

НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ  
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
Кафедра інженерії програмного забезпечення

## ПОЯСНЮВАЛЬНА ЗАПИСКА

до бакалаврської роботи

на ступінь вищої освіти бакалавр

на тему: «**РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ  
ОБЛІКУ ОБЛАДНАННЯ ЛАБОРАТОРІЙ КАФЕДР  
ЗАСОБАМИ МОВИ C#**»

Виконав: студент 5 курсу, групи ППЗ-51

спеціальності

121 Інженерія програмного забезпечення

(шифр і назва спеціальності)

Сидоренко В.О.

(прізвище та ініціали)

Керівник Золотухіна О.А.

(прізвище та ініціали)

Рецензент \_\_\_\_\_

(прізвище та ініціали)

Нормоконтроль \_\_\_\_\_

(прізвище та ініціали)

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ  
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ  
ТЕХНОЛОГІЙ**

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти - «Бакалавр»

Спеціальність - 121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного  
забезпечення

О.В. Негоденко

“ \_\_\_\_\_ ” \_\_\_\_\_ 2021 року

**З А В Д А Н Н Я  
НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ**

Сидоренко Владиславу Олександровичу

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка програмного забезпечення для обліку обладнання лабораторій кафедр засобами мови C#»

Керівник роботи Золотухіна Оксана Анатоліївна, к.т.н., доц., доцент каф. ІІЗ,  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від “16” лютого 2022 року №22.

2. Строк подання студентом роботи 01.06.2022.

3. Вихідні дані до роботи:

3.1. Положення для побудови реляційної бази даних;

3.2. Методи побудови баз даних для обліку обладнання;

3.3. Існуючі інструменти обліку обладнання лабораторій;

3.4. Науково-технічна література;

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

4.1. Аналіз існуючих методів обліку обладнання лабораторій  
кафедр

4.2. Розробка бази даних для обліку обладнання

- 4.3. Програмна реалізація додатку
- 4.4. Приклади використання системи
- 4.5. Висновки

5. Перелік графічного матеріалу.

- 5.1. Титульний слайд
- 5.2. Мета, об'єкт та предмет дослідження
- 5.3. Актуальність роботи
- 5.4. Аналоги
- 5.5. Порівняння з аналогами
- 5.6. Технічне завдання
- 5.7. Програмні засоби реалізації
- 5.8. Структура реляційної БД
- 5.9. Приклади програмної реалізації
- 5.10. Апробація результатів дослідження
- 5.11. Висновки

6. Дата видачі завдання: 19.04.2022

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	19.04.2022	Виконано
2	Дослідження існуючих інструментів для обліку обладнання лабораторій	20.04.2022	Виконано
3	Проектування схеми побудови бази даних	21.04.2022 – 30.04.2022	Виконано
4	Розробка програмного забезпечення для бакалаврської роботи	01.05.2022 – 17.05.2022	Виконано
5	Висновки, оформлення роботи	17.05.2022 – 20.05.2022	Виконано
6	Розробка демонстраційних матеріалів	20.05.2022	Виконано
7	Попередній захист роботи	26.05.2022	
8	Здача роботи	01.06.2022	

Студент \_\_\_\_\_ Сидоренко В.О.  
( підпис ) ( прізвище та ініціали )

Керівник роботи \_\_\_\_\_ Золотухіна О.А.  
( підпис ) ( прізвище та ініціали )





## РЕФЕРАТ

Текстова частина бакалаврської роботи 63 с., 37 рис., 5 таблиць, 20 джерел.

Мета роботи - спростити облік обладнання в навчальних лабораторіях кафедр Державного університету телекомунікацій за рахунок впровадження програмного забезпечення

Об'єкт дослідження - процес обліку обладнання в навчальних лабораторіях кафедр Державного університету телекомунікацій

Предмет дослідження - програмні засоби для обліку обладнання в навчальних лабораторіях

Результатом роботи став програмний продукт, який оптимізував ведення обліку паспортів лабораторій, його обладнання та програмного забезпечення.

Головні компоненти, які реалізовані у програмному забезпеченні:

- Інтерфейс системи забезпечує ключові операції обліку обладнання навчальних лабораторій кафедр в Державному університеті телекомунікацій.
- База даних системи містить інформацію про обладнання та програмне забезпечення, яке знаходиться в лабораторіях, та інші дані відповідно до паспортів кабінетів.
- Програма забезпечує можливість вибірки необхідної інформації з бази даних паспортів кабінетів, в тому числі, експорт даних в Microsoft Word.
- Програма надає можливість зміни інформації щодо неактуальних даних, їх видалення та перенесення.

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ОБЛІК ОБЛАДНАННЯ, VISUAL STUDIO, C#, SQL, PHPMYADMIN

## ЗМІСТ

<b>РЕФЕРАТ</b> .....	<b>6</b>
<b>ВСТУП</b> .....	<b>9</b>
<b>1.АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПРОГРАМНИХ ПРОДУКТІВ</b> .....	<b>10</b>
1.1 Огляд існуючих засобів та програм для обліку обладнання .....	10
1.2 Використання клієнт-серверної архітектури.....	16
1.3 Постановка задач дипломної роботи.....	18
<b>2. ОПИС ПРОЕКТУВАННЯ СИСТЕМИ</b> .....	<b>19</b>
2.1 Структура додатку .....	20
2.2 Діаграма варіантів використання додатку .....	21
2.3 Засоби розробки .....	22
2.3.1 Microsoft Visual Studio .....	22
2.3.2 Windows Forms .....	24
2.3.3 Microsoft SQL Server .....	27
2.3.4 phpMyAdmin .....	29
2.3.5 using MySQLConnector .....	31
<b>3. ПРОГРАМНА РЕАЛІЗАЦІЯ</b> .....	<b>32</b>
3.1 Створення БД в phpMyAdmin .....	32
3.2 Підключення БД до проекту VisualStudio .....	40
3.3 Робота програмного забезпечення з відображенням бази даних	40
3.4 Тестування програмного забезпечення.....	48
<b>ВИСНОВКИ</b> .....	<b>52</b>
<b>ПЕРЕЛІК ПОСИЛАНЬ</b> .....	<b>53</b>
<b>ДОДАТОК 1</b> .....	<b>55</b>
Лістинг програми .....	55
<b>ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ</b> .....	<b>59</b>

## ВСТУП

На сьогоднішній день питання щодо автоматизації тих чи інших процесів перед працівниками лабораторій постає все частіше. А саме: автоматизація обліку обладнання, програмного забезпечення, інформації щодо наявних в лабораторіях меблів, їх інвентарних номерів, аудиторій в яких вони знаходяться. Все це повинно бути відображено максимально простою та продуманою структурою, для отримання актуальної інформації щодо паспортів лабораторій та економії часу на заповнення документів власноруч.

Основою для ведення обліку обладнання лабораторій є вже готові паспорти аудиторій, які обов'язково мають бути внесені в систему. Вони подаються у вигляді паперових або Word документів.

Головною ціллю розробки програмного забезпечення для обліку обладнання навчальних лабораторій кафедр є програмний продукт, який оптимізує ведення обліку обладнання, а також збереже час співробітників лабораторії, завдяки автоматизації комплексу задач.



## 1.АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПРОГРАМНИХ ПРОДУКТІВ

### 1.1 Огляд існуючих засобів та програм для обліку обладнання

Засоби для обліку обладнання навчальних лабораторій кафедр можуть бути описані наступним переліком :

- паперові засоби;
- документи Word;

Засоби паперового обліку включають різноманітні документи, які затверджуються та мають особливий характер ведення.

Звичайно паперовий облік обладнання є застарілою моделлю , тим паче для освітнього закладу обладнаним такою кількістю сучасного обладнання. Але списувати такий варіант все одно не слід, тому що у нього є ряд переваг:

- паперовий варіант є звичним для старшого покоління і не потребує додаткової адаптації та навчання;
- використання паперового блокноту не потребує залучення сторонніх спеціалістів чи обов'язкового застосування програмних засобів.

До недоліків використання паперового документів можна віднести наступне:

- при внесенні даних можна допустити помилки в розрахунках;
- зміна даних потребує закреслення або витирання;
- робочий поле для внесення даних обмежено площиною;
- дані в документі не захищені від стороннього доступу;
- формування статистичних даних за паперовими даними доволі ускладнене.

Документи Word є більш зручним способом організації обліку та контролю актуальної інформації.

Саме в цьому форматі зберігається та вноситься інформація щодо паспортів аудиторій в Державному Університеті Телекомунікацій(Рис 1.1). Інформація для заповнення бази даних в програмному забезпеченні використовувалась з наданих

Word документів. Дивлячись на структуру та спосіб ведення документації можна виділити такі недоліки:

- дані в таблиці доводиться вводити в ручну, що може призвести до пропуску позицій та багато витраченого часу для їх заповнення;
- засоби захисту даних в Word обмежені, тому не можуть виключити сторонній доступ до них.
- створення якісних шаблонів документів вимагає певного досвіду від користувача;

**Паспорт кабінету**  
(лабораторії, аудиторії, викладацької, деканат, та інше.)

Найменування : лабораторія  
(лабораторії, аудиторії, викладацької, деканат, та інше.)

Кому належить: кафедра Інженерії програмного забезпечення ННІТ  
(найменування кафедри, інституту)

Місце розташування кабінету: поверх 3, № кабінету 310  
(поверх, № кабінету)

Загальна площа кабінету (лабораторії, аудиторії, викладацької, деканат, та інше.) 47.9 м<sup>2</sup>.

Кількість вікон 3 шт.

Меблі та обладнання:

<p>столи <u>25</u> шт.,</p> <p>робочий стіл викладача <u>1</u> шт.,</p> <p>дошка <u>1</u> шт.,</p> <p>екран <u>1</u> шт.,</p> <p>комп'ютер <u>25</u> шт.,</p> <p>шафа <u>1</u> шт.,</p> <p>стілці для студентів <u>32</u> шт.</p>	<p>магнітна дошка <u>1</u> шт.,</p> <p>проектор <u>1</u> шт.,</p> <p>жалюзі <u>3</u> шт.,</p> <p>вогнегасники <u>2</u> шт.,</p> <p>плафони освітлення <u>15</u> шт.,</p> <p>розетки електричні <u>48</u> шт.,</p> <p>розетки Інтернет( і що стосується Інтернету) <u>1</u> шт.,</p>
---	---

стілці викладача, інженера лабораторії 1 шт.,  
кондиціонер 1 шт.;

Орієнтовний перелік навчально-наочних посібників і навчального обладнання:

№	Назва	Кількість і місце знаходження	Інвентарний №
1	<u>Відеопроєктор</u>	<u>Acer X 112 H (MR/JKV11.001)</u>	1
2	Дошка	магнітно-маркерна ALU23	1
3	Жалюзі вертикальні	<u>Line 1310</u>	3
4	Колонки	<u>Microlab 2.0 FC30 Whle</u>	1
5	Комп'ютер	<u>HPt5630w Compaq 2GF/1GR MS</u>	25
6	Кондиціонер	<u>підлогово-потолочного типу CH-F24NK2/CH-U24NK2</u>	1

Рис. 1.1 – Паспорт аудиторії представлений в Microsoft Word

Microsoft Excel.

Зручний і популярний інструмент для ведення обліку. Має інтуїтивний інтерфейс, але потребує певних навичок роботи. Підходить для ведення обліку не багатьох таблиць.

Недоліки:

- Не підходить для комплексного внесення даних.
- Не оптимізований вивід інформації для користувача, при великій кількості даних.

<b>A.Ritikov</b>					
	3	Intel Celeron 3066	MB Asus P4S 800MX/ 128 Mb DDR / HDD 40,0Gb Samsung 7200 rpm/ FDD 1,44 /Video GeForce4 MX440 64Mb / sound / Case ATX 300W / keyboard	Кей	10 340.00 21.12.2011
	17	LCD Acer AL1511	діагональ 15"	Кей	7 090.00 16.12.2011
					17 430.00
<b>admin</b>					
	1	Intel Celeron 2800	MB Asus P4S 800MX/ 128 Mb DDR / HDD 40,0Gb Samsung 7200 rpm/ FDD 1,44 /Video GeForce4 MX440 64Mb / sound / Case ATX 300W / keyboard	Кей	8 235.50 28.12.2011
	2	Intel Celeron 2933	MB Asus P4S 800MX/ 128 Mb DDR / HDD 40,0Gb Samsung 7200 rpm/ FDD 1,44 /Video GeForce4 MX440 64Mb / sound / Case ATX 300W / keyboard	Рин	9 669.50 21.12.2011
	14	LCD Acer AL1511	діагональ 15"	Рин	7 090.00 16.12.2011
	26	Epson LaserJet 1020	A4, 14с/н, 1200x600dpi, ОЗУ 2 Mb,USB	Кей	4 950.00 10.02.2011
	28	Nokia 6100	GSM900/1800/1900, 76г, виворывоз, полифония, цветной дисплей, WAP, GPRS	Кей	1 000.00 07.09.2011
					30 945.00
<b>D.Ershov</b>					
	7	AMD Sempron 2500	MB Gigabyte GA-K8NS s.754/ 128 Mb DDR / HDD 40,0Gb Samsung 7200rpm/ FDD 1,44 / Video GeForce4 MX440 64Mb / sound / LAN/ Case ATX 300W / keyboard	Кей	8 693.00 21.12.2011
	16	LCD Acer AL1511	діагональ 15"	Кей	7 090.00 16.12.2011
	29	Nokia 6100	GSM900/1800/1900, 76г, виворывоз, полифония, цветной дисплей, WAP, GPRS	Кей	1 000.00 07.09.2011
					16 783.00

Рис.1.2 Приклад обліку в Microsoft Excel

Програма Eqman.

Програма створена для обліку будівельних інструментів та електроніки. Має комплексну вибірку даних, інтеграцію в 1С.

Переваги:

- Точна кількість активів – фіксація кожної одиниці оргтехніки та витратних матеріалів, що надійшли у розпорядження.

- Відсутність крадіжок – кожна дрібниця закріплена за працівниками, несанкціоновані списання та розкрадання зводяться до нуля.

- Ведення бази фізичного обладнання та придбаного ПЗ.

- Позначки про дати техобслуговування – своєчасно віддавайте оргтехніку на ремонт та модернізацію

Недоліки:

- Тяжкий для освоєння інтерфейс

- Обмежена кількість даних для внесення.

- Проблеми з локалізацією, половина меню на українській, а інша половина на англійській мовах.

The screenshot displays the Eqrman application interface. At the top, there are four summary cards:

- Общая стоимость** (Total cost): 65117, with a note "Учитывая количественные ТМЦ" (Considering quantitative TMC).
- Всего ТМЦ** (Total TMC): 5, with a note "Не учитывая количественные ТМЦ" (Not considering quantitative TMC).
- В сервисе** (In service): 0 ТМЦ, with a note "В данный момент в сервисе" (At the moment in service).
- Списано** (Written off): 0 ТМЦ, with a note "В данный момент списаны" (At the moment written off).

Below the cards is a search bar and several action buttons: "CREATE ITEM", "UPLOAD ITEMS FILE", "INTEGRATION IC", and "DOWNLOAD REPORT".

The main part of the interface is a table with the following columns: QR Code, Tool type, brand\_model, Location, Status, Responsible, Last Change, Quantity, and Price. The table contains five rows of equipment data:

QR Code	Tool type	brand_model	Location	Status	Responsible	Last Change	Quantity	Price
-	Бур (сверло) Расходные материалы	Makita D-00131	Завод Цех 2	Worker	Vlad	26 мая	1 шт	44.00 UAH
a0001	Перфоратор Электроинструмент	Makita HR2470	Завод Цех 2	Worker	Vlad	26 мая	1 шт	4099.00 UAH
a0002	Бур (сверло) Расходные материалы	Makita D-00131	-	Worker	Vlad	26 мая	4 шт	176.00 UAH
a0003	М1 Компьютер	Apple Macbook	-	Worker	Vlad	26 мая	1	59399.00 UAH
-	Болгарка Электроинструмент	Bosch GWS 750-125	Завод Цех 1	Not used	-	26 мая	1 шт	1399.00 UAH

At the bottom right of the table, there is a pagination control showing "Row per page" and "1 - 5 of 5".

Рис.1.3 Приклад обліку в додатку Eqrman

## IT Invent.

Створена для ведення інвентаризаційного обліку комп'ютерів, оргтехніки, IT-обладнання, програмного забезпечення, комплектуючих та витратних матеріалів. Облік ведеться у межах організацій та його філій.

### Ключові особливості IT Invent:

- Підтримка бази даних Microsoft Access та Microsoft SQL Server;
- Розрахований на багато користувачів режим роботи – всі філії працюють з єдиною базою;

- можливість створення власних додаткових властивостей різних типів;
- Облік замовлень постачальникам на всі види облікових одиниць;
- Логічне зв'язування програм та комплектуючих з обладнанням;
- Закріплення конфігураційних одиниць за співробітниками організації;
- Гнучке розмежування прав доступу користувачів системи;

### Недоліки:

- Заплутаний та старомодний інтерфейс
- Безкоштовна версія лише до 200 записів.

The screenshot displays the IT Invent 1.64 Free software interface. The main window is titled 'IT Invent 1.64 Free' and features a menu bar with options like 'Файл', 'Оборудование', 'Программы', 'Комплектующие', 'Расходники', 'Инвентарь', 'Сеть', 'Документы', 'Задачи', 'Расписания', 'Отчеты', 'Админ', and 'Помощь'. Below the menu is a toolbar with various icons. The main area is divided into several panels, each displaying a table of data:

- ОБОРУДОВАНИЕ**: Table with columns for 'Всего' (80), 'Активно (не списано)' (70), and various status/location filters.
- ПРОГРАММЫ**: Table with columns for 'Всего' (80), 'Активно (не списано)' (80), and various license/status filters.
- КОМПЛЕКТУЮЩИЕ**: Table with columns for 'Всего' (80), 'Активно (не списано)' (74), and various status/location filters.
- РАСХОДНИКИ**: Table with columns for 'Всего' (40), 'Активно (не списано)' (40), and various date/status filters.
- ИНВЕНТАРЬ**: Table with columns for 'Всего' (50), 'Активно (не списано)' (50), and various status/location filters.
- РЕМОНТЫ**: Table with columns for 'Всего' (25), 'Открытых Ремонтов' (10), and 'Закрытых Ремонтов' (15).
- ПЕРЕМЕЩЕНИЯ**: Table with columns for 'Всего' (1), 'Статус Создано' (0), 'Статус Отправлено' (0), and 'Статус Получено' (1).
- ЗАКАЗЫ**: Table with columns for 'Всего' (8), 'Открытых Заказов' (2), 'Частично принятых заказов' (0), 'Закрытых Заказов' (6), and 'Открытых Заказов старше 20 дней' (2).
- РАБОТЫ**: Table with columns for 'Всего' (10), 'Не Выполненных работ на сегодня' (0), 'Назначенных Работ' (2), 'Работ В Процессе Выполнения' (2), 'Выполненных Работ' (6), 'Срок истекает в течении 3 дней' (0), 'Просроченных работ старше 5 дней' (2), 'Просроченных работ старше 10 дней' (2), and 'Просроченных работ старше 30 дней' (2).
- ЗАДАЧИ**: Table with columns for 'Всего' (8), 'Созданных задач' (4), 'Назначенных задач' (1), 'Задач выполняется' (1), 'Выполненных задач' (0), 'Задач на проверке' (0), 'Закрытых задач' (2), 'Не закрытые где я автор' (6), 'Просроченные где я автор' (6), 'Не закрытые где я исполнитель' (0), and 'Просроченные где я исполнитель' (0).

The status bar at the bottom indicates: 'Тип лицензии: Free (200), Sched Free (1, 3) | Пользователь: Administrator | Роль: Администратор | База: C:\Program Files (x86)\IT Invent\ITInvent.mdb | Сегодня: Пт 16.03.2018

Рис. 1.4 Приклад обліку в IT Invent

Таблиця 1.1 - Порівняння аналогів

Назва	Microsoft Excel	Eqman	IT Invent
Наявність безкоштовної версії	+	-	+/- (до 200 облікових одиниць)
Легкість в освоєні інтерфейсу	+	-	-
Можливість формуванати специфічні звіти	+/- Формування звітів потребує глибокого знання функцій Excel та важке для звичайного користувача	- тільки шаблонні звіти	- тільки шаблонні звіти
Можливість експорту даних	+	+	+
Наявність засобів пошуку, фільтрації, вибірки	+	+	+
Наявність засобів для обмеження доступу до інформації	+/- обмежений доступ до окремих комірок/аркушів з використанням пароля	+	+

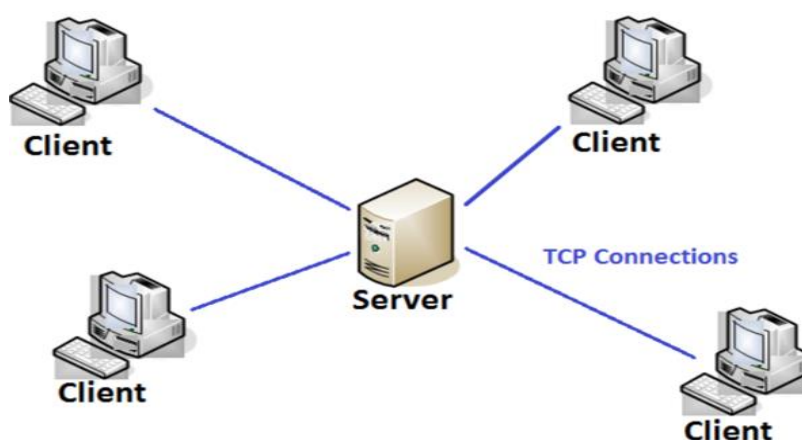
## 1.2 Використання клієнт-серверної архітектури

Архітектура клієнтського сервера – це обчислювальна модель, в якій сервер розміщує, постачає та керує більшістю ресурсів і послуг, які споживає клієнт. Цей тип архітектури має один або кілька клієнтських комп'ютерів, під'єднаних до центрального сервера через мережу або Інтернет. Ця система спільно використовує обчислювальні ресурси. Архітектура клієнт/сервер також відома як модель мережових обчислень або мережа клієнт/сервер, оскільки всі запити та послуги доставляються через мережу.

Архітектура клієнт-сервер — це архітектура комп'ютерної мережі, в якій багато клієнтів (віддалені процесори) запитують і отримують послугу від централізованого сервера (хост комп'ютера). Клієнтські комп'ютери забезпечують інтерфейс, що дозволяє користувачеві комп'ютера запитувати послуги сервера та відображати результати, які повертає сервер. Сервери чекають надходження запитів від клієнтів, а потім відповідають на них. В ідеалі сервер надає клієнтам стандартизований прозорий інтерфейс, щоб клієнтам не потрібно було знати про особливості системи (тобто апаратного та програмного забезпечення), яка надає послугу.

Клієнти часто знаходяться на робочих станціях або на персональних комп'ютерах, тоді як сервери розташовані в інших місцях мережі, як правило, на більш потужних машинах. Ця обчислювальна модель особливо ефективна, коли клієнти і сервер мають різні завдання, які вони регулярно виконують. При обробці лікарняних даних, наприклад, на клієнтському комп'ютері може бути запущена прикладна програма для введення інформації про пацієнта, тоді як на серверному комп'ютері запущена інша програма, яка керує базою даних, в якій постійно зберігається інформація. Багато клієнтів можуть отримувати доступ до інформації сервера одночасно, і в той же час клієнтський комп'ютер може виконувати інші завдання, наприклад надсилати електронну пошту. Оскільки і клієнтські, і серверні комп'ютери вважаються інтелектуальними пристроями, модель клієнт-сервер

повністю відрізняється від старої моделі «mainframe», в якій централізований мейнфрейм виконував усі завдання для пов'язаних з ним «dumb» терміналів.



Ми знаходимося в епоху, коли інформаційні технології відіграють важливу роль у бізнес-додатках, розглядаючись як область, яку організація має дуже інвестувати, щоб розширити можливості, доступні для конкуренції на світовому ринку. «Конкурентоспроможна глобальна економіка забезпечить застарілість і неясність для тих, хто не може або не хоче конкурувати» (Client/Server Architecture, 2011), згідно з цією заявою, організаціям необхідно підтримувати свої ринкові позиції шляхом реінжинірингу переважаючих організаційних структур і ділової практики, щоб досягти своїх бізнес-цілей. Коротше кажучи, це основна потреба розвиватися зі зміною технологічних аспектів. Тому організації повинні пройти механізм отримання та обробки своїх корпоративних даних, щоб зробити бізнес-процедури більш ефективними, щоб досягти успіху або вижити на світовому ринку.

Модель клієнт/сервер відкриває логічну перспективу розподіленої корпоративної обробки, де сервер обробляє та обробляє всі клієнтські запити. Це також можна розглядати як революційну віху в галузі обробки даних. «Обчислення клієнт/сервер є найефективнішим джерелом інструментів, які надають співробітникам повноваження та відповідальність». Бізнес є силами, що створюють потребу в клієнт-серверних обчисленнях». Комп'ютерні обчислення клієнт-сервер мають величезний прогрес у комп'ютерній індустрії, залишаючи недоторканими будь-яку область чи куток. Часто гібридні навички необхідні для розробки програм



клієнт-сервер, включаючи проектування баз даних, обробку транзакцій, комунікаційні навички, дизайн і розробку графічного інтерфейсу користувача тощо. Розширені програми вимагають досвіду розподілених об'єктів та компонентних інфраструктур. Найбільш поширеною сьогодні стратегією клієнт/сервер є реалізація локальної мережі ПК, оптимізована для використання груп/пакетів. Це в основному дало поріг для багатьох нових розподілених підприємств, оскільки усуває обчислення, орієнтовані на хост.

Характеристики архітектури клієнт-сервер:

- Клієнтські та серверні машини потребують різної кількості апаратних та програмних ресурсів.
- Клієнтські та серверні машини можуть належати різним постачальникам.
- Горизонтальна масштабованість (збільшення клієнтських машин) і вертикальна масштабованість (міграція на більш потужний сервер або на багатосерверне рішення)
- Клієнтська або серверна програма безпосередньо взаємодіє з протоколом транспортного рівня, щоб встановити зв'язок та надіслати чи отримати інформацію.
- Потім транспортний протокол використовує протоколи нижнього рівня для надсилання або отримання окремих повідомлень. Таким чином, комп'ютеру потрібен повний набір протоколів для роботи клієнта або сервера.
- Один комп'ютер серверного класу може надавати кілька послуг одночасно; для кожної служби потрібна окрема серверна програма.

### **1.3 Постановка задач дипломної роботи**

Мета роботи - спростити облік обладнання в навчальних лабораторіях кафедр Державного університету телекомунікацій за рахунок впровадження програмного забезпечення

Об'єкт дослідження - процес обліку обладнання в навчальних лабораторіях кафедр Державного університету телекомунікацій

Предмет дослідження - програмні засоби для обліку обладнання в навчальних лабораторіях

В кожному навчальному закладі ведеться облік обладнання лабораторій, для розуміння наповненості та призначання аудиторії.

Аналіз інформації щодо паспортів лабораторій дозволяє сформулювати вимоги до майбутнього програмного продукту. Програма для обліку обладнання в навчальних лабораторіях кафедр повинна бути реалізована для персональних комп'ютерів та забезпечувати наступні функції:

1. Створити для користувача максимально простого інтерфейсу;
2. Внесення в систему даних про актуальне обладнання та програмне забезпечення, яке знаходиться в лабораторіях;
3. Можливість сортувати всю інформацію по аудиторіям, лабораторіям, яка була взята з їх паспортів;
4. Можливість зміни інформації щодо неактуальних даних, їх видалення та перенесення(імпорту).
5. Створення оптимізованих зв'язків між таблицями даних

На основі інформації паспортів кабінетів та порядку обліку інформації про навчальні лабораторії, необхідно створити програмне забезпечення для обліку обладнання лабораторій кафедр.

Для реалізації дипломного проекту було створено базу даних за допомогою веб-додатку phpMyAdmin, який дає змогу створювати та управляти реляційними базами даних. Інформаційна система написана мовою програмування C# в середовищі програмування Visual Studio. Графічний інтерфейс розроблено з використанням інтерфейсу програмування додатків Windows Forms.

## **2. ОПИС ПРОЕКТУВАННЯ СИСТЕМИ**

Для створення БД та реалізації зв'язків було використано веб додаток phpMyAdmin та OpenServer для запуску локального сервера.

В якості середовища розробки програмного забезпечення було обрано Visual Studio, реалізація користувацького інтерфейсу створена завдяки віконному додатку Windows Forms та мові програмування C#.

Запити , вибірки та відображення бази в компоненті DataGridView реалізовано завдяки реляційній системі керуванням бази даних MySQL.

## 2.1 Структура додатку

Для реалізації поставлених задач було створено 2 бази даних. Перша пов'язана з модулем управління доступом до програми, друга містить всю інформацію щодо паспортів аудиторій(Рис 2.1).

В базі даних з паспортами аудиторій присутні три модулі:

1. Модуль формування вибірки даних
2. Модуль роботи з базою даних паспортів кабінетів
3. Модуль експорту даних

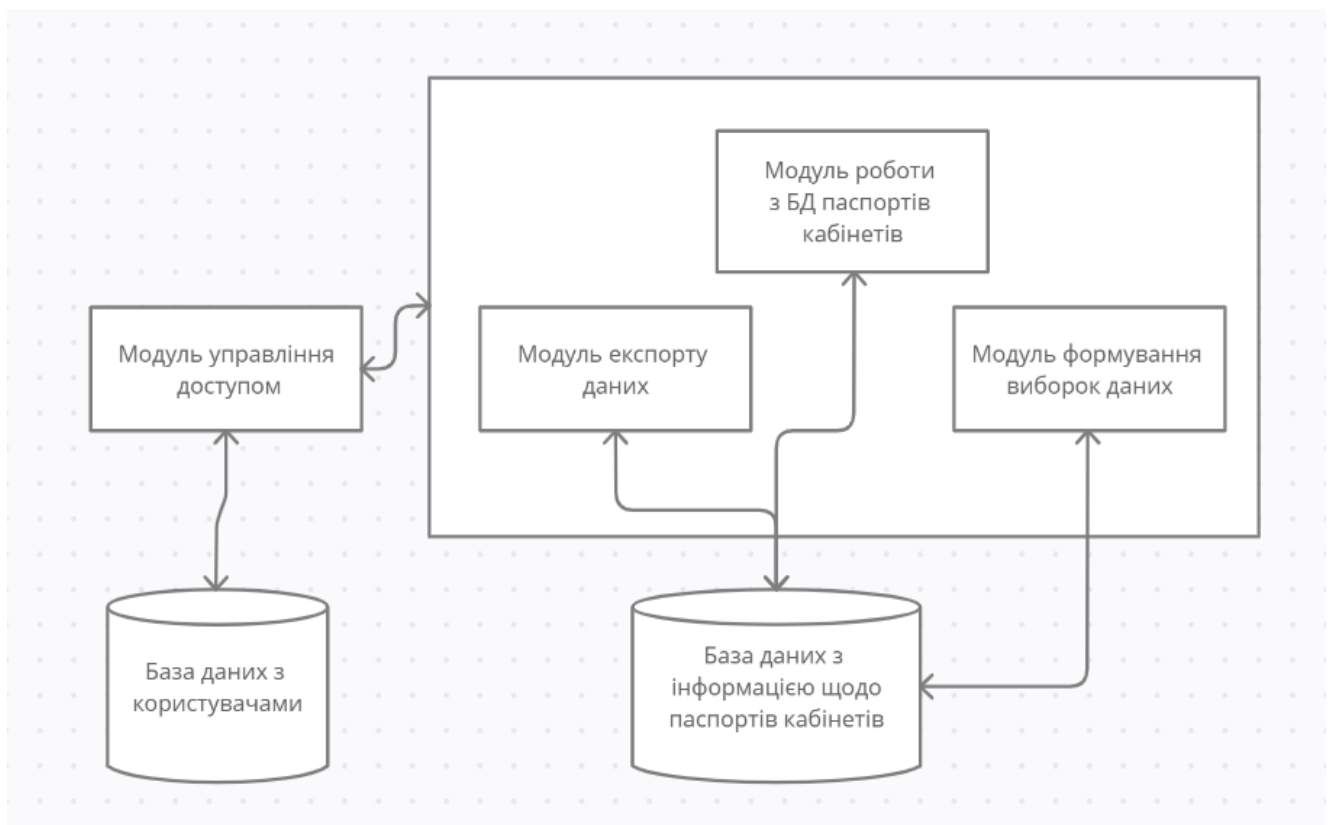


Рис 2.1 Схема структури додатку

## 2.2 Діаграма варіантів використання додатку

Діаграма варіантів використання — це динамічна або поведінкова діаграма в UML. Діаграми варіантів використання моделюють функціональність системи за допомогою акторів і варіантів використання. Варіанти використання — це набір дій, послуг і функцій, які система повинна виконувати. У цьому контексті «система» — це щось, що розробляється або функціонує, наприклад веб-сайт. «Актори» — це люди або організації, які виконують певні ролі в системі (Рис 2.2).

В системі обліку обладнання навчальних лабораторій кафедр було виділено 2 актори:

1. Користувач
2. Модуль управління доступом

Користувач повинен пройти авторизацію, яка з'єднана з модулем управління доступом, після чого для нього будуть доступні наступні функції:

- Робота з базою даних паспортів, яка включає додавання, редагування та видалення даних паспорту
- Формування вибірки з бази даних, яка включає сортування та має змогу формувати комплексну вибірку
- Експорт в Ms Word



Рис 2.2 Діаграма варіантів використання

## 2.3 Засоби розробки

### 2.3.1 Microsoft Visual Studio

Microsoft Visual Studio є інтегрованим середовищем розробки. По суті, це програма, яка дозволяє розробляти, писати та редагувати код.

Visual Studio робить більше, ніж більшість, дозволяючи використовувати практично будь-яку мову кодування. Серед інших, до них належать:

C, C++, C++/CLI, .NET, JavaScript, TypeScript, XML, XSLT, HTML, CSS, Інші, такі як Python, Ruby, Node.js і N, доступні через плагіни.

Цей майже повний список означає, що у вас є широкий спектр варіантів. Microsoft Visual Studio має на меті бути повною та вичерпною для всіх ваших потреб у кодуванні. Для цього він працює з вами на чотирьох рівнях: розробка, налагодження, тестування та спільна робота.

Visual Studio пропонує вказівки та допомогу під час кодування, незалежно від мови:

- IntelliSense описує APIS під час введення тексту та використовує автозавершення для прискорення кодування;
- Програма відстежує, де ви перебуваєте у своєму кодуванні, навіть коли ви шукаєте інші частини коду;
- Функція «Знайти всі посилання» дозволяє групувати, фільтрувати та здійснювати пошук у результатах;
- Code Lens дозволяє вам зрозуміти структуру виклику вашого коду та перейти до пов'язаних функцій, а також дасть вам знати, хто останній редагував код;
- Піктограми з лампочками дають вам знати, коли вам потрібно виправити поширену помилку кодування, навіть коли ви вводите код вперше;
- Список помилок об'єднує всі ваші проблеми з кодуванням в один простір, тому ви можете легко вирішити всі проблеми;
- Code Link може шукати потенційні рішення складних проблем.

Visual Studios виконує всю основну роботу з рефакторингу за вас у міру розвитку вашого проекту;

Налагодження:

- Налагодження працює, навіть якщо ви кодуєте кількома мовами;
- Ви можете налагоджувати незалежно від того, де працює ваш код (додатки Windows, Android, Azure, iOS тощо);
- Visual Studio пропонує вам повний контроль над налагодженням і дозволяє вибирати, де ви хочете призупинити потоки та перевірити код;
- Програма також пропонує вам гнучкість у перевірці коду, надаючи вам можливість переглядати змінні та складні вирази в будь-якому місці коду;
- Ви можете отримувати сповіщення, коли трапляються винятки;
- Програма дозволяє легко перевіряти код у кількох потоках;
- PerfTips та інструменти діагностики дозволяють дізнатися більше про продуктивність вашого коду та характеристики пам'яті;

Тестування:

- Visual Studio надає велику кількість шаблонів і фреймворків, щоб ви могли писати, виконувати та налагоджувати модульні тести;
- IntelliTest зменшує зусилля, необхідні для створення та підтримки модульних тестів;
- Модульне тестування в реальному часі дозволяє вам переконатися, що ваші зміни не порушують тести;
- Тестування користувальницького інтерфейсу дозволяє вам управляти вашим додатком через інтерфейс користувача;
- У великому масштабі ви можете протестувати свій код для сотень тисяч одночасних користувачів з усього світу;
- Ви можете бачити результати тестування навіть під час організації, виконання та налагодження тесту. Ви можете автоматично запускати тести після кожної збірки;

Колоборації:

- Visual Studio дозволяє відстежувати зміни, внесені в код, і виділяти їх;
- Інтеграція з іншими популярними IDE означає, що ви можете працювати зі своєю командою на всіх платформах;
- Параметри візуалізації допомагають відстежувати зміни у філіях;
- Детальні дозволи дозволяють контролювати доступ до коду та забезпечувати його безпеку;
- Active Directory дозволяє легко відстежувати людей і дозволи;
- Налаштована інформаційна панель дозволяє легко відстежувати вашу команду та останні зміни;
- Marketplace розширень допоможе вам знайти більше розширень для подальшої інтеграції;
- Пошук семантичного коду дозволяє швидко шукати код;

### **2.3.2 Windows Forms**

Windows Forms є платформою інтерфейсу користувача, яка створює багаті клієнтські програми для настільних комп'ютерів з операційною системою Windows. Платформа розробки Windows Forms підтримує широкий набір функцій розробки додатків, включаючи елементи керування, графіку, прив'язування даних і введення користувача. Windows Forms містить візуальний дизайнер із перетягуванням у Visual Studio для легкого створення програм Windows Forms.

Windows Forms — це структура інтерфейсу користувача для створення настільних програм Windows. Він забезпечує один з найпродуктивніших способів створення настільних програм на основі візуального дизайнера, наданого у Visual Studio. Такі функціональні можливості, як розміщення візуальних елементів керування перетягуванням, дозволяють легко створювати настільні програми.

За допомогою Windows Forms ви розробляєте графічні програми, які легко розгортати, оновлювати та працювати в автономному режимі або підключені до Інтернету. Програми Windows Forms можуть отримати доступ до локального обладнання та файлової системи комп'ютера, на якому запущена програма.

Windows Forms — це технологія інтерфейсу користувача для .NET, набору керованих бібліотек, які спрощують звичайні завдання програми, такі як читання та запис у файлову систему. Коли ви використовуєте таке середовище розробки, як Visual Studio, ви можете створювати інтелектуальні клієнтські програми Windows Forms, які відображають інформацію, запитують введення від користувачів і спілкуються з віддаленими комп'ютерами через мережу.

У Windows Forms форма — це візуальна поверхня, на якій ви відображаєте інформацію для користувача. Зазвичай ви створюєте програми Windows Forms, додаючи елементи керування до форм і розробляючи відповіді на дії користувача, наприклад клацання мишею або натискання клавіш.

Елемент керування — це дискретний елемент інтерфейсу користувача, який відображає дані або приймає введення даних.

Коли користувач щось робить з вашою формою або одним із елементів керування, ця дія створює подію. Ваш додаток реагує на ці події за допомогою коду та обробляє події, коли вони відбуваються.

Windows Forms містить різноманітні елементи керування, які можна додати до форм: елементи керування, які відображають текстові поля, кнопки, спадні вікна, перемикач і навіть веб-сторінки. Якщо наявний елемент керування не відповідає вашим потребам, Windows Forms також підтримує створення власних спеціальних елементів керування за допомогою класу `UserControl`.

Windows Forms має багаті елементи керування інтерфейсом користувача, які імітують функції високоякісних програм, таких як Microsoft Office. Коли ви використовуєте елементи керування `ToolStrip` і `MenuStrip`, ви можете створювати панелі інструментів і меню, які містять текст і зображення, відображати підменю та розміщувати інші елементи керування, такі як текстові поля та поля зі списком.

За допомогою конструктора Windows Forms з перетягуванням у Visual Studio ви можете легко створювати програми Windows Forms. Просто виберіть елементи керування за допомогою курсора та розташуйте їх у потрібному місці на формі. Конструктор надає такі інструменти, як лінії сітки та лінії прив'язки, щоб позбутися від проблем із вирівнюванням елементів керування. Ви можете використовувати



елементи керування `FlowLayoutPanel`, `TableLayoutPanel` і `SplitContainer` для створення розширених макетів форм за менший час.

Нарешті, якщо вам потрібно створити власні елементи інтерфейсу користувача, простір імен `System.Drawing` містить великий вибір класів для відтворення ліній, кіл та інших фігур безпосередньо на формі.

Багато програм повинні відображати дані з бази даних, файлів XML або JSON, веб-служби чи іншого джерела даних. `Windows Forms` надає гнучкий елемент керування, який називається елементом керування `DataGridView` для відображення таких табличних даних у традиційному форматі рядків і стовпців, так що кожна частина даних займає власну клітинку. Коли ви використовуєте `DataGridView`, ви можете налаштувати зовнішній вигляд окремих комірок, блокувати довільні рядки та стовпці на місці, а також відображати складні елементи керування всередині клітинок, серед інших функцій.

Підключення до джерел даних через мережу — це просте завдання з `Windows Forms`. Компонент `BindingSource` представляє з'єднання з джерелом даних і надає методи прив'язки даних до елементів керування, переходу до попереднього та наступного записів, редагування записів та збереження змін до вихідного джерела. Елемент керування `BindingNavigator` забезпечує простий інтерфейс через компонент `BindingSource`, щоб користувачі могли переміщатися між записами.

Ви можете легко створювати елементи керування, пов'язані з даними, використовуючи вікно «Джерела даних» у `Visual Studio`. У вікні відображаються такі джерела даних, як бази даних, веб-сервіси та об'єкти у вашому проекті. Ви можете створювати елементи керування, пов'язані з даними, перетягуючи елементи з цього вікна на форми у своєму проекті. Ви також можете зв'язати наявні елементи керування з даними, перетягуючи об'єкти з вікна джерел даних на наявні елементи керування.

Іншим типом прив'язки даних, яким ви можете керувати в `Windows Forms`, є налаштування. Більшість програм повинні зберігати деяку інформацію про стан під час виконання, наприклад останній відомий розмір форм, і зберігати дані про налаштування користувача, наприклад розташування за умовчанням для

збережених файлів. Функція «Параметри програми» відповідає цим вимогам, забезпечуючи простий спосіб збереження обох типів налаштувань на клієнтському комп'ютері. Після того, як ви визначите ці параметри за допомогою Visual Studio або редактора коду, налаштування зберігаються як XML і автоматично зчитуються назад у пам'ять під час виконання.

Після того, як ви написали програму, ви повинні надіслати її користувачам, щоб вони змогли встановити та запустити її на своїх клієнтських комп'ютерах. Коли ви використовуєте технологію ClickOnce, ви можете розгортати свої програми з Visual Studio, використовуючи лише кілька кліків, і надавати користувачам URL-адресу, яка вказує на вашу програму в Інтернеті. ClickOnce керує всіма елементами та залежностями у вашій програмі та гарантує, що програма правильно встановлена на клієнтському комп'ютері.

Додатки ClickOnce можна налаштувати на запуск, лише коли користувач підключений до мережі, або на запуск як онлайн, так і офлайн. Коли ви вказуєте, що програма має підтримувати роботу в автономному режимі, ClickOnce додає посилання на вашу програму в меню «Пуск» користувача. Після цього користувач може відкрити програму, не використовуючи URL-адресу.

Коли ви оновлюєте свою програму, ви публікуєте новий маніфест розгортання та нову копію програми на веб-сервері. ClickOnce виявить, що доступне оновлення, і оновить установку користувача. Для оновлення старих програм не потрібно спеціального програмування.

### **2.3.3 Microsoft SQL Server**

Microsoft SQL Server — це система керування реляційною базою даних (RDBMS), яка підтримує широкий спектр додатків для обробки транзакцій, бізнес-аналітики та аналітики в корпоративних IT-середовищах. Microsoft SQL Server є однією з трьох провідних на ринку технологій баз даних, поряд з Oracle Database і IBM DB2.

Як і інше програмне забезпечення RDBMS, Microsoft SQL Server побудовано на основі SQL, стандартизованої мови програмування, яку адміністратори баз даних (DBA) та інші IT-фахівці використовують для керування базами даних і запитів даних, які вони містять. SQL Server пов'язаний з Transact-SQL (T-SQL), реалізацією SQL від Microsoft, яка додає набір власних розширень програмування до стандартної мови.

Як і інші технології RDBMS, SQL Server в основному побудований на основі структури таблиць на основі рядків, яка з'єднує пов'язані елементи даних у різних таблицях один з одним, уникаючи необхідності надлишкового зберігання даних у кількох місцях у базі даних. Реляційна модель також забезпечує посиальну цілісність та інші обмеження цілісності для підтримки точності даних. Ці перевірки є частиною більш широкого дотримання принципів атомарності, узгодженості, ізоляції та довговічності, спільно відомих як властивості ACID, і призначені для гарантії надійної обробки транзакцій бази даних.

Основним компонентом Microsoft SQL Server є SQL Server Database Engine, який контролює зберігання, обробку та безпеку даних. Він включає в себе реляційний механізм, який обробляє команди та запити, і механізм зберігання, який керує файлами бази даних, таблицями, сторінками, індексами, буферами даних і транзакціями. Збережені процедури, тригери, уявлення та інші об'єкти бази даних також створюються та виконуються Database Engine.

Під Database Engine знаходиться операційна система SQL Server або SQLOS. SQLOS обробляє функції нижнього рівня, такі як управління пам'яттю та вводом-виводом, планування завдань і блокування даних, щоб уникнути конфліктних оновлень. Рівень мережевого інтерфейсу знаходиться над Database Engine і використовує протокол табличного потоку даних Microsoft для полегшення взаємодії запитів і відповідей із серверами баз даних. А на рівні користувача адміністратори баз даних та розробники SQL Server пишуть оператори T-SQL для створення та модифікації структур баз даних, керування даними, впровадження засобів захисту та резервного копіювання баз даних, серед інших завдань.

### 2.3.4 phpMyAdmin

phpMyAdmin — це програмне забезпечення з відкритим вихідним кодом, представлене 9 вересня 1998 року, написане на PHP. По суті, це сторонній інструмент для керування таблицями та даними всередині бази даних. phpMyAdmin підтримує різні типи операцій на MariaDB і MySQL. Основною метою phpMyAdmin є керування адмініструванням MySQL через Інтернет.

Це найпопулярніший додаток для керування базами даних MySQL. За допомогою цього програмного забезпечення ми можемо створювати, оновлювати, скидати, змінювати, видаляти, імпортувати та експортувати таблиці бази даних MySQL. phpMyAdmin також підтримує широкий спектр операцій, таких як керування базами даних, відносинами, таблицями, стовпцями, індексами, дозволами, користувачами тощо, на MySQL та MariaDB. Ці операції можна виконувати через користувальницький інтерфейс, при цьому ми все ще маємо можливість виконувати будь-який оператор SQL.

phpMyAdmin перекладено 72 мовами, а також підтримує мови RTL і LTR, щоб широке коло людей могло легко використовувати це програмне забезпечення. Ми можемо виконувати запити MySQL, ремонтувати, оптимізувати, перевірити таблиці, а також виконувати інші команди керування базою даних. phpMyAdmin також можна використовувати для виконання адміністративних завдань, таких як створення бази даних, виконання запитів.

phpMyAdmin — це програма на основі графічного інтерфейсу, яка використовується для керування базою даних MySQL. Ми можемо вручну створити базу даних і таблицю і виконати запит до них. Він забезпечує веб-інтерфейс і може працювати на будь-якому сервері. Оскільки він є веб-основою, ми можемо отримати до нього доступ з будь-якого комп'ютера.

phpMyAdmin підтримує кілька функцій, наведених нижче:

- phpMyAdmin може створювати, змінювати, переглядати та скидати бази даних, представлення даних, таблиці, стовпці та індекси.

- Він може відображати кілька наборів результатів за допомогою запитів і збережених процедур.
  - phpMyAdmin використовує збережені процедури та запити для відображення кількох наборів результатів.
  - Він підтримує зовнішні ключі та таблиці InnoDB.
  - phpMyAdmin може відстежувати зміни, внесені в базах даних, представленнях і таблицях.
  - Ми також можемо створити PDF-графіку макета нашої бази даних.
  - phpMyAdmin можна експортувати в різні формати, такі як XML, CSV, PDF, ISO/IEC 26300 - OpenDocument Text and Spreadsheet.
  - Він підтримує mysqli, який є покращеним розширенням MySQL.
  - phpMyAdmin може взаємодіяти з 80 різними мовами.
  - phpMyAdmin може редагувати, виконувати та створювати закладки будь-які оператори SQL і навіть пакетні запити.
  - Використовуючи набір попередньо визначених функцій, він може перетворювати збережені дані в будь-який формат. Наприклад - BLOB-дані як зображення або посилання для завантаження.
  - Він забезпечує можливість резервного копіювання бази даних у різних формах.
- Перевага phpMyAdmin:
- phpMyAdmin може працювати на будь-якому сервері або будь-якій ОС, оскільки він має веб-браузер.
  - Ми можемо легко створювати, видаляти та редагувати базу даних і можемо керувати всіма елементами за допомогою графічного інтерфейсу phpMyAdmin, який набагато простіше, ніж редактор командного рядка MySQL.
  - phpMyAdmin допомагає нам контролювати дозвіл користувача та працювати з кількома серверами одночасно.

- Ми також можемо створити резервну копію нашої бази даних та експортувати дані в різні формати, як-от XML, CSV, SQL, PDF, OpenDocument Text, Excel, Word та електронні таблиці тощо.

- Ми можемо виконувати складні оператори та запити SQL, створювати та редагувати функції, тригери та події за допомогою графічного інтерфейсу phpMyAdmin.

Різниця між phpMyAdmin і MySQL Console:

- phpMyAdmin надає інтерфейс користувача, за допомогою якого ми можемо виконувати запит у SQL. Ми також можемо вставити запит у SQL, щоб перевірити наш вихід, тоді як на MySQL Console ми не можемо копіювати та вставляти запити. Ми повинні щоразу писати запит для виконання на консолі MySQL;

- MySQL - це сервер баз даних, з іншого боку, phpMyAdmin - це серверний інструмент для доступу до бази даних на сервері MySQL;

- Нам потрібно навчитися створювати, видаляти, змінювати, оновлювати та скидати запити до бази даних на консолі MySQL, тоді як у phpMyAdmin ми можемо це робити за допомогою графічного інтерфейсу користувача, який автоматично виконує ці запити у фоновому режимі;

### 2.3.5 using MySQLConnector

MySqlConnection - високопродуктивна бібліотека MySQL для .NET

- Асинхронний: справжній асинхронний ввід-вивод для максимальної пропускної здатності;

- Швидко: оптимізовано для швидкості та низького використання пам'яті

- Надійний: виправляє десятки відкритих помилок у Oracle Connector/NET, проходить всі тести специфікації ADO.NET;

- Передовий: перша бібліотека MySQL з підтримкою .NET Core, використовує найновіші функції .NET;

- Відкритий: ліцензія MIT, розробка відбувається на GitHub з загальнодоступними дорожніми картами, проблемами;

Дуже корисна бібліотека, без якої неможливе підключення як локального, так і віддаленого серверу. Використовується в усіх формах для підключення до бази даних.

### **3. ПРОГРАМНА РЕАЛІЗАЦІЯ**

#### **3.1 Створення БД в phpMyAdmin**

Всього за допомогою phpMyAdmin було створено 2 бази даних:

1. Diplom з такими таблицями:

- Institute
- Kafedra
- Auditories
- Laboratories
- Lab\_equipment
- Lab\_furniture
- Lab\_soft
- Passport\_lab
- Responsible
- Software
- Type\_equipment
- Type\_furniture

2. Users з таблицею:

- Users

Для зручності вводу інформації, в базі даних були створені таблиці з статичною інформацією, яка використовується для створення паспортів аудиторій. При додаванні нової лабораторії, більшість даних не потрібно вводити в ручну, така інформація як ПЗ, відповідальний, номер аудиторії, кафедра, обладнання та

меблі вже внесені в БД , тому потрібно вибрати лише доступний варіант , якщо його не має , то його можна без проблем додати до таблиці.

Додавання інформації в базу даних відбувається безпосередньо через панель phpMyAdmin.

Актуальна інформація після внесення в базу даних відображається в програмі.

За допомогою первинного ключа в БД можна створити зв'язки для зручності введення та виведення інформації з таблиць . Сама структура бази даних виглядає наступним чином(Рис 3.1):

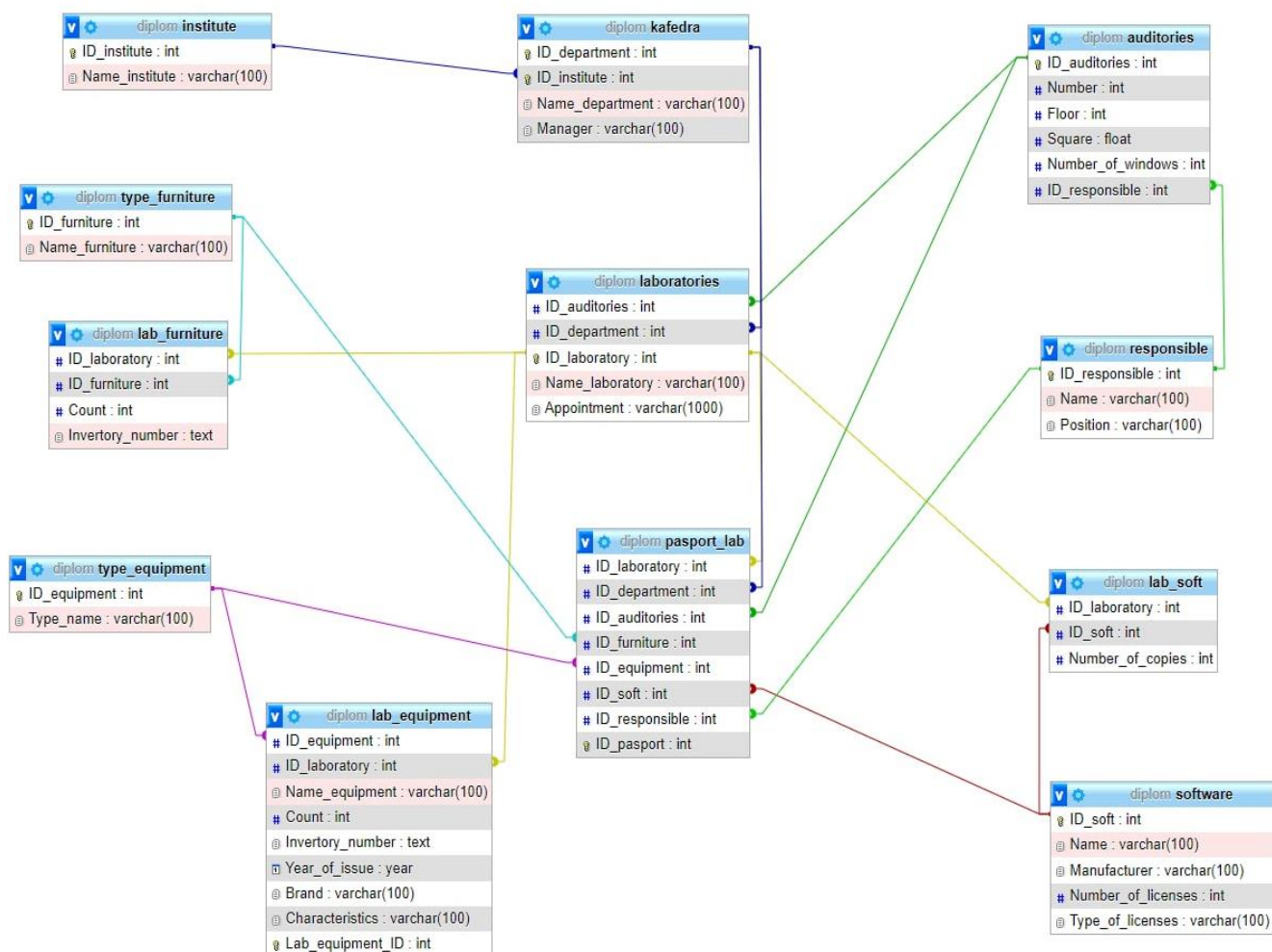


Рисунок 3.1 — Структура бази даних



В таблиці Institute 2 поля:

- ID\_institute
- Name\_institute

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 ID_institute	int			Нет	Нет		AUTO_INCREMENT	
<input type="checkbox"/>	2 Name_institute	varchar(100)	utf8mb4_0900_ai_ci		Нет	Нет			

Рисунок 3.2 — Структура таблиці Institute

В даній таблиці відображаються інститути , які внесені в базу для подальшого їх вибору в таблиці Kafedra(Рис 3.2).

В таблиці Kafedra 4 поля:

- ID\_department
- ID\_institute
- Name\_department
- Manager

В таблиці Kafedra відображається назва інституту, до якого вона відноситься, назва кафедри та завідувач кафедри. ID кафедри виставляється автоматично для подальшого використання(Рис 3.3).

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 ID_department	int			Нет	Нет		AUTO_INCREMENT	
<input type="checkbox"/>	2 ID_institute	int			Нет	Нет			
<input type="checkbox"/>	3 Name_department	varchar(100)	utf8mb4_0900_ai_ci		Нет	Нет			
<input type="checkbox"/>	4 Manager	varchar(100)	utf8mb4_0900_ai_ci		Нет	Нет			

Рисунок 3.3 — Структура таблиці Kafedra

В таблиці Auditoreis 6 полів:

- ID\_auditories
- Number

- Floor
- Square
- Number\_of\_windows
- ID\_responsible

Таблиця має свій ID, а також ID відповідального за цю аудиторію, він підтягується з іншої таблиці(responsible). Інформація щодо номеру, поверху, площі та кількістю вікон вноситься в таблицю власноруч(Рис 3.4).

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 ID_auditories	int			Нет	Нет		AUTO_INCREMENT	
<input type="checkbox"/>	2 Number	int			Нет	Нет			
<input type="checkbox"/>	3 Floor	int			Нет	Нет			
<input type="checkbox"/>	4 Square	float			Нет	Нет			
<input type="checkbox"/>	5 Number_of_windows	int			Нет	Нет			
<input type="checkbox"/>	6 ID_responsible	int			Нет	Нет			

Рисунок 3.4 — Структура таблиці Auditories

В таблиці laboratories 5 полів:

- ID\_auditories
- ID\_department
- ID\_laboratory
- Name\_laboratory
- Appointment

Номер аудиторії, назва департаменту підтягуються з відповідних таблиць. Назва лабораторії та її призначення вписується власноруч(Рис 3.5).

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 ID_auditories	int			Нет	Нет			
<input type="checkbox"/>	2 ID_department	int			Нет	Нет			
<input type="checkbox"/>	3 ID_laboratory	int			Нет	Нет		AUTO_INCREMENT	
<input type="checkbox"/>	4 Name_laboratory	varchar(100)	utf8mb4_0900_ai_ci		Нет	Нет			
<input type="checkbox"/>	5 Appointment	varchar(1000)	utf8mb4_0900_ai_ci		Нет	Нет			

Рисунок 3.5 — Структура таблиці laboratories

Таблиця responsible має 3 поля:

- ID\_responsible

- Name

- Position

Дана таблиця створена для вибору та відображення відповідального за аудиторію(Рис 3.6).

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 ID_responsible	int			Нет	Нет		AUTO_INCREMENT	
<input type="checkbox"/>	2 Name	varchar(100)	utf8mb4_0900_ai_ci		Нет	Нет			
<input type="checkbox"/>	3 Position	varchar(100)	utf8mb4_0900_ai_ci		Нет	Нет			

Рисунок 3.6 — Структура таблиці responsible

Таблиця software має 5 полів:

- ID\_soft

- Name

- Manufacturer

- Number\_of\_licenses

- Type\_of\_licenses

Створена для подальшого використання назви програм в лабораторіях(Рис 3.7).

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 ID_soft	int			Нет	Нет			
<input type="checkbox"/>	2 Name	varchar(100)	utf8mb4_0900_ai_ci		Нет	Нет			
<input type="checkbox"/>	3 Manufacturer	varchar(100)	utf8mb4_0900_ai_ci		Да	NULL			
<input type="checkbox"/>	4 Number_of_licenses	int			Да	NULL			
<input type="checkbox"/>	5 Type_of_licenses	varchar(100)	utf8mb4_0900_ai_ci		Да	NULL			

Рисунок 3.7 — Структура таблиці software

Таблиця lab\_soft має 4 поля:

- ID\_laboratory

- ID\_soft
- Number\_of\_copies
- Lab\_soft

В таблиці відображено назву лабораторії та програмне забезпечення, яке використовується, а також кількість копій ПЗ(Рис 3.8).

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 ID_laboratory	int			Нет	Нет			
<input type="checkbox"/>	2 ID_soft	int			Нет	Нет			
<input type="checkbox"/>	3 Number_of_copies	int			Да	NULL			
<input type="checkbox"/>	4 Lab_soft_ID	int			Нет	Нет		AUTO_INCREMENT	

Рисунок 3.8 — Структура таблиці lab\_soft

Таблиці Type\_equipment та Type\_furniture мають по 2 поля :

- ID\_equipment
- Name\_equipment

та

- ID\_furniture
- Name\_furniture

Використовуються для відображення назви обладнання та назви меблів для подальшого використання їх в таблицях з повною інформацією щодо кількості та інвентарного номеру обладнання і меблів(Рис 3.9).

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 ID_furniture	int			Нет	Нет		AUTO_INCREMENT	
<input type="checkbox"/>	2 Name_furniture	varchar(100)	utf8mb4_0900_ai_ci		Нет	Нет			

Рисунок 3.9 — Структура таблиці type\_furniture

Таблиця lab\_furniture має 4 поля:

- ID\_laboratory
- ID\_furniture

- Count
- Invertoory\_number

Ця таблиця показує тип, кількість та інвентарний номер меблів в кожній лабораторії(Рис 3.10).

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input checked="" type="checkbox"/>	1 ID_laboratory	int			Нет	Нет			
<input type="checkbox"/>	2 ID_furniture	int			Нет	Нет			
<input type="checkbox"/>	3 Count	int			Нет	Нет			
<input type="checkbox"/>	4 Invertoory_number	text	utf8mb4_0900_ai_ci		Да	NULL			

Рисунок 3.10 — Структура таблиці lab\_furniture

Таблиця lab\_equiprment має 9 полів:

- ID\_equiprment
- ID\_laboratory
- Name\_equiprment
- Count
- Invertoory\_number
- Year\_of\_issue
- Brand
- Characteristics
- Lab\_equiprment\_ID

Дана таблиця містить в собі всю інформацію по обладнанню в лабораторіях. А саме:

Тип обладнання, назва лабораторії, назва обладнання, кількість, інвентарний номер, рік випуску, фірма та характеристики.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 ID_equipment	int			Нет	Нем			
<input type="checkbox"/>	2 ID_laboratory	int			Нет	Нем			
<input type="checkbox"/>	3 Name_equipment	varchar(100)	utf8mb4_0900_ai_ci		Нет	Нем			
<input type="checkbox"/>	4 Count	int			Нет	Нем			
<input type="checkbox"/>	5 Inventory_number	text	utf8mb4_0900_ai_ci		Да	NULL			
<input type="checkbox"/>	6 Year_of_issue	year			Да	NULL			
<input type="checkbox"/>	7 Brand	varchar(100)	utf8mb4_0900_ai_ci		Да	NULL			
<input type="checkbox"/>	8 Characteristics	varchar(100)	utf8mb4_0900_ai_ci		Да	NULL			
<input type="checkbox"/>	9 Lab_equipment_ID	int			Нет	Нем		AUTO_INCREMENT	

Рисунок 3.11 — Структура таблиці lab\_equipment

Таблиця passport\_lab має 8 полів:

- ID\_laboratory
- ID\_department
- ID\_auditories
- ID\_furniture
- ID\_equipment
- ID\_soft
- ID\_responsible
- ID\_pasport

Дана таблиця служить для вибірки конкретної інформації для створення точного паспорту аудиторії. Має зв'язки з усіма ключовими таблицями(Рис 3.12).

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 ID_laboratory	int			Нет	Нем			
<input type="checkbox"/>	2 ID_department	int			Нет	Нем			
<input type="checkbox"/>	3 ID_auditories	int			Нет	Нем			
<input type="checkbox"/>	4 ID_furniture	int			Нет	Нем			
<input type="checkbox"/>	5 ID_equipment	int			Нет	Нем			
<input type="checkbox"/>	6 ID_soft	int			Нет	Нем			
<input type="checkbox"/>	7 ID_responsible	int			Нет	Нем			
<input type="checkbox"/>	8 ID_pasport	int			Нет	Нем		AUTO_INCREMENT	

Рисунок 3.12 — Структура таблиці passport\_lab

### 3.2 Підключення БД до проекту VisualStudio

Для підключення бази з phpMyAdmin до Visual Studio було створено окремий клас DB. Він використовується в формі для входу в систему та у формах для відображення бази даних в компоненті DataGridView.

Для створення локальної бази даних була застосована програма OpenServerPanel.

```

1
2 using System;
3 using System.Collections.Generic;
4 using System.Linq;
5 using System.Text;
6 using System.Threading.Tasks;
7 using MySqlConnection;
8
9 namespace laboratory_equipment
10 {
11     class DB
12     {
13         MySqlConnection connection = new MySqlConnection("server=localhost;username=root;password=;database=users");
14         MySqlConnection con = new MySqlConnection("server=localhost;username=root;password=;database=auditories");
15
16         public void openConnection()
17         {
18             if (connection.State == System.Data.ConnectionState.Closed)
19                 connection.Open();
20         }
21         public void closeConnection()
22         {
23             if (connection.State == System.Data.ConnectionState.Open)
24                 connection.Close();
25         }
26
27         public MySqlConnection getConnection() {
28             return connection;
29         }
30     }
31 }
32
33
34

```

Рисунок 3.13 — Реалізація класу DB

### 3.3 Робота програмного забезпечення з відображенням бази даних

Додаток в VisualStudio при запуску має поле для авторизації користувача. Користувач має буди внесений в окрему БД з вже створеним логіном та паролем для нього, як що це не так, користувачеві буде виведено діалогове вікно про помилку вводу логіну або пароллю.

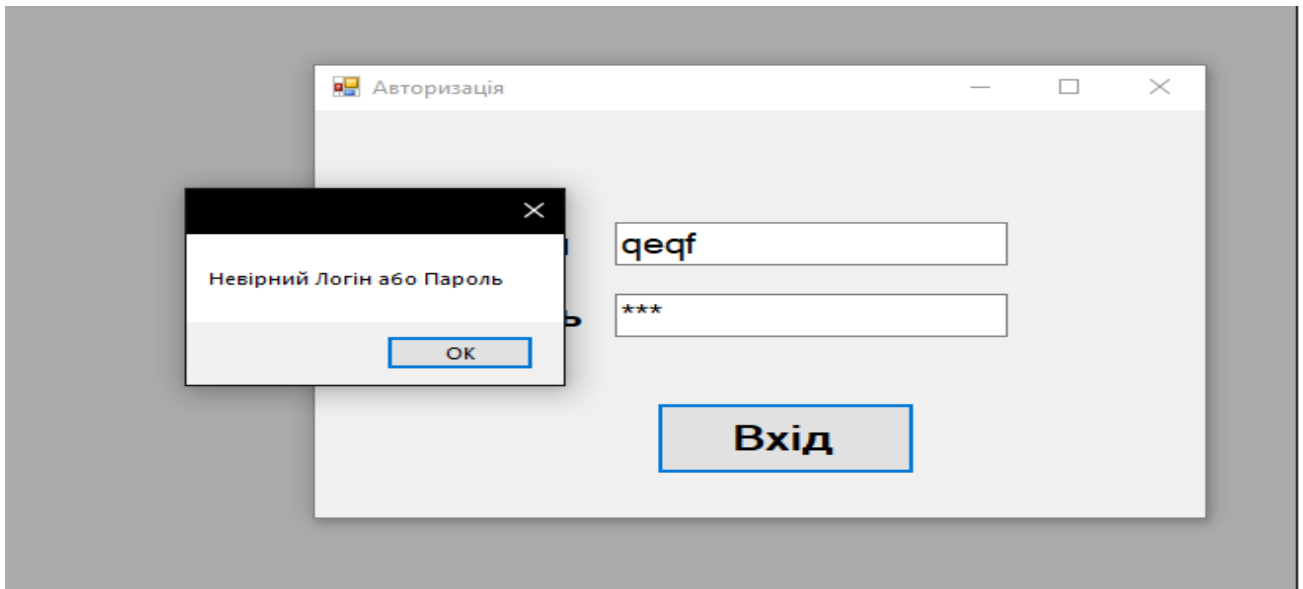


Рисунок 3.14 — Вікно входу з невірним логіном або паролем

```
private void button1_Click(object sender, EventArgs e)
{
    String loginUser = textBox1.Text;
    String passUser = textBox2.Text;

    DB db = new DB();
    DataTable table = new DataTable();
    MySqlDataAdapter adapter = new MySqlDataAdapter();

    MySqlCommand command = new MySqlCommand("SELECT * FROM `users` WHERE `login` = @uL AND `pass` = @uP", db.getConnection());
    command.Parameters.Add("@uL", MySqlDbType.VarChar).Value = loginUser;
    command.Parameters.Add("@uP", MySqlDbType.VarChar).Value = passUser;

    adapter.SelectCommand = command;
    adapter.Fill(table);

    if (table.Rows.Count > 0)
    {
        Choice s = new Choice();
        s.Show();
        this.Hide();
    }
    else
        MessageBox.Show("Невірний логін або Пароль");
}

private void textBox2_TextChanged(object sender, EventArgs e)
{
}
}
```

Рисунок 3.15 — Перевірка правильності та наявності логіну та паролю в БД

При успішній авторизації користувачеві буде надано вибір інформації, яку потрібно відобразити (Рис 3.16).



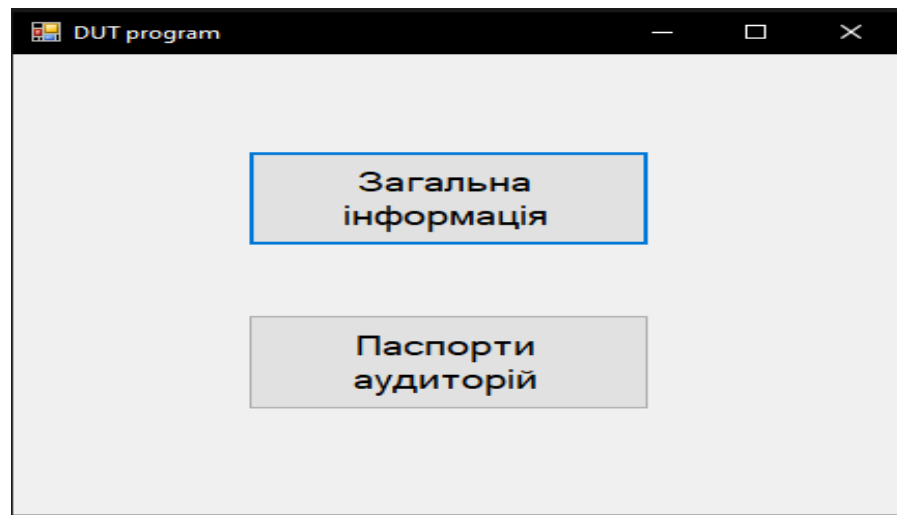


Рисунок 3.16 — Форма вибору інформації

При виборі кнопки “Загальна інформація” користувачеві буде показана нова форма та виведена база з усією інформацією щодо інститутів, кафедр, аудиторій, лабораторій, відповідальних, програмного забезпечення, типу обладнання, типу меблів і т.д.

Нижче наведено приклад того, як інформація відображається в елементі TabControl та через DataGridView(Рис 3.17).

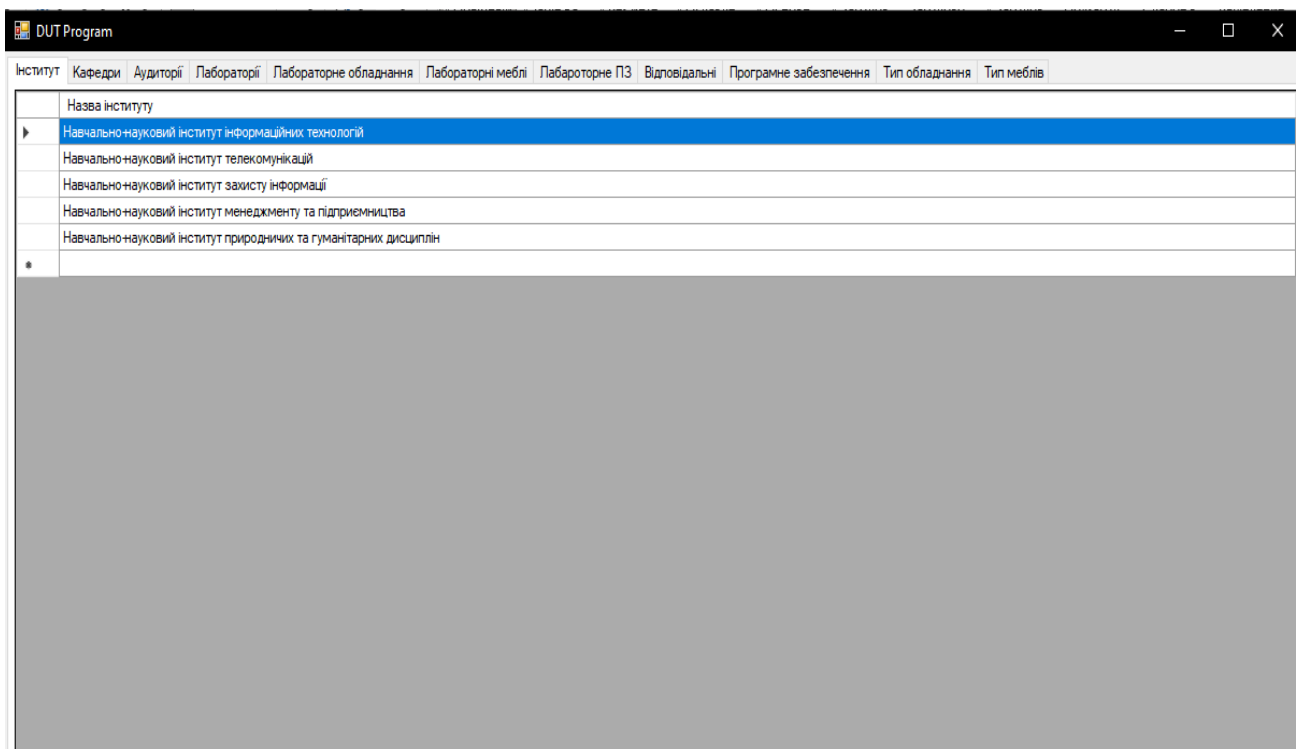


Рисунок 3.17 — Виведення в компонент DataGridView таблиці institute

Відображення кафедри відбувається за допомогою операції LEFT JOIN для усунення відображення ID замість назви Інституту

Інститут	Кафедри	Аудиторії	Лабораторії	Лабораторне обладнання	Лабораторні меблі	Лабораторне ПЗ	Відповідальні	Програмне забезпечення	Тип обладнання	Тип меблів
	Назва кафедри						Завідувач			Назва інституту
	Інженерія Програмного Забезпечення						О.В.Негоденко			Навчально-науковий інститут інформаційних технологій

Рисунок 3.18 — Виведення в компонент DataGridView таблиці kafedra

Інститут	Кафедри	Аудиторії	Лабораторії	Лабораторне обладнання	Лабораторні меблі	Лабораторне ПЗ	Відповідальні	Програмне забезпечення	Тип обладнання	Тип меблів
		Номер аудиторії	Поверх				Площа		Кількість вікон	Відповідальний
		310	3				47,9		3	А.Б.Коба
		325	3				50,1		3	А.Б.Коба
		326	3				44,7		3	А.Б.Коба

Рисунок 3.19 — Виведення в компонент DataGridView таблиці auditories

Інститут	Кафедри	Аудиторії	Лабораторії	Лабораторне обладнання	Лабораторні меблі	Лабораторне ПЗ	Відповідальні	Програмне забезпечення	Тип обладнання	Тип меблів
		Номер аудиторії	Назва кафедри				Назва лабораторії			Призначення
		310	Інженерія Програмного Забезпечення				Лабораторія EPAM та ProjectManagement			проведення практичного навчання з питань: вивчення мов програмування C++, C#, JAVA, PYTHON, JAVA SCRIPT, CSS, HTML, BUSH, PHP, навчитися працювати в хмарному середовищі Office 365 проведення наукових досліджень з питань: на базі об'єктно-орієнтованого моделювання
		325	Інженерія Програмного Забезпечення				Технології «Блок-чейн»			проведення практичного навчання з питань: вивчення мов програмування C++, C#, JAVA, PYTHON, JAVA SCRIPT, CSS, HTML, BUSH, PHP, навчитися працювати в хмарному середовищі Office 365 проведення наукових досліджень з питань: на базі об'єктно-орієнтованого моделювання
		326	Інженерія Програмного Забезпечення				Лабораторія мов програмування			проведення практичного навчання з питань: вивчення мов програмування C++, C#, JAVA, PYTHON, JAVA SCRIPT, CSS, HTML, BUSH, PHP, навчитися працювати в хмарному середовищі Office 365 проведення наукових досліджень з питань: на базі об'єктно-орієнтованого моделювання

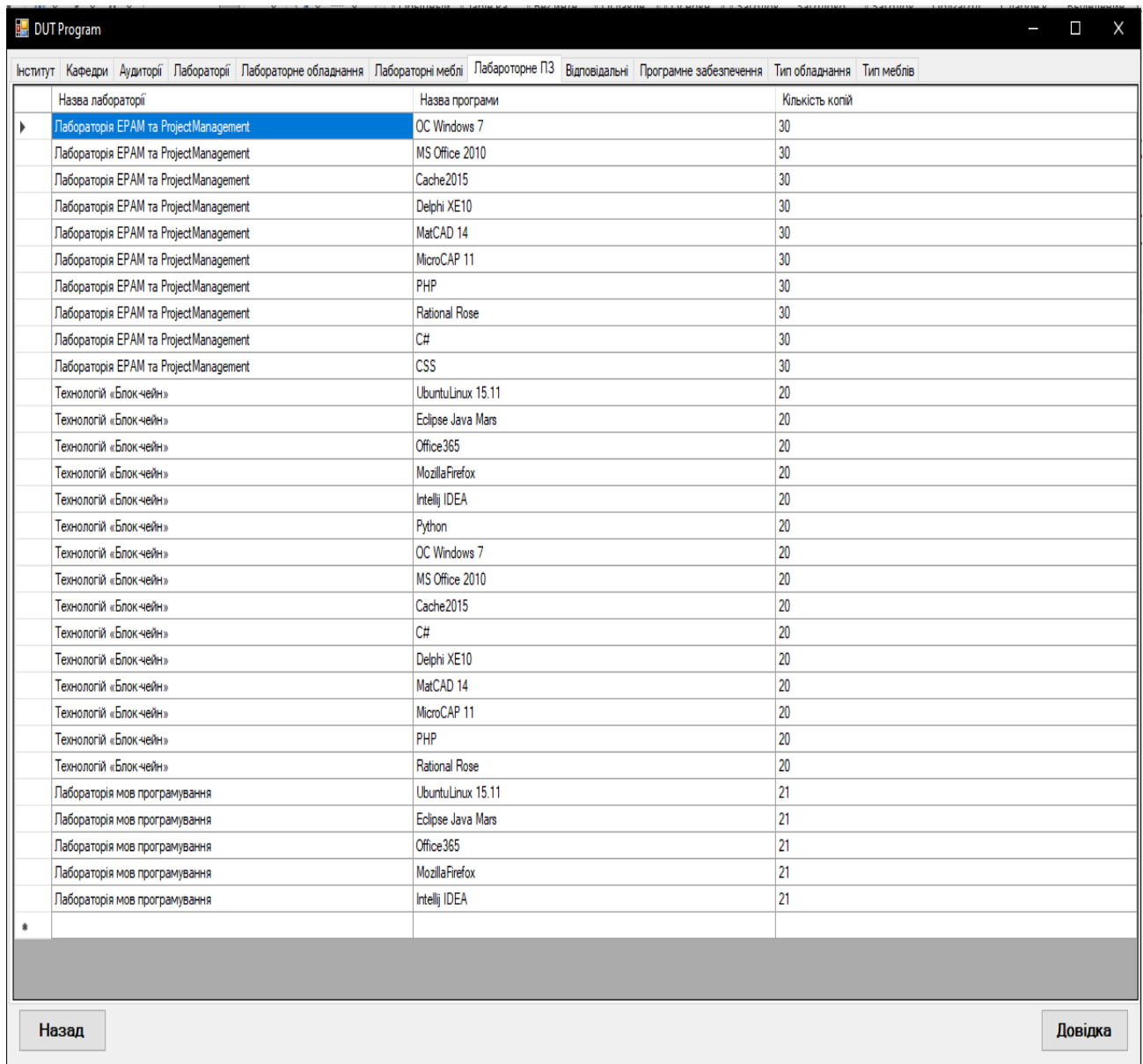
Рисунок 3.20 — Виведення в компонент DataGridView таблиці laboratories

Назва лабораторії	Тип обладнання	Назва обладнання	Кількість	Інвентарний номер	Рік випуску	Марка	Характеристики
Лабораторія EPAM та ProjectManagement	Відеопроєктор	Acer X 112 H (MR/JKV11.001)	1			Acer	
Лабораторія EPAM та ProjectManagement	Колонки	Microlab 2.0 FC30 White	1			Microlab	
Лабораторія EPAM та ProjectManagement	Комп'ютер	HP5630w Compaq 2GF/1GR MS	25			HP	
Лабораторія EPAM та ProjectManagement	Кондиціонер	підлогово-потолочного типу CH-F24NK2/CH-U24NK2	1				
Лабораторія EPAM та ProjectManagement	Монітор	Compaq LA1951g LCD Monitor	50			Compaq	
Лабораторія EPAM та ProjectManagement	Проекційний екран	Acer M80-SO 1MW	1			Acer	
Лабораторія EPAM та ProjectManagement	Сплітер	VGA 1x2 (MT-2502-A)	1				
Технології «Блок-чейн»	Відеопроєктор	Acer X 1223H(MR/JKV11.001)	1	104293562	2019	Acer	
Технології «Блок-чейн»	Комп'ютер	Everest Enterprise 7600	20	113162670 - 113162687, 113163431, 113163432	2019	виробник - TM "Everest", країна походження - Україна	Ryzen 3 4C/4T 1200 3.1/3.4 GHz; DDR4 4 Gb 2666;SSD 2.5" 120 GB; GeForceGT710-SL-1GD5; SX632CR-400W
Технології «Блок-чейн»	Монітор	Acer ET241 Ybi	20	113162670 - 113162687, 113241193, 113241194.	2019	Acer	Монітор 23.8" IPS
Технології «Блок-чейн»	Кондиціонер	Кондиціонер	1	104293563			
Технології «Блок-чейн»	Точка доступу Wi-Fi	APC 2M-14	1	113160881			
Лабораторія мов програмування	Проекційний екран	Acer M80-SO 1MW	1	113158186		Acer	
Лабораторія мов програмування	Колонки	Microlab 2.0 FC30 White	1	113161175		Microlab	
Лабораторія мов програмування	Кондиціонер	SIH/SOH-0855HDB	1	104293490			
Лабораторія мов програмування	Мережевий фільтр	4XSchuko з вимикачем 3 G1.5'3m	1	113161185			
Лабораторія мов програмування	Монітор	Acer ET240 Ybi (UM.GE1EE.001)	21	113241209 - 113241220, 113163260 - 113163268.	2020	Acer	IPS
Лабораторія мов програмування	Принтер	БФП HP LaserJet Pro M426FDN	1	104293575		HP	
Лабораторія мов програмування	Комп'ютер	Everest Enterprise 7600	21	113163447 - 113163458, 113163288 - 113163296.	2020	виробник - TM "Everest", країна походження - Україна	Ryzen 3 4C/4T 1200 3.1/3.4 GHz; DDR4 4 Gb 2666;SSD 2.5" 120 GB; GeForceGT710-SL-1GD5; SX632CR-400W
Лабораторія мов програмування	Сплітер	VGA 1x2 (MT-2502-A)	1	113160876			

Рисунок 3.21 — Виведення в компонент DataGridView таблиці lab\_equipment

Назва лабораторії	Тип меблів	Кількість	Інвентарний номер
Лабораторія EPAM та ProjectManagement	Стіл комп'ютерний	25	
Лабораторія EPAM та ProjectManagement	Дошка магнітно-маркерна ALU23	1	
Лабораторія EPAM та ProjectManagement	Жалюзі вертикальні Line 1310	3	
Лабораторія EPAM та ProjectManagement	Кріплення для проєктора Bateek PRB-2 Silver	1	
Лабораторія EPAM та ProjectManagement	Стілець для навчання	32	113160543 - 113160574
Лабораторія EPAM та ProjectManagement	Стіл учбовий для забезпечення учбового процесу	1	
Технології «Блок-чейн»	Кріплення для проєктора Bateek PRB-2 Silver	1	113158533
Технології «Блок-чейн»	Дошка магнітно-маркерна ALU23	1	113162277
Технології «Блок-чейн»	Стілець для навчання	35	113162569 - 113162603
Технології «Блок-чейн»	Стіл комп'ютерний	26	113159306, 18983, 18983
Технології «Блок-чейн»	Жалюзі вертикальні Line 1310	3	113158316 - 113158318
Лабораторія мов програмування	Жалюзі вертикальні 1750 x 2350	3	113158319 - 113158321
Лабораторія мов програмування	Кріплення для проєктора Sunte PR002	1	113158180
Лабораторія мов програмування	Дошка TSA-1510 для письма маркером	2	113158314, 113158315
Лабораторія мов програмування	Стіл комп'ютерний для учбової лабораторії	20	113158224 - 1131582243
Лабораторія мов програмування	Стіл для учбової лабораторії	3	113158427, 113158455, 113158456
Лабораторія мов програмування	Стілець для учбової лабораторії	30	113158327 - 113158355, 113158411 - 113158413.
Лабораторія мов програмування	Шафа для учбової лабораторії	1	10610388
Лабораторія мов програмування	Шафа інструментальна ШИ-15	1	113162255

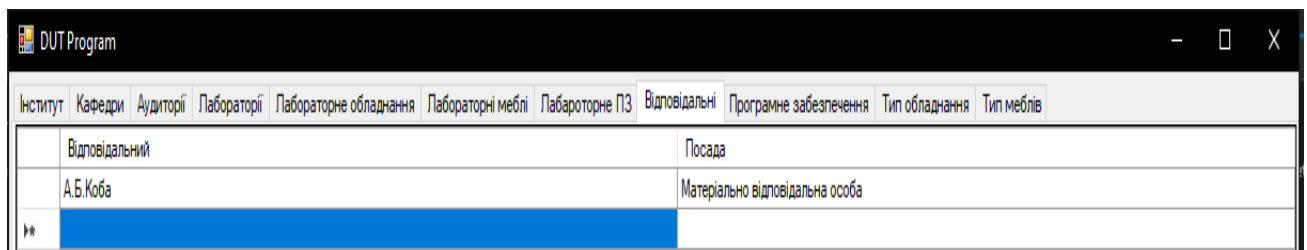
Рисунок 3.22 — Виведення в компонент DataGridView таблиці lab\_furniture



The screenshot shows the 'DUT Program' application window. At the top, there is a menu bar with options: Інститут, Кафедри, Аудиторії, Лабораторії, Лабораторне обладнання, Лабораторні меблі, Лабораторне ПЗ, Відповідальні, Програмне забезпечення, Тип обладнання, and Тип меблів. Below the menu is a table with three columns: Назва лабораторії, Назва програми, and Кількість копій. The table contains 30 rows of data, with the first row highlighted in blue. At the bottom of the window, there are two buttons: 'Назад' and 'Довідка'.

Назва лабораторії	Назва програми	Кількість копій
Лабораторія ЕРАМ та ProjectManagement	OC Windows 7	30
Лабораторія ЕРАМ та ProjectManagement	MS Office 2010	30
Лабораторія ЕРАМ та ProjectManagement	Cache2015	30
Лабораторія ЕРАМ та ProjectManagement	Delphi XE10	30
Лабораторія ЕРАМ та ProjectManagement	MatCAD 14	30
Лабораторія ЕРАМ та ProjectManagement	MicroCAP 11	30
Лабораторія ЕРАМ та ProjectManagement	PHP	30
Лабораторія ЕРАМ та ProjectManagement	Rational Rose	30
Лабораторія ЕРАМ та ProjectManagement	C#	30
Лабораторія ЕРАМ та ProjectManagement	CSS	30
Технологій «Блок-чейн»	UbuntuLinux 15.11	20
Технологій «Блок-чейн»	Eclipse Java Mars	20
Технологій «Блок-чейн»	Office365	20
Технологій «Блок-чейн»	MozillaFirefox	20
Технологій «Блок-чейн»	IntelliJ IDEA	20
Технологій «Блок-чейн»	Python	20
Технологій «Блок-чейн»	OC Windows 7	20
Технологій «Блок-чейн»	MS Office 2010	20
Технологій «Блок-чейн»	Cache2015	20
Технологій «Блок-чейн»	C#	20
Технологій «Блок-чейн»	Delphi XE10	20
Технологій «Блок-чейн»	MatCAD 14	20
Технологій «Блок-чейн»	MicroCAP 11	20
Технологій «Блок-чейн»	PHP	20
Технологій «Блок-чейн»	Rational Rose	20
Лабораторія мов програмування	UbuntuLinux 15.11	21
Лабораторія мов програмування	Eclipse Java Mars	21
Лабораторія мов програмування	Office365	21
Лабораторія мов програмування	MozillaFirefox	21
Лабораторія мов програмування	IntelliJ IDEA	21

Рисунок 3.23 — Виведення в компонент DataGridView таблиці lab\_soft



The screenshot shows the 'DUT Program' application window. At the top, there is a menu bar with options: Інститут, Кафедри, Аудиторії, Лабораторії, Лабораторне обладнання, Лабораторні меблі, Лабораторне ПЗ, Відповідальні, Програмне забезпечення, Тип обладнання, and Тип меблів. Below the menu is a table with two columns: Відповідальний and Посада. The table contains two rows of data, with the first row highlighted in blue. At the bottom of the window, there are two buttons: 'Назад' and 'Довідка'.

Відповідальний	Посада
А.Б.Коба	Матеріально відповідальна особа

Рисунок 3.24 — Виведення в компонент DataGridView таблиці responsible

Назва	Виробник	Кількість ліцензій	Тип ліцензії
OC Windows 7	Microsoft	25	
MS Office 2010	Microsoft	25	
Cache2015	InterSystems	25	
Delphi XE10	Delphi	25	
MatCAD 14	PTC	25	
MicroCAP 11	Spectrum Software	25	
PHP	PHP	25	
Rational Rose	Rational Software	25	
C#	Microsoft	25	
CSS	CSS Working Group	25	
UbuntuLinux 15.11	Canonical Ltd.	20	
Eclipse Java Mars	Java	20	
Office365	Microsoft	20	
MozillaFirefox	Mozilla Corporation	20	
IntelliJ IDEA	JetBrains	20	
Python	Python Software Foundation	20	
**			

Рисунок 3.25 — Виведення в компонент DataGridView таблиці software

Назва типу обладнання
▶ Відеопроєктор
Колонки
Комп'ютер
Кондиціонер
Монітор
Проекційний екран
Сплітер
Точка доступу Wi-Fi
Принтер
Мережевий фільтр
*

Рисунок 3.26 — Виведення в компонент DataGridView таблиці type\_equipments

Назва
▶ Стіл комп'ютерний
Дошка магнітно-маркерна ALU23
Жалюзі вертикальні Line 1310
Кріплення для проєктора Brateek PRB-2 Silver
Стілець для навчання
Стіл учбовий для забезпечення учбового процесу
Жалюзі вертикальні 1750 x 2350
Кріплення для проєктора Sunne PR002
Дошка TSA-1510 для письма маркером
Стіл комп'ютерний для учбової лабораторії
Стіл для учбової лабораторії
Стілець для учбової лабораторії
Шафа для учбової лабораторії
Шафа інструментальна ШИ-15
*

Рисунок 3.27 — Виведення в компонент DataGridView таблиці type\_furniture

При виборі кнопки “Паспорти аудиторій” користувачеві буде показана нова форма та виведена база з вибіркою інформації щодо паспортів кожної аудиторії в базі даних. Також у користувача є можливість експорту даних із цієї таблиці в документ Word.

Назва лабораторії	Назва кафедри	Номер аудиторії	Поверх	Площа	Меблі	Обладнання	Програмне забезпечення	Відповідальний
Лабораторія EPAM та Project Management	Інженерія Програмного Забезпечення	310	3	47,9	Стіл комп'ютерний, Дошка магнітно-маркерна ALU23, Жалюзі вертикальні Line 1310, Кріплення для проектора Vrateek PRB-2 Silver, Стілець для навчання, Стіл учбовий для забезпечення учбового процесу	Відеопроєктор, Колонки, Комп'ютер, Монітор, Проекційний екран, Сплітер	OC Windows 7, MS Office 2010, Cache2015, Delphi XE10, MatCAD 14, MicroCAP 11, PHP, Rational Rose, C#, CSS	А.Б.Коба
Технологій «Блок-чейн»	Інженерія Програмного Забезпечення	325	3	50,1	Кріплення для проектора Vrateek PRB-2 Silver, Дошка магнітно-маркерна ALU23, Стілець для навчання, Стіл комп'ютерний, Жалюзі вертикальні Line 1310	Відеопроєктор, Комп'ютер, Монітор, Кондиціонер, Точка доступу Wi-Fi	UbuntuLinux 15.11, Eclipse Java Mars, Office365, MozillaFirefox, IntelliJ IDEA, Python, OC Windows 7, MS Office 2010, Cache2015, C#, Delphi XE10, MatCAD 14, MicroCAP 11, PHP, Rational Rose	А.Б.Коба
Лабораторія мов програмування	Інженерія Програмного Забезпечення	326	3	44,7	Жалюзі вертикальні 1750 x 2350, Кріплення для проектора Suppe PR002, Дошка TSA-1510 для письма маркером, Стіл комп'ютерний для учбової лабораторії, Стіл для учбової лабораторії, Стілець для учбової лабораторії, Шафа для учбової лабораторії, Шафа інструментальна ШИ-15	Проекційний екран, Колонки, Кондиціонер, мережевий фільтр, Монітор, Принтер, Комп'ютер, Сплітер	UbuntuLinux 15.11, Eclipse Java Mars, Office365, MozillaFirefox, IntelliJ IDEA	А.Б.Коба
»*								

Назад      Export to Word      Довідка

Рисунок 3.28 — Виведення в компонент DataGridView таблиці passport\_lab

Для вибірки потрібної інформації із багатьох таблиць бази було використано функцію `SELECT GROUP_CONCAT`.

Для експорту даних з таблиці потрібно натиснути на кнопку ‘Export to Word’, після чого користувачеві буде запропоновано зберегти файл у потрібну йому папку

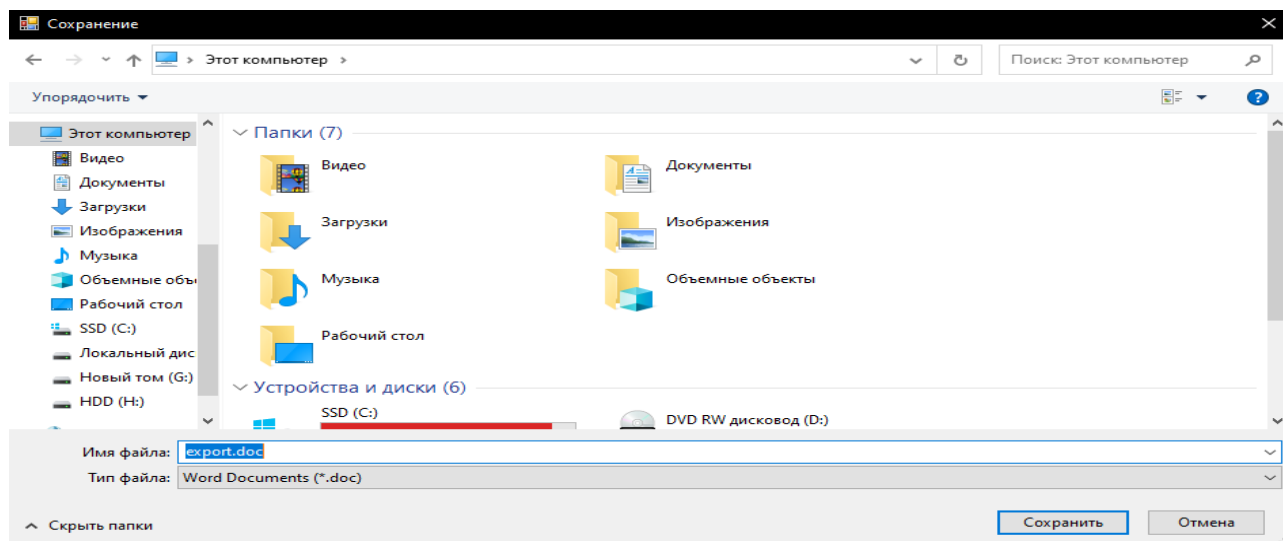


Рисунок 3.29 — Экспорт таблиці passport\_lab в Word документ

### 3.4 Тестування програмного забезпечення

Основною задачею тестування є перевірка програмного забезпечення на відповідність вимог, які поставлені до нього. Згідно вимогам створюються тест кейси, які повинні перевіряти правильність роботи програмного забезпечення, описуючи поетапно шляхи для відтворення ситуації відповідно до вимог. У випадку, якщо функціонал не відповідає вимогам, фіксується баг. В залежності від своєї важливості, баги сортують по ступеням, зазвичай це цифри від 1-5, де 1 це критичний баг, який потрібно фіксувати в першу чергу, а 5ий навряд чи хтось буде брати до уваги.

Таблиця – 3.1 Вимоги до програмного забезпечення

Назва	Опис вимоги
Авторизація	Вхід в програму повинен виконуватись лише за умови вводу правильної комбінації логін/пароль (Admin/123). При вводі іншої комбінації, користувач отримає сповіщення з текстом (Логін або Пароль введено не вірно)

Перехід між формами	При натисканні на відповідні кнопки у вікні додатку, повинен відбуватися перехід між формами та відображення тієї інформації, для якої призначалась дана кнопка. Якщо це кнопка (Назад), то відповідно вона повина перекинути користувача на попередню форму.
Поле DataGridView	Дане поле не можливо змінити, вписати туди свою інформацію, стерти дані, які в ній знаходяться. Воно повине бути заблоковане для змін.

Таблиця 3.2 – Тест кейс для авторизації користувача

№	Дії	Очікуваний результат	Статус проходження
1	1.Запустити програму 2.Ввести правильну комбінацію логіну та пароллю (Admin/123)	Успішна авторизація в програму	pass
2	1.Запустити програму 2.Ввести не правильну комбінацію логіну та пароллю (Adm/1234)	Користувачу відображається діалогове вікно з текстом (Невірний Логін або Пароль)	pass

Таблиця 3.3 – Тест кейс для вимоги “Перехід між формами”



№	Дії	Очікуваний результат	Статус проходження
1	1.Запустити програму 2.Ввести комбінацію логін пароль (Admin/123) 3.Натиснути на кнопку “Загальна інформація”	Відкриється форма з всією інформацією щодо інститутів, кафедр, аудиторій, лабораторій, обладнання які є в Державному Університеті телекомунікацій	pass
2	1.Запустити програму 2.Ввести комбінацію логін пароль (Admin/123) 3.Натиснути на кнопку “Паспорти аудиторій”	Відкриється форма з вибіркою інформації щодо конкретних лабораторій та аудиторій в яких вони знаходяться.	pass
3	1.Запустити програму 2.Ввести комбінацію логін пароль (Admin/123) 3.Натиснути на кнопку “Паспорти аудиторій” або “Загальна інформація” 4.Натиснути на кнопку “Назад”	Перехід на форму вибору інформації, яку потрібно відобразити	pass

Таблиця 3.4 – Тест кейс для поля DataGridView

№	Дії	Очікуваний результат	Статус проходження
---	-----	----------------------	--------------------

1	<p>1.Запустити програму</p> <p>2.Ввести комбінацію логін пароль (Admin/123)</p> <p>3.Натиснути на кнопку “Загальна інформація”</p> <p>4.Спробувати написати у вільне поле DataGridView будь яку інформацію та стерти існуючу</p>	<p>Поле DataGridView не повине бути доступне для запису та зміни з інтерфейсу користувача</p>	pass
2	<p>1.Запустити програму</p> <p>2.Ввести комбінацію логін пароль (Admin/123)</p> <p>3.Натиснути на кнопку “Загальна інформація”</p> <p>4.Спробувати змінити розмір тіблиці у DataGridView</p>	<p>Поле DataGridView не повинно змінювати розмір з інтерфейсу користувача</p>	fail

## ВИСНОВКИ

1. Проведено аналіз структури та форматів документації, яка використовується для обліку обладнання в навчальних лабораторіях кафедр Державного університету телекомунікацій. Визначено недоліки існуючого обліку із використанням Microsoft Word.

2. Виконано огляд програмного забезпечення, яке може бути застосовано для обліку обладнання в навчальних лабораторіях кафедр, визначено особливості роботи програм, їх недоліки та переваги. Зазначена інформація використана при проектування та розробці нового програмного продукту.

3. Спроектовано та реалізовано реляційну базу даних для збереження інформації про обладнання та програмне забезпечення, яке знаходиться в лабораторіях, та інші дані відповідно до паспортів кабінетів.

4. Спроектовано та реалізовано інтерфейс системи, який забезпечує ключові операції обліку обладнання навчальних лабораторій кафедр, в тому числі вибірку необхідної інформації з бази даних паспортів кабінетів, експорт даних в Microsoft Word, підтримка актуальної інформації в БД.

Перспективи розвитку системи:

- інтеграція з бухгалтерською системою Державного університету телекомунікацій;
- розширення переліку звітів та зведених таблиць, які генерує система.

## ПЕРЕЛІК ПОСИЛАНЬ

1. <https://www.techtarget.com/searchdatamanagement/definition/SQL-Server#:~:text=The%20core%20component%20of%20Microsoft,indexes%2C%20data%20buffers%20and%20transactions.>
2. <https://docs.microsoft.com/en-us/dotnet/desktop/winforms/overview/?view=netdesktop-6.0>
3. <https://docs.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2022>
4. <https://www.incredibuild.com/integrations/visual-studio#:~:text=is%20Visual%20Studio%3F-Microsoft%20Visual%20Studio%20is%20an%20IDE%20made%20by%20Microsoft%20and,facilitate%20the%20software%20development%20process.>
5. <https://www.geeksforgeeks.org/introduction-to-visual-studio/>
6. <https://softwarekeep.com/help-center/what-is-microsoft-visual-studio-where-can-i-download-it>
7. <https://docs.microsoft.com/ru-ru/sql/sql-server/?view=sql-server-2017&viewFallbackFrom=sql-server-2022>
8. <https://docs.microsoft.com/en-us/dotnet/api/system.data.entity>
9. Энтони М. SQL. Сборник рецептов / Молинаро Энтони // SQL. Сборник рецептов / Молинаро Энтони.
10. <https://metanit.com/sharp/windowsforms/1.1.php>
11. [https://professorweb.ru/my/csharp/charp\\_theory/level2/2\\_4.php](https://professorweb.ru/my/csharp/charp_theory/level2/2_4.php)
12. <https://ospanel.io/>
13. Козак І.А., Ковальська Л.А. Удосконалення документаційного забезпечення управління підприємством на основі використання інформаційних технологій. Вісник студентського наукового товариства ДонНУ імені Василя Стуса, Вип. 1, №. 12,. 2020. С. 164-168
14. Правила організації діловодства та архівного зберігання документів у державних органах, органах місцевого самоврядування, на підприємствах, в

установах і організаціях, затверджених наказом Міністерства юстиції 85 України від 18 червня 2015 року № 1000/5, зареєстрованих в Міністерстві юстиції України 22 червня 2015 року за № 736/27181.

15. Буч Г., Рамбо Дж., Джекобсон А. Язык UML. Руководство пользователя: Пер. с англ. – М.: ДМК, 2000.

16. Ларман К. Применение UML и шаблонов проектирования: Пер. с англ.: Учебное пособие – М.: Вильямс, 2001.

17. Боггс У., Боггс М. UML и Rational Rose: Пер. с англ. – М.: Лори, 2000.

18. Леоненков В. А. Самоучитель UML – Санкт-Петербург: ВHV, 2001.

19. Коберн А. Современные методы описания функциональных требований к системам.: Пер. с англ. – М.: ЛОРИ, 2002.

20. Брайан Клифтон // Google Analytics для профессионалов // – Санкт-Петербург: Диалектика, 2013. – 608с

## ДОДАТОК 1

### Лістинг програми

#### Метод підключення до БД:

```
using MySqlConnection;
class DB
{
    MySqlConnection connection = new
MySqlConnection("server=localhost;username=root;password=;database=users");
    MySqlConnection con = new
MySqlConnection("server=localhost;username=root;password=;database=auditories");
    public void openConnection()
    {
        if (connection.State == System.Data.ConnectionState.Closed)
            connection.Open();
    }
    public void closeConnection()
    {
        if (connection.State == System.Data.ConnectionState.Open)
            connection.Close();
    }
    public MySqlConnection getConnection() {
        return connection;
    }
}
```

#### Авторизація користувача:

```
using MySqlConnection;
private void button1_Click(object sender, EventArgs e)
{
    String loginUser = textBox1.Text;
    String passUser = textBox2.Text;
    DB db = new DB();
    DataTable table = new DataTable();
    MySqlDataAdapter adapter = new MySqlDataAdapter();
    MySqlCommand command = new MySqlCommand("SELECT * FROM `users` WHERE `login` =
@uL AND `pass` = @uP", db.getConnection());
    command.Parameters.Add("@uL", MySqlDbType.VarChar).Value = loginUser;
    command.Parameters.Add("@uP", MySqlDbType.VarChar).Value = passUser;
    adapter.SelectCommand = command;
    adapter.Fill(table);
    if (table.Rows.Count > 0)
    {
        Choice s = new Choice();
        s.Show();
        this.Hide();
    }
    else
        MessageBox.Show("Невірний Логін або Пароль");
}
```

Метод виведення інформації із таблиць в оптимізований під відображення великої кількості даних компонент DataGridView:

```
private void Equipments_Load(object sender, EventArgs e)
{
    Dbauditories db = new Dbauditories();
```

```

dataGridView1.AutoSizeRowsMode = DataGridViewAutoSizeRowsMode.AllCells;
dataGridView1.DefaultCellStyle.WrapMode = DataGridViewTriState.True;
dataGridView2.AutoSizeRowsMode = DataGridViewAutoSizeRowsMode.AllCells;
dataGridView2.DefaultCellStyle.WrapMode = DataGridViewTriState.True;
dataGridView3.AutoSizeRowsMode = DataGridViewAutoSizeRowsMode.AllCells;
dataGridView3.DefaultCellStyle.WrapMode = DataGridViewTriState.True;
dataGridView4.AutoSizeRowsMode = DataGridViewAutoSizeRowsMode.AllCells;
dataGridView4.DefaultCellStyle.WrapMode = DataGridViewTriState.True;
dataGridView5.AutoSizeRowsMode = DataGridViewAutoSizeRowsMode.AllCells;
dataGridView5.DefaultCellStyle.WrapMode = DataGridViewTriState.True;
dataGridView6.AutoSizeRowsMode = DataGridViewAutoSizeRowsMode.AllCells;
dataGridView6.DefaultCellStyle.WrapMode = DataGridViewTriState.True;
dataGridView7.AutoSizeRowsMode = DataGridViewAutoSizeRowsMode.AllCells;
dataGridView7.DefaultCellStyle.WrapMode = DataGridViewTriState.True;
dataGridView8.AutoSizeRowsMode = DataGridViewAutoSizeRowsMode.AllCells;
dataGridView8.DefaultCellStyle.WrapMode = DataGridViewTriState.True;
dataGridView9.AutoSizeRowsMode = DataGridViewAutoSizeRowsMode.AllCells;
dataGridView9.DefaultCellStyle.WrapMode = DataGridViewTriState.True;
dataGridView10.AutoSizeRowsMode = DataGridViewAutoSizeRowsMode.AllCells;
dataGridView10.DefaultCellStyle.WrapMode = DataGridViewTriState.True;
dataGridView11.AutoSizeRowsMode = DataGridViewAutoSizeRowsMode.AllCells;
dataGridView11.DefaultCellStyle.WrapMode = DataGridViewTriState.True;

```

```

MySqlDataAdapter adapter = new MySqlDataAdapter("SELECT institute.Name_institute AS 'Назва
інституту' FROM `institute` ", db.getConnection());

```

```

DataSet institute = new DataSet();
adapter.Fill(institute);
dataGridView1.DataSource = institute.Tables[0];

```

```

//-----

```

```

MySqlDataAdapter adapter1 = new MySqlDataAdapter("SELECT kafedra.Name_department AS 'Назва
кафедри', kafedra.Manager AS 'Завідувач', institute.Name_institute AS 'Назва інституту' FROM `kafedra` LEFT JOIN
institute ON kafedra.ID_institute=institute.ID_institute", db.getConnection());

```

```

DataSet kafedra = new DataSet();
adapter1.Fill(kafedra);
dataGridView2.DataSource = kafedra.Tables[0];

```

```

//-----

```

```

MySqlDataAdapter adapter2 = new MySqlDataAdapter("SELECT auditories.Number AS 'Номер
аудиторії', auditories.Floor AS 'Поверх', auditories.Square AS 'Площа', auditories.Number_of_windows AS
'Кількість вікон', responsible.Name AS 'Відповідальний' FROM `auditories` LEFT JOIN responsible ON
auditories.ID_responsible=responsible.ID_responsible", db.getConnection());

```

```

DataSet auditories = new DataSet();
adapter2.Fill(auditories);
dataGridView3.DataSource = auditories.Tables[0];

```

```

//-----

```

```

MySqlDataAdapter adapter3 = new MySqlDataAdapter("SELECT auditories.Number AS 'Номер
аудиторії', kafedra.Name_department AS 'Назва кафедри', laboratories.Name_laboratory AS 'Назва лобараторії'
, laboratories.Appointment AS 'Призначення' FROM `laboratories` LEFT JOIN auditories ON
laboratories.ID_auditories=auditories.ID_auditories LEFT JOIN kafedra ON
laboratories.ID_department=kafedra.ID_department", db.getConnection());

```

```

DataSet laboratories = new DataSet();
adapter3.Fill(laboratories);
dataGridView4.DataSource = laboratories.Tables[0];

```

```

//-----

```

```

MySqlDataAdapter adapter4 = new MySqlDataAdapter("SELECT laboratories.Name_laboratory
AS 'Назва лабораторії', type_equipment.Туре_name AS 'Тип обладнання', lab_equipment.Name_equipment AS
'Назва обладнання', lab_equipment.Count AS 'Кількість', lab_equipment.Inventory_number AS 'Інвентарний
номер', lab_equipment.Year_of_issue AS 'Рік випуску', lab_equipment.Brand AS 'Марка',
lab_equipment.Characteristics AS 'Характеристики' FROM `lab_equipment` LEFT JOIN laboratories ON

```

```

lab_equipment.ID_laboratory=laboratories.ID_laboratory LEFT JOIN type_equipment ON
lab_equipment.ID_equipment=type_equipment.ID_equipment", db.getConnection());

        DataSet lab_equipment = new DataSet();
        adapter4.Fill(lab_equipment);
        dataGridView5.DataSource = lab_equipment.Tables[0];
//-----
        MySqlDataAdapter adapter5 = new MySqlDataAdapter("SELECT laboratories.Name_laboratory
AS 'Назва лабораторії', type_furniture.Name_furniture AS 'Тип меблів', lab_furniture.Count AS 'Кількість',
lab_furniture.Inventory_number AS 'Інвентарний номер' FROM `lab_furniture` LEFT JOIN laboratories ON
lab_furniture.ID_laboratory=laboratories.ID_laboratory LEFT JOIN type_furniture ON
lab_furniture.ID_furniture=type_furniture.ID_furniture", db.getConnection());

        DataSet lab_furniture = new DataSet();
        adapter5.Fill(lab_furniture);
        dataGridView6.DataSource = lab_furniture.Tables[0];
//-----

        MySqlDataAdapter adapter6 = new MySqlDataAdapter("SELECT laboratories.Name_laboratory
AS 'Назва лабораторії', software.Name AS 'Назва програми', lab_soft.Number_of_copies AS 'Кількість копій'
FROM `lab_soft` LEFT JOIN software ON lab_soft.ID_soft=software.ID_soft LEFT JOIN laboratories ON
lab_soft.ID_laboratory=laboratories.ID_laboratory", db.getConnection());

        DataSet lab_soft = new DataSet();
        adapter6.Fill(lab_soft);
        dataGridView7.DataSource = lab_soft.Tables[0];
//-----

        MySqlDataAdapter adapter7 = new MySqlDataAdapter("SELECT responsible.Name AS
'Відповідальний', responsible.Position AS 'Посада' FROM `responsible`", db.getConnection());

        DataSet responsible = new DataSet();
        adapter7.Fill(responsible);
        dataGridView8.DataSource = responsible.Tables[0];
//-----

        MySqlDataAdapter adapter8 = new MySqlDataAdapter("SELECT software.Name AS 'Назва',
software.Manufacturer AS 'Виробник', software.Number_of_licenses AS 'Кількість ліцензій',
software.Type_of_licenses AS 'Тип ліцензії' FROM `software`", db.getConnection());

        DataSet software = new DataSet();
        adapter8.Fill(software);
        dataGridView9.DataSource = software.Tables[0];
//-----

        MySqlDataAdapter adapter9 = new MySqlDataAdapter("SELECT type_equipment.Type_name AS
'Назва типу обладнання' FROM `type_equipment`", db.getConnection());

        DataSet type_equipment = new DataSet();
        adapter9.Fill(type_equipment);
        dataGridView10.DataSource = type_equipment.Tables[0];
//-----

        MySqlDataAdapter adapter10 = new MySqlDataAdapter("SELECT type_furniture.Name_furniture
AS 'Назва' FROM `type_furniture`", db.getConnection());

        DataSet type_furniture = new DataSet();
        adapter10.Fill(type_furniture);
        dataGridView11.DataSource = type_furniture.Tables[0];

    }

```

**Метод вибірки інформації з таблиць в окремий компонент DataGridView:**

```

Dbauditories db = new Dbauditories();

```

```

dataGridView12.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.AllCells;

```



```
dataGridView12.DefaultCellStyle.WrapMode = DataGridViewTriState.True;
```

```
MySqlDataAdapter adapter11 = new MySqlDataAdapter("SELECT laboratories.Name_laboratory AS
'Назва лабораторії', kafedra.Name_department AS 'Назва кафедри', auditories.Number AS 'Номер аудиторії',
auditories.Floor AS 'Поверх', auditories.Square AS 'Площа', (SELECT
GROUP_CONCAT(type_furniture.Name_furniture SEPARATOR ', ') FROM type_furniture , lab_furniture
WHERE type_furniture.ID_furniture=lab_furniture.ID_furniture and
passport_lab.ID_laboratory=lab_furniture.ID_laboratory) AS 'Меблі', (SELECT
GROUP_CONCAT(type_equipment.Type_name SEPARATOR ', ') FROM type_equipment , lab_equipment
WHERE type_equipment.ID_equipment=lab_equipment.ID_equipment and
lab_equipment.ID_laboratory=passport_lab.ID_laboratory) AS 'Обладнання', (SELECT
GROUP_CONCAT(software.Name SEPARATOR ', ') FROM software , lab_soft WHERE
software.ID_soft=lab_soft.ID_soft and passport_lab.ID_laboratory=lab_soft.ID_laboratory) AS 'Програмне
забезпечення', responsible.Name AS 'Відповідальний' FROM `passport_lab` LEFT JOIN laboratories ON
passport_lab.ID_laboratory=laboratories.ID_laboratory LEFT JOIN auditories ON
passport_lab.ID_auditories=auditories.ID_auditories LEFT JOIN kafedra ON
passport_lab.ID_department=kafedra.ID_department LEFT JOIN responsible ON
passport_lab.ID_responsible=responsible.ID_responsible ", db.getConnection());
```

```
DataSet passport_lab = new DataSet();
adapter11.Fill(passport_lab);
dataGridView12.DataSource = passport_lab.Tables[0];
```

### Метод експорту інформації у Word файл:

```
string stOutput = "";

string sHeaders = "";

for (int j = 0; j < dGV.Columns.Count; j++)
    sHeaders = sHeaders.ToString() + Convert.ToString(dGV.Columns[j].HeaderText) + "\t";
stOutput += sHeaders + "\r\n";

for (int i = 0; i < dGV.RowCount - 1; i++)
{
    string stLine = "";
    for (int j = 0; j < dGV.Rows[i].Cells.Count; j++)
        stLine = stLine.ToString() + Convert.ToString(dGV.Rows[i].Cells[j].Value) + "\t";
    stOutput += stLine + "\r\n";
}
Encoding utf8 = Encoding.GetEncoding(1251);
byte[] output = utf8.GetBytes(stOutput);
FileStream fs = new FileStream(filename, FileMode.Create);
BinaryWriter bw = new BinaryWriter(fs);
bw.Write(output, 0, output.Length);
bw.Flush();
bw.Close();
fs.Close();
}

private void button3_Click(object sender, EventArgs e)
{
    SaveFileDialog sfd = new SaveFileDialog();

    sfd.Filter = "Word Documents (*.doc)*.doc";

    sfd.FileName = "export.doc";

    if (sfd.ShowDialog() == DialogResult.OK)

        ToCsV(dataGridView12, sfd.FileName);
```

## ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ  
 НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ  
 ТЕХНОЛОГІЙ  
 КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



### Розробка програмного забезпечення для обліку обладнання навчальних лабораторій кафедр засобами мови C#

Виконав студент 5 курсу  
 групи ППЗ-51  
 Сидоренко Владислав Олександрович  
 Керівник роботи  
 к.т.н., доц., доцент каф. ІПЗ  
 Золотухіна Оксана Анатоліївна

Київ – 2022

## АНАЛОГИ

Microsoft Excel



№	Назва	Конфігурація	Тип	Ціна	Дата
3	Intel Celeron 2006	MB Asus P4S 800HV / 128 Mb DDR / HDD 40.0GB Samsung 7200 rpm / FDD 1.44 / Video GeForce MK440 64Mb / Sound / Case ATX 300W / Keyboard	Каб	10 340.00	21.12.2011
17	LCD Acer AL1511	Дягномаль 15"	Каб	7 090.00	16.12.2011
				17 430.00	
<b>Admin</b>					
1	Intel Celeron 2800	MB Asus P4S 800HV / 128 Mb DDR / HDD 40.0GB Samsung 7200 rpm / FDD 1.44 / Video GeForce MK440 64Mb / Sound / Case ATX 300W / Keyboard	Каб	8 235.50	28.12.2011
2	Intel Celeron 2932	MB Asus P4S 800HV / 128 Mb DDR / HDD 40.0GB Samsung 7200 rpm / FDD 1.44 / Video GeForce MK440 64Mb / Sound / Case ATX 300W / Keyboard	Рам	9 669.50	21.12.2011
14	LCD Acer AL1511	Дягномаль 15"	Рам	7 090.00	16.12.2011
26	Слов ЛангВет 1520	A4, 140w, 1300x600обр, ОЗУ 2 Mb, USB	Каб	4 950.00	10.02.2011
28	Nokia 6100	QSPM00/1800/1900, 7Мв, вібродвигун, фотофункція, камерний джойстик, Wi-Fi, GPS	Каб	1 000.00	07.09.2011
				30 943.00	
<b>D.Krabov</b>					
7	AHD Semprom 2800	MB Gigabyte GA-K8NS 4.75W 128 Mb DDR / HDD 40.0GB Samsung 7200rpm / FDD 1.44 / Video GeForce MK440 64Mb / Sound / Case ATX 300W / Keyboard	Каб	8 693.00	21.12.2011
16	LCD Acer AL1511	Дягномаль 15"	Каб	7 090.00	16.12.2011
29	Nokia 6100	QSPM00/1800/1900, 7Мв, вібродвигун, фотофункція, камерний джойстик, Wi-Fi, GPS	Каб	1 000.00	07.09.2011
				16 783.00	

Egman



№	Назва	Конфігурація	Тип	Ціна	Дата
3	Intel Celeron 2006	MB Asus P4S 800HV / 128 Mb DDR / HDD 40.0GB Samsung 7200 rpm / FDD 1.44 / Video GeForce MK440 64Mb / Sound / Case ATX 300W / Keyboard	Каб	10 340.00	21.12.2011
17	LCD Acer AL1511	Дягномаль 15"	Каб	7 090.00	16.12.2011
				17 430.00	
<b>Admin</b>					
1	Intel Celeron 2800	MB Asus P4S 800HV / 128 Mb DDR / HDD 40.0GB Samsung 7200 rpm / FDD 1.44 / Video GeForce MK440 64Mb / Sound / Case ATX 300W / Keyboard	Каб	8 235.50	28.12.2011
2	Intel Celeron 2932	MB Asus P4S 800HV / 128 Mb DDR / HDD 40.0GB Samsung 7200 rpm / FDD 1.44 / Video GeForce MK440 64Mb / Sound / Case ATX 300W / Keyboard	Рам	9 669.50	21.12.2011
14	LCD Acer AL1511	Дягномаль 15"	Рам	7 090.00	16.12.2011
26	Слов ЛангВет 1520	A4, 140w, 1300x600обр, ОЗУ 2 Mb, USB	Каб	4 950.00	10.02.2011
28	Nokia 6100	QSPM00/1800/1900, 7Мв, вібродвигун, фотофункція, камерний джойстик, Wi-Fi, GPS	Каб	1 000.00	07.09.2011
				30 943.00	
<b>D.Krabov</b>					
7	AHD Semprom 2800	MB Gigabyte GA-K8NS 4.75W 128 Mb DDR / HDD 40.0GB Samsung 7200rpm / FDD 1.44 / Video GeForce MK440 64Mb / Sound / Case ATX 300W / Keyboard	Каб	8 693.00	21.12.2011
16	LCD Acer AL1511	Дягномаль 15"	Каб	7 090.00	16.12.2011
29	Nokia 6100	QSPM00/1800/1900, 7Мв, вібродвигун, фотофункція, камерний джойстик, Wi-Fi, GPS	Каб	1 000.00	07.09.2011
				16 783.00	

IT Invent



№	Назва	Конфігурація	Тип	Ціна	Дата
3	Intel Celeron 2006	MB Asus P4S 800HV / 128 Mb DDR / HDD 40.0GB Samsung 7200 rpm / FDD 1.44 / Video GeForce MK440 64Mb / Sound / Case ATX 300W / Keyboard	Каб	10 340.00	21.12.2011
17	LCD Acer AL1511	Дягномаль 15"	Каб	7 090.00	16.12.2011
				17 430.00	
<b>Admin</b>					
1	Intel Celeron 2800	MB Asus P4S 800HV / 128 Mb DDR / HDD 40.0GB Samsung 7200 rpm / FDD 1.44 / Video GeForce MK440 64Mb / Sound / Case ATX 300W / Keyboard	Каб	8 235.50	28.12.2011
2	Intel Celeron 2932	MB Asus P4S 800HV / 128 Mb DDR / HDD 40.0GB Samsung 7200 rpm / FDD 1.44 / Video GeForce MK440 64Mb / Sound / Case ATX 300W / Keyboard	Рам	9 669.50	21.12.2011
14	LCD Acer AL1511	Дягномаль 15"	Рам	7 090.00	16.12.2011
26	Слов ЛангВет 1520	A4, 140w, 1300x600обр, ОЗУ 2 Mb, USB	Каб	4 950.00	10.02.2011
28	Nokia 6100	QSPM00/1800/1900, 7Мв, вібродвигун, фотофункція, камерний джойстик, Wi-Fi, GPS	Каб	1 000.00	07.09.2011
				30 943.00	
<b>D.Krabov</b>					
7	AHD Semprom 2800	MB Gigabyte GA-K8NS 4.75W 128 Mb DDR / HDD 40.0GB Samsung 7200rpm / FDD 1.44 / Video GeForce MK440 64Mb / Sound / Case ATX 300W / Keyboard	Каб	8 693.00	21.12.2011
16	LCD Acer AL1511	Дягномаль 15"	Каб	7 090.00	16.12.2011
29	Nokia 6100	QSPM00/1800/1900, 7Мв, вібродвигун, фотофункція, камерний джойстик, Wi-Fi, GPS	Каб	1 000.00	07.09.2011
				16 783.00	

## Зведена таблиця порівняння аналогів

Назва	Microsoft Excel	Eqman	IT Invent
Наявність безкоштовної версії	+	-	+/- (до 200 облікових одиниць)
Легкість в освоєнні інтерфейсу	+	-	-
Можливість формувати специфічні звіти	+/-	-	-
	Формування звітів потребує глибокого знання функцій Excel та важке для звичайного користувача	тільки шаблонні звіти	тільки шаблонні звіти
Можливість експорту даних	+	+	+
Наявність засобів пошуку, фільтрації, вибірки	+	+	+
Наявність засобів для обмеження доступу до інформації	+/-	+	+
	обмежений доступ до окремих комірок/аркушів з використанням пароля		

3

## МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

- **Мета роботи:** спростити облік обладнання в навчальних лабораторіях кафедр Державного університету телекомунікацій за рахунок впровадження програмного забезпечення
- **Об'єкт дослідження:** процес обліку обладнання в навчальних лабораторіях кафедр Державного університету телекомунікацій
- **Предмет дослідження:** програмні засоби для обліку обладнання в навчальних лабораторіях

4

## Приклад паспорту кабінету в Державному Університеті Телекомунікацій (Microsoft Word)

**Паспорт кабінету**  
(лабораторії, аудиторії, викладацької, деканат, та інше.)

Найменування : лабораторія  
(лабораторії, аудиторії, викладацької, деканат, та інше.)

Кому належить: кафедра Інженерії програмного забезпечення ННІПТ  
(найменування кафедри, інституту)

Місце розташування кабінету: поверх 3, № кабінету 310  
(поверх, № кабінету)

Загальна площа кабінету (лабораторії, аудиторії, викладацької, деканат, та інше.) 47,9 м<sup>2</sup>.

Кількість вікон 3 шт.

Меблі та обладнання:

столи <u>25</u> шт.,	магнітна дошка <u>1</u> шт.,
робочий стіл викладача <u>1</u> шт.,	проектор <u>1</u> шт.,
дошка <u>1</u> шт.,	жалюзі <u>3</u> шт.,
екран <u>1</u> шт.,	вогнегасники <u>2</u> шт.,
комп'ютер <u>25</u> шт.,	плафони освітлення <u>15</u> шт.,
шафа <u>1</u> шт.,	розетки електричні <u>48</u> шт.,
стілці для студентів <u>32</u> шт.	розетки Інтернет( і що стосується Інтернету) <u>1</u> шт.,

стілці викладача, інженера лабораторії 1 шт.,  
кондиціонер 1 шт.;

Орієнтовний перелік навчально-наочних посібників і навчального обладнання:

№	Назва	Кількість і місце знаходження	Інвентарний №
1	<u>Відеопроєктор</u>	<u>Acer X 112 H (MR/JKV11.001)</u>	1
2	Дошка	магнітно-маркерна ALU23	1
3	Жалюзі вертикальні	<u>Line 1310</u>	3
4	Колонки	<u>Microlab 2.0 FC30 White</u>	1
5	Комп'ютер	<u>HPt5630w Compaq</u>	25

5

## ТЕХНІЧНЕ ЗАВДАННЯ

1. Інтерфейс системи повинен забезпечувати ключові операції обліку обладнання навчальних лабораторій кафедр в Державному університеті телекомунікацій.
2. База даних системи повинна містити інформацію про обладнання та програмне забезпечення, яке знаходиться в лабораторіях, та інші дані відповідно до паспортів кабінетів.
3. Програма повинна забезпечувати можливість вибірки необхідної інформації з бази даних паспортів кабінетів, в тому числі, експорт даних в Microsoft Word.
4. Програма повинна забезпечувати можливість зміни інформації щодо неактуальних даних, їх видалення та перенесення.

6

## ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ

**phpMyAdmin** — веб-додаток з відкритим кодом, написаний мовою PHP і що є веб-інтерфейсом для адміністрування СУБД MySQL



**OpenServer** - використовується з метою розробки, налагодження та тестування веб-проектів, а також для надання веб-сервісів у локальних мережах.



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

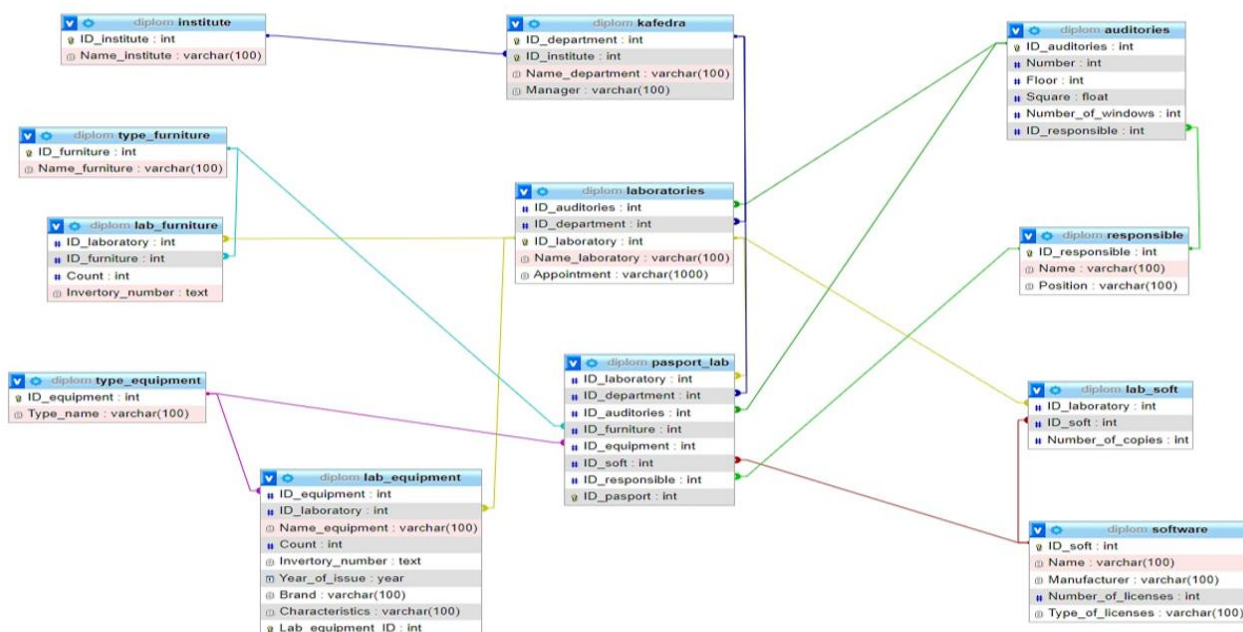


**MySQL** — вільна система керування реляційними базами даних, яка була розроблена компанією «ТсХ» для підвищення швидкодії обробки великих баз даних.



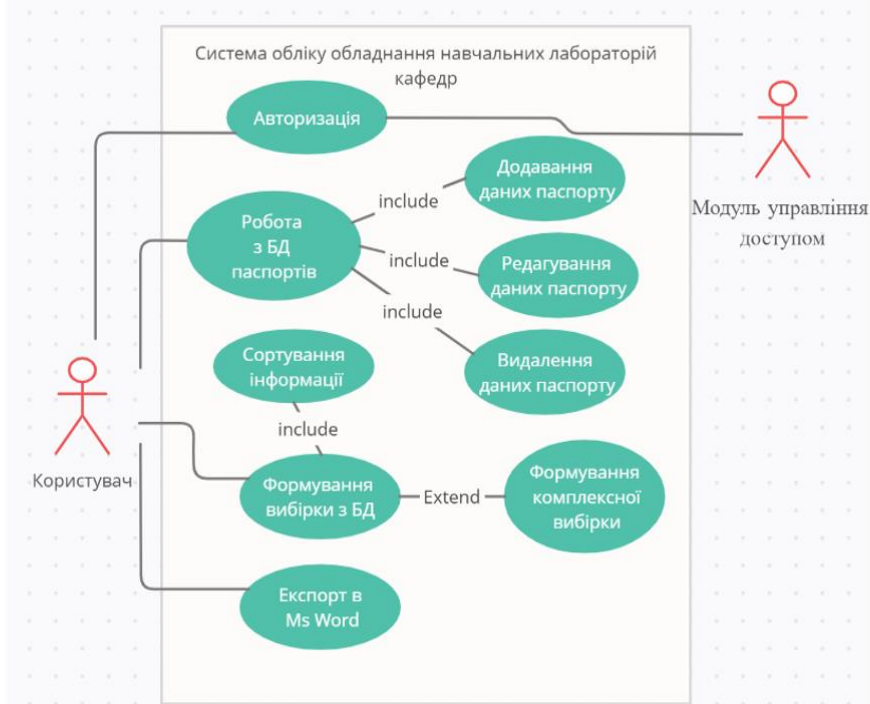
7

## СТРУКТУРА РЕЛЯЦІЙНОЇ БАЗИ ДАНИХ



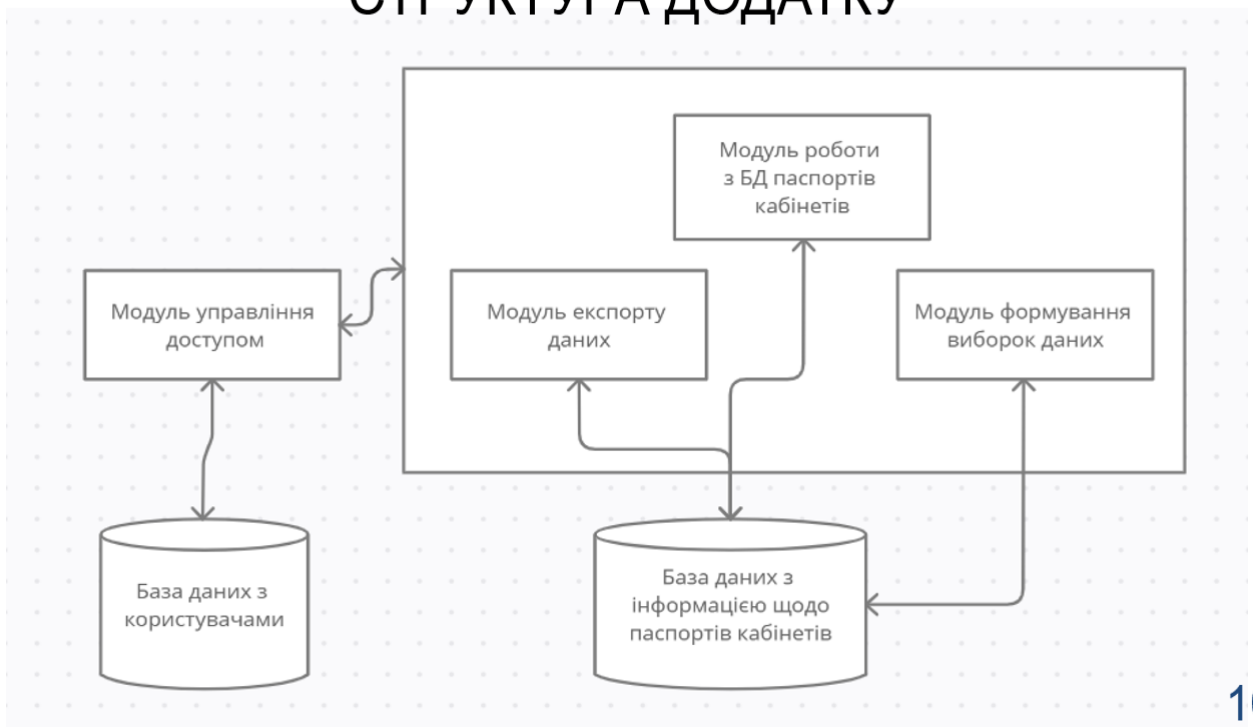
8

## ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ



9

## СТРУКТУРА ДОДАТКУ



10

## ПЕРЕВІРКА КОРИСТУВАЧА НА НАЯВНІСТЬ В БД

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
1	id	int		UNSIGNED	Нет	Нет		AUTO_INCREMENT	
2	login	varchar(100)	utf8_general_ci		Нет	Нет			
3	pass	varchar(32)	utf8_general_ci		Нет	Нет			

```

DB db = new DB();
DataTable table = new DataTable();
MySQLDataAdapter adapter = new MySQLDataAdapter();
MySQLCommand command = new MySQLCommand("SELECT * FROM 'users' WHERE 'login' = @u AND 'pass' = @p", db.getConnection());
command.Parameters.Add("@u", DbType.VarChar).Value = loginUser;
command.Parameters.Add("@p", DbType.VarChar).Value = passUser;
adapter.SelectCommand = command;
adapter.Fill(table);

if (table.Rows.Count > 0)
{
    Choice s = new Choice();
    s.Show();
    s.Hide();
}
else
    MessageBox.Show("Невірний логін або пароль.");

```

11

## ЗАГАЛЬНА БД З ВСІЄЮ ІНФОРМАЦІЄЮ

Назва лабораторії	Назва програми	Кількість копій
Лабораторія EPAM та ProjectManagement	OC Windows 7	30
Лабораторія EPAM та ProjectManagement	MS Office 2010	30
Лабораторія EPAM та ProjectManagement	Cache2015	30
Лабораторія EPAM та ProjectManagement	Delphi XE10	30
Лабораторія EPAM та ProjectManagement	MatCAD 14	30
Лабораторія EPAM та ProjectManagement	MicroCAP 11	30
Лабораторія EPAM та ProjectManagement	PHP	30
Лабораторія EPAM та ProjectManagement	Rational Rose	30
Лабораторія EPAM та ProjectManagement	C#	30
Лабораторія EPAM та ProjectManagement	CSS	30
Технології «Блок-чейн»	UbuntuLinux 15.11	20
Технології «Блок-чейн»	Eclipse Java Mars	20
Технології «Блок-чейн»	Office365	20
Технології «Блок-чейн»	MozillaFirefox	20
Технології «Блок-чейн»	Intelly IDEA	20
Технології «Блок-чейн»	Python	20
Технології «Блок-чейн»	OC Windows 7	20
Технології «Блок-чейн»	MS Office 2010	20
Технології «Блок-чейн»	Cache2015	20
Технології «Блок-чейн»	C#	20
Технології «Блок-чейн»	Delphi XE10	20
Технології «Блок-чейн»	MatCAD 14	20
Технології «Блок-чейн»	MicroCAP 11	20
Технології «Блок-чейн»	PHP	20
Технології «Блок-чейн»	Rational Rose	20
Лабораторія мов програмування	UbuntuLinux 15.11	21
Лабораторія мов програмування	Eclipse Java Mars	21
Лабораторія мов програмування	Office365	21
Лабораторія мов програмування	MozillaFirefox	21
Лабораторія мов програмування	Intelly IDEA	21

12

## КОМПЛЕКСНА ВИБІРКА ДАНИХ З ТАБЛИЦЬ

Назва лабораторії	Назва кафедри	Номер аудиторії	Поверх	Площа	Меблі	Обладнання	Програмне забезпечення	Відповідальний
Лабораторія ЕРАМ та ProjectManagement	Інженерія Програмного Забезпечення	310	3	47,9	Стіл комп'ютерний, Дошка магнітно-маркерна ALU23, Жалози вертикальні Line 1310, Кріплення для проектора Vrateek PRB-2 Silver, Стілець для навчання, Стіл учбовий для забезпечення учбового процесу	Відеопроєктор, Колонки, Комп'ютер, Кондиціонер, Монітор, Проекційний екран, Сплітер	OS Windows 7, MS Office 2010, Cache2015, Delphi XE10, MatCAD 14, MicroCAP 11, PHP, Rational Rose, C#, CSS	А.Б.Коба
Технології «Блок-чейн»	Інженерія Програмного Забезпечення	325	3	50,1	Кріплення для проектора Vrateek PRB-2 Silver, Дошка магнітно-маркерна ALU23, Стілець для навчання, Стіл комп'ютерний, Жалози вертикальні Line 1310	Відеопроєктор, Комп'ютер, Монітор, Кондиціонер, Точка доступу Wi-Fi	UbuntuLinux 15.11, Eclipse Java Mars, Office365, MozillaFirefox, IntelliJ IDEA, Python, OC Windows 7, MS Office 2010, Cache2015, C#, Delphi XE10, MatCAD 14, MicroCAP 11, PHP, Rational Rose	А.Б.Коба
Лабораторія мов програмування	Інженерія Програмного Забезпечення	326	3	44,7	Жалози вертикальні 1750 x 2350, Кріплення для проектора Suppe PR002, Дошка TSA-1510 для письма маркером, Стіл комп'ютерний для учбової лабораторії, Стіл для учбової лабораторії, Шафа для учбової лабораторії, Шафа інструментальна ШИ-15	Проекційний екран, Колонки, Кондиціонер, Мережевий фільтр, Монітор, Принтер, Комп'ютер, Сплітер	UbuntuLinux 15.11, Eclipse Java Mars, Office365, MozillaFirefox, IntelliJ IDEA	А.Б.Коба

13

## АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

1. Сидоренко В.О., Золотухіна О.А. Розробка програмного забезпечення для обліку обладнання навчальних лабораторій кафедр засобами мови C# / Науково-технічна конференція «Застосування програмного забезпечення в ІКТ», м. Київ, ДУТ, 20 квітня 2022 року.

14



## ВИСНОВКИ

1. Проведено аналіз структури та форматів документації, яка використовується для обліку обладнання в навчальних лабораторіях кафедр Державного університету телекомунікацій. Визначено недоліки існуючого обліку із використанням Microsoft Word.
2. Виконано огляд програмного забезпечення, яке може бути застосовано для обліку обладнання в навчальних лабораторіях кафедр, визначено особливості роботи програм, їх недоліки та переваги. Зазначена інформація використана при проектування та розробці нового програмного продукту.
3. Спроектовано та реалізовано реляційну базу даних для збереження інформації про обладнання та програмне забезпечення, яке знаходиться в лабораторіях, та інші дані відповідно до паспортів кабінетів.
4. Спроектовано та реалізовано інтерфейс системи, який забезпечує ключові операції обліку обладнання навчальних лабораторій кафедр, в тому числі вибірку необхідної інформації з бази даних паспортів кабінетів, експорт даних в Microsoft Word, підтримка актуальної інформації в БД.

### **Перспективи розвитку системи:**

- інтеграція з бухгалтерською системою Державного університету телекомунікацій;
- розширення переліку звітів та зведених таблиць, які генерує система.

**ДЯКУЮ ЗА УВАГУ!**

