.Values)
    {
        List<string> marks =
DataLayer.GetMarks(GetCurrentTeacherID(), GetCurrentDisID(),
students.Where(item => item.Value.Equals(FIO)).First().Key);
        object[] dr = new object[marks.Count + 1];
        dr[0] = FIO;
        for (int i = 0; i < marks.Count; i++)
        {
            dr[i + 1] = marks[i];
        }
        Journal.Rows.Add(dr);
    }
    dgvJournal.DataSource = Journal;
    dgvJournal.Columns[0].HeaderText = "ФИО";
    RefreshThemes();
}

private Dictionary<int, string[]> Themes;

private int GetCurrentTeacherID()
{
    return teachers.Where(item =>
item.Value.Equals(cbTeachers.SelectedValue.ToString())).First(
).Key;
}

```

```

    }

    private int GetCurrentDisID()
    {
        return subjects.Where(item =>
item.Value.Equals(cbDiscipline.SelectedValue.ToString())).First().Key;
    }

    private int GetCurrentGroup()
    {
        return groups.Where(item =>
item.Value.Equals(cbGroup.SelectedValue.ToString())).First().Key;
    }

    private void btAdd_Click(object sender, EventArgs e)
    {
        AddLesson al = new AddLesson(this);
        al.ShowDialog();
    }

    private void btEdit_Click(object sender, EventArgs e)
    {
        DeleteLesson dl = new DeleteLesson(this);
        dl.ShowDialog();
    }

    private void btDelete_Click(object sender, EventArgs e)
    {
        EditLesson el = new EditLesson(this);
        el.ShowDialog();
    }

    private void dgvJournal_ColumnAdded(object sender,
DataGridViewColumnEventArgs e)
    {
        for(int i=0;i<dgvJournal.Columns.Count;i++)
        {
            dgvJournal.Columns[i].SortMode =
DataGridViewColumnSortMode.NotSortable;
        }
        RefreshThemes();
    }

    private void btNew_Click(object sender, EventArgs e)
    {
        cbTeachers.Enabled = true;
        cbDiscipline.Enabled = true;
        cbGroup.Enabled = true;
        btSelect.Enabled = true;
        btNew.Enabled = false;
        btAdd.Enabled = false;
        btEdit.Enabled = false;
        btDelete.Enabled = false;
    }

```

```

        tcData.Enabled = false;
    }

    private int CalculateTime()
    {
        return dgvPlans.Rows.Count * 2;
    }

    private int GetStudentID(string FIO)
    {
        return students.Where(item =>
item.Value.Equals(FIO)).First().Key;
    }

    private void tcData_SelectedIndexChanged(object sender,
EventArgs e)
    {
        lbTime.Text = CalculateTime().ToString();
        RefreshThemes();
    }

    public void EditLesson(string OldLesson, string NewLesson)
    {
        if (!Journal.Columns.Contains(NewLesson))
        {
            Journal.Columns[OldLesson].ColumnName = NewLesson;
        }
        else
        {
            MessageBox.Show("Таке заняття вже існує!");
        }
    }

    private void btSend_Click(object sender, EventArgs e)
    {
        List<string> less=new
List<string>(Journal.Columns.Count-1);
        for (int i=1;i<Journal.Columns.Count;i++)
        {
            less.Add(Journal.Columns[i].ColumnName);
        }
        Dictionary<int,string[]> mrk=new
Dictionary<int,string[]>();
        for (int i=0;i<Journal.Rows.Count;i++)
        {
            string[] temp=new string[Journal.Columns.Count-1];
            for (int j=1;j<Journal.Columns.Count;j++)
            {
                temp[j-
1]=Journal.Rows[i].ItemArray[j].ToString();
            }

mrk.Add(GetStudentID(Journal.Rows[i].ItemArray[0].ToString()),
temp);

```

```

    }
    string[] Themes = new string[dgvPlans.Rows.Count];
    for (int i=0;i<Themes.Length;i++)
    {
        Themes[i] =
dgvPlans.Rows[i].Cells["Тема"].Value.ToString();
    }
    if (DataLayer.SaveChanges(GetCurrentTeacherID(),
GetCurrentDisID(), GetCurrentGroup(), less, mrk, Themes))
    {
        MessageBox.Show("Дані успішно збережені");
    }
}

private void dgvJournal_CellValidating(object sender,
DataGridViewCellValidatingEventArgs e)
{
    dgvJournal.Rows[e.RowIndex].ErrorText = "";
    int newInteger;
    if (e.ColumnIndex > 0 &&
dgvJournal.Rows[e.RowIndex].Cells[e.ColumnIndex].Value!="")
    {
        if (!int.TryParse(e.FormattedValue.ToString(),
out newInteger) || newInteger < 0)
        {
            e.Cancel = true;
            dgvJournal.Rows[e.RowIndex].ErrorText =
"Оцінка повинна бути невід'ємним цілим числом!";
        }
    }
}

private void btAdminTeachers_Click(object sender,
EventArgs e)
{
    TeacherAdmin tAdmin = new TeacherAdmin();
    tAdmin.ShowDialog();
    FillData();
}

private void button1_Click(object sender, EventArgs e)
{
    AdminDisciplines dAdmin = new AdminDisciplines();
    dAdmin.ShowDialog();
    FillData();
}

private void button3_Click(object sender, EventArgs e)
{
    SelectAdminDictionary sDictionaryAdmin = new
SelectAdminDictionary();
    sDictionaryAdmin.ShowDialog();
    FillData();
}

```

```

        private void button2_Click(object sender, EventArgs e)
        {
            GroupAdmin GAdmin = new GroupAdmin();
            GAdmin.ShowDialog();
            FillData();
        }
    }
}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace TeacherJournal
{
    public partial class EditLEsson : Form
    {
        public EditLEsson(Form1 parental)
        {
            InitializeComponent();
            cbKind.DataSource =
            DataLayer.GetLessonsKind().Values.ToList();
            this.parental = parental;
            this.cbLessons.DataSource = parental.GetLessons();
        }

        private Form1 parental;

        private void cbLessons_SelectedIndexChanged(object sender,
            EventArgs e)
        {
            this.cbKind.SelectedIndex =
            cbKind.Items.IndexOf((cbLessons.SelectedValue.ToString()).Split(
            ('\n'))[0]);
            string
            time=(cbLessons.SelectedValue.ToString()).Split('\n')[1];
            int day=1;
            Int32.TryParse(time.Split('.')[0], out day);
            int month= 1;
            Int32.TryParse(time.Split('.')[1], out month);
            int year = 1;
            Int32.TryParse(time.Split('.')[2], out year);
            this.dtpDate.Value = new DateTime(year,month,day);
        }

        private void button2_Click(object sender, EventArgs e)
        {
            parental.EditLesson(cbLessons.SelectedValue,

```

```

        cbKind.SelectedValue.ToString() + "\n" +
       .dtpDate.Value.ToShortDateString());
        this.Close();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        this.Close();
    }
}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace TeacherJournal
{
    public partial class DeleteLesson : Form
    {
        public DeleteLesson(Form1 Parental)
        {
            InitializeComponent();
            this.Parental = Parental;
            cbLessons.DataSource = Parental.GetLessons();
        }

        private Form1 Parental;

        private void button2_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void cbLessons_SelectedIndexChanged(object sender,
        EventArgs e)
        {
        }

        private void button1_Click(object sender, EventArgs e)
        {
            Parental.DeleteLesson(cbLessons.SelectedValue.ToString());
            this.Close();
        }
    }
}

using System;

```

```

using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data.SqlClient;
using System.Data;

namespace TeacherJournal
{
    class DataLayer
    {
        public static DataSet FlipDataSet(DataSet my_DataSet)
        {
            DataSet ds = new DataSet();
            foreach (DataTable dt in my_DataSet.Tables)
            {
                DataTable table = new DataTable();
                for (int i = 0; i <= dt.Rows.Count; i++)
                {
                    table.Columns.Add(Convert.ToString(i));
                }
                DataRow r=null;
                for (int k = 0; k < dt.Columns.Count; k++)
                {
                    r = table.NewRow();
                    r[0] = dt.Columns[k].ToString();
                    for (int j = 1; j <= dt.Rows.Count; j++)
                        r[j] = dt.Rows[j - 1][k];
                }
                table.Rows.Add(r);
                ds.Tables.Add(table);
            }
            return ds;
        }

        public static string ConnectionString = @"Data
Source=NOTEBOOK\OURFORTRESS;Initial
Catalog=TeacherJournal;Persist Security Info=True;User
ID=ghost;Password=xtimber2x";

        public static Dictionary<int,string> GetTeachers()
        {
            SqlConnection conn=new
            SqlConnection(ConnectionString);
            Dictionary<int, string> teachers = new Dictionary<int,
            string>();
            using (conn)
            {
                try
                {
                    SqlCommand comm = new SqlCommand("SELECT *
FROM Teachers", conn);
                    comm.CommandType = CommandType.Text;
                    conn.Open();
                    SqlDataReader reader = comm.ExecuteReader();
                    if (reader != null)

```

```

        {
            while (reader.Read())
            {
                int Id = reader.GetInt32(0);
                string FIO = reader.GetString(1);
                teachers.Add(Id, FIO);
            }
        }
    }
    catch
    {
        teachers.Clear();
        teachers.Add(0, "Немає викладачів");
    }
    finally
    {
        conn.Close();
    }
}
return teachers;
}

public static Dictionary<int, string> GetSubjects(int
TeacherID)
{
    SqlConnection conn = new
SqlConnection(ConnectionString);
    Dictionary<int, string> subjects = new
Dictionary<int, string>();
    using (conn)
    {
        try
        {
            SqlCommand comm = new SqlCommand(@"SELECT
Disciplines.id_discipline, Disciplines.discipline,
TeacherDiscipline.Id FROM Teachers LEFT JOIN
TeacherDiscipline ON
Teachers.id_teacher=TeacherDiscipline.Id_Teacher LEFT JOIN
Disciplines ON
TeacherDiscipline.Id_Discipline=Disciplines.id_discipline
WHERE Teachers.id_teacher=" + TeacherID.ToString(), conn);
            comm.CommandType = CommandType.Text;
            conn.Open();
            SqlDataReader reader = comm.ExecuteReader();
            if (reader != null)
            {
                while (reader.Read())
                {
                    int Id = reader.GetInt32(2);
                    string Subject = reader.GetString(1);
                    subjects.Add(Id, Subject);
                }
            }
        }
    }
    catch (Exception e)

```



```

        {
            subjects.Add(0, "нет предметов");
        }
        finally
        {
            conn.Close();
        }
    }
    return subjects;
}

public static Dictionary<int, string> GetGroups(int
TeachDisId)
{
    SqlConnection conn = new
SqlConnection(ConnectionString);
    Dictionary<int, string> groups = new Dictionary<int,
string>();
    using (conn)
    {
        try
        {
            SqlCommand comm = new SqlCommand(@"SELECT
Groups.id_group, Groups.name
FROM TeacherDiscipline
LEFT JOIN GroupDisTeach
ON TeacherDiscipline.Id=GroupDisTeach.IdTeachDis
LEFT JOIN Groups
ON Groups.id_group=GroupDisTeach.IdGroup WHERE
TeacherDiscipline.Id="+TeachDisId.ToString(), conn);
            comm.CommandType = CommandType.Text;
            conn.Open();
            SqlDataReader reader = comm.ExecuteReader();
            if (reader != null)
            {
                while (reader.Read())
                {
                    int Id = reader.GetInt32(0);
                    string Name = reader.GetString(1);
                    groups.Add(Id, Name);
                }
            }
        }
        catch
        {
        }
        finally
        {
            conn.Close();
        }
    }
    return groups;
}

```

```

    public static Dictionary<int, string> GetStudents(int
GroupId)
    {
        SqlConnection conn = new
SqlConnection(ConnectionString);
        Dictionary<int, string> students = new Dictionary<int,
string>();
        using (conn)
        {
            try
            {
                SqlCommand comm = new SqlCommand(@"SELECT
Students.id_students, Students.FI FROM Students WHERE
Students.id_group=" + GroupId.ToString(), conn);
                conn.Open();
                SqlDataReader reader = comm.ExecuteReader();
                if (reader != null)
                {
                    while (reader.Read())
                    {
                        int Id = reader.GetInt32(0);
                        string FI = reader.GetString(1);
                        students.Add(Id, FI);
                    }
                }
            }
            catch
            {
            }
            finally
            {
                conn.Close();
            }
        }
        return students;
    }

```

```

    public static Dictionary<int, string> GetLessonsKind()
    {
        SqlConnection conn = new
SqlConnection(ConnectionString);
        Dictionary<int, string> kinds = new Dictionary<int,
string>();
        using (conn)
        {
            try
            {
                SqlCommand comm = new SqlCommand(@"SELECT *
FROM Vid_lessons", conn);
                conn.Open();
                SqlDataReader reader = comm.ExecuteReader();
                if (reader != null)
                {
                    while (reader.Read())
                    {

```

```

        int Id = reader.GetInt32(0);
        string Description =
reader.GetString(1);
        kinds.Add(Id, Description);
    }
}
}
catch
{
}
finally
{
    conn.Close();
}
}
return kinds;
}

```

```

    public static List<string> GetMarks(int TeacherID, int
DisciplineID, int StudentID)
    {
        SqlConnection conn = new
SqlConnection(ConnectionString);
        List<string> marks= new List<string>();
        using (conn)
        {
            try
            {
                SqlCommand comm = new
                SqlCommand(String.Format(@"SELECT Bally.ball
FROM Lessons
LEFT JOIN Teachers
ON Lessons.id_teacher=Teachers.id_teacher
LEFT JOIN Disciplines
ON Disciplines.id_discipline=Lessons.id_discipline
LEFT JOIN Vid_lessons
ON Vid_lessons.id_vid_lesson=Lessons.id_vid_lesson
LEFT JOIN SuperBalls
ON SuperBalls.IdLesson=Lessons.id_lesson
LEFT JOIN Bally
ON Bally.id_super=SuperBalls.Id
WHERE Bally.id_students={0} AND Disciplines.id_discipline={1} AND
Teachers.id_teacher={2}", StudentID, DisciplineID, TeacherID),
conn);

                conn.Open();
                SqlDataReader reader = comm.ExecuteReader();
                if (reader != null)
                {
                    while (reader.Read())
                    {
                        string Ball =
reader.GetInt32(0).ToString();
                        if (Ball == "0")
                        {
                            Ball = "";

```



```

        themes.Add(id, new string[] {
lessons[i].Replace('\n', ' '), Theme });
        count++;
    }
    if (count == 0)
    {
        themes.Add(-1, new string[] {
lessons[i].Replace('\n', ' '), "" });
    }
    reader.Close();
    }
    }
    catch (Exception e)
    {
    }
    finally
    {
        conn.Close();
    }
}
return themes;
}

public static void SaveTheme(int Id, string Theme)
{
    SqlConnection conn = new
SqlConnection(ConnectionString);
    using (conn)
    {
        try
        {
            SqlCommand comm = new
SqlCommand(String.Format(@"UPDATE Lessons SET
Lessons.tema='{0}' WHERE id_lesson={1}", Theme, Id), conn);
            conn.Open();
            comm.ExecuteNonQuery();
        }
        catch (Exception e)
        {
        }
        finally
        {
            conn.Close();
        }
    }
}

public static List<string> GetLessons(int TeacherID, int
DisciplineID, int GroupID)
{
    SqlConnection conn = new
SqlConnection(ConnectionString);
    List<string> lessons = new List<string>();

```

```

        using (conn)
        {
            try
            {
                SqlCommand comm = new
                SqlCommand(String.Format(@"SELECT Vid_lessons.vid_lesson,
                Lessons.date
FROM Lessons
LEFT JOIN Teachers
ON Lessons.id_teacher=Teachers.id_teacher
LEFT JOIN Disciplines
ON Disciplines.id_discipline=Lessons.id_discipline
LEFT JOIN Vid_lessons
ON Vid_lessons.id_vid_lesson=Lessons.id_vid_lesson
WHERE Teachers.id_teacher={0} AND Disciplines.id_discipline={1}
AND Lessons.id_group={2}", TeacherID, DisciplineID, GroupID),
                conn);

                conn.Open();
                SqlDataReader reader = comm.ExecuteReader();
                if (reader != null)
                {
                    while (reader.Read())
                    {
                        string Kind = reader.GetString(0);
                        string Date =
                reader.GetDateTime(1).ToShortDateString();
                        lessons.Add(Kind + "\n" + Date);
                    }
                }
            }
            catch
            {
            }
            finally
            {
                conn.Close();
            }
        }
        return lessons;
    }

    private static int GetLessonSuper(int LessonID)
    {
        SqlConnection conn = new
        SqlConnection(ConnectionString);
        int result=-1;
        using (conn)
        {
            try
            {
                SqlCommand comm = new
                SqlCommand(String.Format(@"SELECT SuperBalls.Id FROM
                SuperBalls WHERE SuperBalls.IdLesson={0}", LessonID), conn);
                conn.Open();
            }
        }
    }

```

```

        SqlDataReader reader = comm.ExecuteReader();
        if (reader != null)
        {
            while (reader.Read())
            {
                result = reader.GetInt32(0);
            }
        }
    }
    catch
    {
    }
    finally
    {
        conn.Close();
    }
}
return result;
}

private static void ClearLesson(int Id)
{
    SqlConnection conn = new
SqlConnection(ConnectionString);
    using (conn)
    {
        try
        {
            SqlCommand comm = new
SqlCommand(String.Format(@"DELETE Bally WHERE id_super={0}
DELETE SuperBalls WHERE IdLesson={1} DELETE Lessons WHERE
Lessons.id_lesson={1}", GetLessonSuper(Id), Id), conn);
            conn.Open();
            comm.ExecuteNonQuery();
        }
        catch
        {
        }
        finally
        {
            conn.Close();
        }
    }
}

private static int CreateLesson(int TeacherID, int
DisciplineID, int GroupID, string Lesson)
{
    SqlConnection conn = new
SqlConnection(ConnectionString);
    string Kind=Lesson.Split('\n')[0];
    string Date=Lesson.Split('\n')[1];
    string[] elems = Date.Split('.');
    int result = 0;
}

```

```

        int
        KindID=DataLayer.GetLessonsKind().Where(item=>item.Value.Equals(Kind)).First().Key;
        using (conn)
        {
            try
            {
                SqlCommand comm = new
                SqlCommand(String.Format(@"INSERT INTO Lessons
                (id_teacher,id_discipline,id_vid_lesson,id_group,date,tema)
                VALUES({0},{1},{2},{3},'{4}-{5}-{6}','{7}') SELECT
                MAX(Lessons.id_lesson) FROM Lessons", TeacherID, DisciplineID,
                KindID, GroupID, elems[2], elems[1], elems[0], ""), conn);
                conn.Open();
                SqlDataReader reader = comm.ExecuteReader();
                if (reader != null)
                {
                    while (reader.Read())
                    {
                        result = reader.GetInt32(0);
                    }
                }
            }
            catch
            {
            }
            finally
            {
                conn.Close();
            }
            return result;
        }
    }

    private static List<int> GetLessonsId(int TeacherID, int
    DisciplineID, int GroupID)
    {
        SqlConnection conn = new
        SqlConnection(ConnectionString);
        List<int> lessons = new List<int>();
        using (conn)
        {
            try
            {
                SqlCommand comm = new
                SqlCommand(String.Format(@"SELECT Lessons.id_lesson
                FROM Lessons
                LEFT JOIN Teachers
                ON Lessons.id_teacher=Teachers.id_teacher
                LEFT JOIN Disciplines
                ON Disciplines.id_discipline=Lessons.id_discipline
                WHERE Teachers.id_teacher={0} AND Disciplines.id_discipline={1}
                AND Lessons.id_group={2}", TeacherID, DisciplineID, GroupID),
                conn);

                conn.Open();
            }
        }
    }

```



```

        SqlDataReader reader = comm.ExecuteReader();
        if (reader != null)
        {
            while (reader.Read())
            {
                lessons.Add(reader.GetInt32(0));
            }
        }
        catch
        {
        }
        finally
        {
            conn.Close();
        }
    }
    return lessons;
}

private static int ExistBalls(int LessonId)
{
    SqlConnection conn = new
SqlConnection(ConnectionString);
    int id=-1;
    using (conn)
    {
        try
        {
            SqlCommand comm = new
SqlCommand(String.Format(@"SELECT SuperBalls.Id FROM
SuperBalls WHERE SuperBalls.IdLesson={0}", LessonId), conn);
            conn.Open();
            SqlDataReader reader = comm.ExecuteReader();
            if (reader != null)
            {
                while (reader.Read())
                {
                    id = reader.GetInt32(0);
                }
            }
        }
        catch
        {
        }
        finally
        {
            conn.Close();
        }
    }
    return id;
}

private static void UpdateBalls(int SuperID,
Dictionary<int, string[]> Marks,int counter)

```

```

    {
        SqlConnection conn = new
SqlConnection(ConnectionString);
        using (conn)
        {
            try
            {
                conn.Open();
                foreach (int key in Marks.Keys)
                {
                    SqlCommand comm = new
SqlConnection(String.Format(@"UPDATE Bally
SET ball={0}
WHERE id_students={1} AND id_super={2}",
Marks[key][counter]==""?"0":Marks[key][counter],key,SuperID),
conn);

                    comm.ExecuteNonQuery();
                }
            }
            catch
            {
            }
            finally
            {
                conn.Close();
            }
        }
    }

    private static void InsertBalls(int SuperID,
Dictionary<int, string[]> Marks,int counter)
    {
        SqlConnection conn = new
SqlConnection(ConnectionString);
        using (conn)
        {
            try
            {
                conn.Open();
                foreach (int key in Marks.Keys)
                {
                    SqlCommand comm = new
SqlConnection(String.Format(@"INSERT INTO Bally
(id_students,id_super,ball)
VALUES({0},{1},{2})", key, SuperID,
Marks[key][counter]==""?"0":Marks[key][counter]), conn);

                    comm.ExecuteNonQuery();
                }
            }
            catch
            {
            }
            finally
            {
                conn.Close();
            }
        }
    }

```

```

    }
}

private static bool CreateBalls(int LessonID,
Dictionary<int,string[]> Marks, int counter)
{
    int superId = ExistBalls(LessonID);
    SqlConnection conn = new
SqlConnection(ConnectionString);
    if (superId == -1)
    {
        using (conn)
        {
            try
            {
                SqlCommand comm = new
SqlCommand(String.Format(@"INSERT INTO SuperBalls (IdLesson)
VALUES ({0}) SELECT MAX(Id) FROM SuperBalls", LessonID), conn);
                conn.Open();
                SqlDataReader reader =
comm.ExecuteReader();
                if (reader != null)
                {
                    while (reader.Read())
                    {
                        superId = reader.GetInt32(0);
                    }
                }
            }
            catch
            {
            }
            finally
            {
                conn.Close();
            }
        }
        InsertBalls(superId, Marks, counter);
    }
    else
    {
        UpdateBalls(superId, Marks, counter);
    }

    return true;
}

public static bool SaveChanges(int TeacherID, int
DisciplineID, int GroupID, List<string> Lessons,
Dictionary<int, string[]> Marks, string[] Themes)

```

```

    {
        List<int> lessonsId = new List<int>();
        List<string> OldLessons = GetLessons(TeacherID,
DisciplineID, GroupID);
        lessonsId=GetLessonsId(TeacherID, DisciplineID,
GroupID);
        for (int i=0;i<OldLessons.Count;i++)
        {
            if (!Lessons.Contains(OldLessons[i]))
            {
                ClearLesson(lessonsId[i]);
                lessonsId[i]=-1;
            }
        }
        OldLessons = GetLessons(TeacherID, DisciplineID,
GroupID);
        lessonsId = GetLessonsId(TeacherID, DisciplineID,
GroupID);
        for (int i = 0; i < Lessons.Count;i++ )
        {
            if (!OldLessons.Contains(Lessons[i]))
            {
                int lessId = (CreateLesson(TeacherID,
DisciplineID, GroupID,Lessons[i]));
                CreateBalls(lessId, Marks, i);
                SaveTheme(lessId, Themes[i]);
            }
            else
            {
                CreateBalls(lessonsId[OldLessons.IndexOf(Lessons[i])], Marks,
i);

                SaveTheme(lessonsId[OldLessons.IndexOf(Lessons[i])],
Themes[i]);
            }
        }
        return true;
    }
}

}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace TeacherJournal

```

```

{
    public partial class AdminDisciplines : Form
    {
        public AdminDisciplines()
        {
            InitializeComponent();
        }

        private void AdminDisciplines_Load(object sender,
        EventArgs e)
        {
            // TODO: This line of code loads data into the
            'teacherJournalDS.Disciplines' table. You can move, or remove
            it, as needed.

            this.disciplinesTableAdapter.Fill(this.teacherJournalDS.Discip
            lines);

        }

        private void saveToolStripButton_Click(object sender,
        EventArgs e)
        {
            try
            {
                disciplinesBindingSource.EndEdit();
                disciplinesTableAdapter.Update(teacherJournalDS);
                teacherJournalDS.AcceptChanges();
                MessageBox.Show("Вдало збережено", "Результат
                операції", MessageBoxButtons.OK, MessageBoxIcon.Information);
            }
            catch
            {
                MessageBox.Show("Помилка при збереженні",
                "Результат операції", MessageBoxButtons.OK,
                MessageBoxIcon.Error);
            }
        }

        private void button1_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

```

```
namespace TeacherJournal
{
    public partial class SelectAdminDictionary : Form
    {
        public SelectAdminDictionary()
        {
            InitializeComponent();
        }

        private void btEdit_Click(object sender, EventArgs e)
        {
            switch (cbDictionaryName.SelectedIndex)
            {
                case 0:
                {
                    VidLessonsAdmin vAdmin = new
VidLessonsAdmin();
                    vAdmin.ShowDialog();
                }
                break;
                case 1:
                {
                    TeacherDisciplineAdmin tda = new
TeacherDisciplineAdmin();
                    tda.ShowDialog();
                }
                break;
                default:
                break;
            }
        }
    }
}
```

Додаток Г.

Скрипт формування бази даних

```
CREATE TABLE [Bally] (  
    [id_bally] [int] IDENTITY (0, 1) NOT NULL ,  
    [id_students] [int] NOT NULL ,  
    [ball] [int] NOT NULL ,  
    [id_super] int NOT NULL FOREIGN KEY REFERENCES SuperBalls(Id)  
    CONSTRAINT [PK_Bally] PRIMARY KEY CLUSTERED  
    (  
        [id_bally]  
    ) ON [PRIMARY] ,  
    CONSTRAINT [FK_Bally_Students] FOREIGN KEY  
    (  
        [id_students]  
    ) REFERENCES [Students] (  
        [id_students]  
    ) ON DELETE CASCADE ON UPDATE CASCADE  
    ) ON [PRIMARY]  
GO  
  
CREATE TABLE [Disciplines] (  
    [id_discipline] [int] IDENTITY (0, 1) NOT NULL ,  
    [discipline] [varchar] (50) COLLATE Cyrillic_General_CI_AS NOT  
    NULL ,  
    CONSTRAINT [PK_Disciplines] PRIMARY KEY CLUSTERED  
    (  
        [id_discipline]  
    ) ON [PRIMARY]  
    ) ON [PRIMARY]  
GO  
  
CREATE TABLE [Groups] (  
    [id_group] [int] IDENTITY (0, 1) NOT NULL ,  
    [name] [char] (10) COLLATE Cyrillic_General_CI_AS NOT NULL ,  
    CONSTRAINT [PK_Groups] PRIMARY KEY CLUSTERED  
    (  
        [id_group]  
    ) ON [PRIMARY]  
    ) ON [PRIMARY]  
GO  
  
CREATE TABLE [Lessons] (  
    [id_lesson] [int] IDENTITY (0, 1) NOT NULL ,  
    [id_teacher] [int] NOT NULL ,  
    [id_discipline] [int] NOT NULL ,  
    [id_vid_lesson] [int] NOT NULL ,  
    [id_group] [int] NOT NULL FOREIGN KEY REFERENCES  
    Groups(id_group),  
    [date] [datetime] NOT NULL ,  
    [tema] [varchar] (50) COLLATE Cyrillic_General_CI_AS NOT NULL  
    ,  
    CONSTRAINT [PK_Lessons] PRIMARY KEY CLUSTERED
```

```

(
    [id_lesson]
) ON [PRIMARY] ,
CONSTRAINT [FK_Lessons_Disciplines] FOREIGN KEY
(
    [id_discipline]
) REFERENCES [Disciplines] (
    [id_discipline]
) ON DELETE CASCADE ON UPDATE CASCADE ,
CONSTRAINT [FK_Lessons_Teachers] FOREIGN KEY
(
    [id_teacher]
) REFERENCES [Teachers] (
    [id_teacher]
) ON DELETE CASCADE ON UPDATE CASCADE ,
CONSTRAINT [FK_Lessons_Vid_lessons] FOREIGN KEY
(
    [id_vid_lesson]
) REFERENCES [Vid_lessons] (
    [id_vid_lesson]
) ON DELETE CASCADE ON UPDATE CASCADE
) ON [PRIMARY]
GO

```

```

CREATE TABLE [Students] (
    [id_students] [int] IDENTITY (0, 1) NOT NULL ,
    [id_group] [int] NOT NULL ,
    [FI] [varchar] (50) COLLATE Cyrillic_General_CI_AS NOT NULL ,
    CONSTRAINT [PK_Students] PRIMARY KEY CLUSTERED
(
    [id_students]
) ON [PRIMARY] ,
CONSTRAINT [FK_Students_Groups] FOREIGN KEY
(
    [id_group]
) REFERENCES [Groups] (
    [id_group]
) ON DELETE CASCADE ON UPDATE CASCADE
) ON [PRIMARY]
GO

```

```

CREATE TABLE [Teachers] (
    [id_teacher] [int] IDENTITY (0, 1) NOT NULL ,
    [FIO] [varchar] (50) COLLATE Cyrillic_General_CI_AS NOT NULL ,
    CONSTRAINT [PK_Teachers] PRIMARY KEY CLUSTERED
(
    [id_teacher]
) ON [PRIMARY]
) ON [PRIMARY]
GO

```

```

CREATE TABLE [Vid_lessons] (
    [id_vid_lesson] [int] IDENTITY (0, 1) NOT NULL ,

```



```
[vid_lesson] [varchar] (50) COLLATE Cyrillic_General_CI_AS NOT  
NULL ,  
CONSTRAINT [PK_Vid_lessons] PRIMARY KEY CLUSTERED  
(  
    [id_vid_lesson]  
) ON [PRIMARY]  
) ON [PRIMARY]  
GO
```