

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Комп'ютеризована система обліку успішності та відвідування занять
здобувачами освіти

Виконав(ла): студент(ка) IV курсу, групи СІс-43

спеціальності 123 «Комп'ютерна інженерія»

(шифр і назва спеціальності)

(підпис)

Дзюбак Ю.Ю.

(прізвище та ініціали)

Керівник

(підпис)

Луцик Н.С.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Тши Є.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Осухівська Г.М.

(прізвище та ініціали)

Рецензент

(підпис)

(прізвище та ініціали)

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

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

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

ЗАТВЕРДЖУЮ

Завідувач кафедри

Осухівська Г.М.

(підпис)

(прізвище та ініціали)

« ____ » _____ 2022 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня бакалавр
(назва освітнього ступеня)

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

студенту Дзюбаку Юрію Юрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Комп'ютеризована система обліку успішності та відвідування занять здобувачами освіти

Керівник роботи Луцик Надія Степанівна, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 23 » 03 2022 року № 4/7-180

2. Термін подання студентом завершеної роботи 22.06.2022 р.

3. Вихідні дані до роботи Інформація про особливості ведення навчального процесу у коледжі (НЕМК), апаратне забезпечення для сервера комп'ютеризованої системи, програмне забезпечення для розробки комп'ютеризованої системи

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

Вступ. 1. Сучасні методи автоматизації контролю навчального процесу. 2. Проектування комп'ютеризованої системи "Класний журнал". 3. Реалізація проекту. 4. Безпека життєдіяльності, основи охорони праці. Висновки

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

1. Логічна модель спроектованої бази даних.

2. EER-діаграма створеної бази даних.

3. Структура модулів програмної реалізації комп'ютеризованої системи.

4. Алгоритм роботи програмного забезпечення комп'ютеризованої системи.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
<i>Безпека життєдіяльності, основи охорони праці</i>	<i>Лазарюк В.В., асистент каф. МТ</i>		

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	<i>Розробка та аналіз вимог технічного завдання</i>	<i>23.03-31.03</i>	
2	<i>Аналіз організаційної структури закладу освіти</i>	<i>25.03-02.04</i>	
3	<i>Проектування схеми бази даних та архітектури програмного забезпечення</i>	<i>02.04-14.04</i>	
4	<i>Розробка програмного забезпечення комп'ютеризованої системи</i>	<i>14.04-15.05</i>	
5	<i>Налаштування апаратної складової комп'ютеризованої системи</i>	<i>16.05-20.05</i>	
6	<i>Тестування і виправлення помилок комп'ютеризованої системи</i>	<i>20.05-01.06</i>	
7	<i>Безпека життєдіяльності, основи охорони праці</i>	<i>02.06-06.06</i>	
8	<i>Оформлення кваліфікаційної роботи</i>	<i>06.06-12.06</i>	
9	<i>Попередній захист кваліфікаційної роботи</i>	<i>12.06-17.06</i>	
10	<i>Захист кваліфікаційної роботи</i>	<i>22.06-24.06</i>	

Студент

_____ (підпис)

Дзюбак Ю.Ю.

_____ (прізвище та ініціали)

Керівник роботи

_____ (підпис)

Луцик Н.С.

_____ (прізвище та ініціали)

АНОТАЦІЯ

Комп'ютеризована система обліку успішності та відвідування занять здобувачами освіти // Кваліфікаційна робота на здобуття освітнього ступеня бакалавр // Дзюбак Юрій Юрійович // ТНТУ, спеціальність 123 «Комп'ютерна інженерія»// Тернопіль, 2022 // с. – 69, рис. – 51, табл. – 3, аркушів А1 – 4, бібліогр. – 11.

Ключові слова: електронний журнал, комп'ютеризована система, сервер, crud-застосунок, java, javafx, ui/ux дизайн, база даних, mysql, pdf-формат.

У кваліфікаційній роботі розроблено проект комп'ютеризованої системи обліку успішності та відвідування занять здобувачами освіти до складу якої входять серверна частина та програмне забезпечення управління процесом контролю обліку успішності та відвідування занять.

Серверна частина побудована на основі фізичного сервера HP ProLiant DL160 G6 під керуванням операційної системи Windows Server 2012 R2 Standard Server. Забезпечення зв'язку між робочими станціями і сервером бази даних реалізовано через існуючу локальну мережу навчального закладу за допомогою базових протоколів передачі даних використовуваних інструментів розробки прикладного застосунку.

Керування процесом контролю та обліку навчальних занять забезпечено на рівні програмного забезпечення, розробленого на основі архітектури "клієнт- сервер" засобами мови програмування Java. База даних побудована на основі реляційного підходу та з використанням системи керування базами даних MySQL Server.

ABSTRACT

Computerized system of accounting for performance and attendance of students // Bachelor's thesis // Dziubak Yuriy Yuriyovych // TNTU, speciality 123 «Computer engineering»// Ternopil, 2022 // p. – 69, fig. – 51, tab. – 3, posters A1 – 4, ref. – 11.

Keywords: digital classbook, computerized system, server, crud-application, java, javafx, ui/ux design, database, mysql, pdf-format.

In the qualification work, a project of the computerized system of the of accounting for performance and attendance of students, which structure includes a server part and the software for management of process for control of the account performance and attendance is developed.

The server part is based on the physical server HP ProLiant DL160 G6 running Windows Server 2012 R2 Standard Server. Communication between workstations and the database server is implemented through the existing local network of the educational institution using the basic data transmission protocols of the tools used to develop the application.

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

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1 СУЧАСНІ МЕТОДИ АВТОМАТИЗАЦІЇ КОНТРОЛЮ НАВЧАЛЬНОГО ПРОЦЕСУ	10
1.1 Організація навчального процесу з використанням електронного журналу..	10
1.2 Огляд сучасних програмних засобів для розв’язання задачі організації ведення електронного журналу	11
1.3 Обґрунтування доцільності створення комп’ютеризованої системи «Класний журнал» під потреби окремого вищого навчального закладу	14
РОЗДІЛ 2 ПРОЕКТУВАННЯ КОМП’ЮТЕРИЗОВАНОЇ СИСТЕМИ «КЛАСНИЙ ЖУРНАЛ».....	18
2.1 Аналіз предметної області та виділення інформаційних об’єктів	18
2.2 Побудова логічної моделі та проектування реляційної бази даних	21
2.3 Вибір системи управління базами даних, середовища розробки програмної частини.....	25
2.4 Вибір програмно-апаратних рішень для реалізації серверної частини	28
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОЕКТУ	32
3.2 Створення бази даних	32
3.2 Опис методів та ресурсів для розробки комп’ютеризованої системи	34
3.3 Розробка інтерфейсів програми для користувачів системи	38
3.4 Реалізація функції друку сторінок журналу на основі заповнених даних	48
3.5 Збірка та компіляція програми у виконуваний файл.....	51

					КС КРБ 123.213.00.00 ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	Комп’ютеризована система обліку успішності та відвідування занять здобувачами освіти	Літ.	Арк.	Аркушів
Розроб.		Дзюбак Ю.Ю.					6	4
Перевір.		Луцик Н.С.				ТНТУ, каф. КС, гр. СІс-43		
Реценз.								
Н. Контр.		Тиш Є.В.						
Затверд.		Осухівська Г.М.						

3.6 Встановлення спеціалізованої ОС на виділений сервер та налаштування на ньому СУБД.....	53
3.7 Тестування та налагодження комп'ютеризованої системи	56
РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	59
4.1 Працездатність людини – оператора	59
4.2 Вимоги ергономіки до організації робочого місця оператора ПК.....	62
ВИСНОВКИ.....	67
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	68
Додаток А. Технічне завдання	
Додаток Б. Лістинги коду основних функцій програми	

					КС КРБ 123.213.00.00 ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Сучасні комп'ютеризовано-інформаційні технології зараз є актуальними в усіх видах людської діяльності, в тому числі і в освітньому просторі. Розвинені комп'ютеризовані технології та ефективне їх використання значно впливають на будь-якої виробничо-професійної діяльності. Особливою тут є роль сфери освіти, що формує інтелектуальний потенціал країни, що використовує і примножує її інформаційні ресурси.

Комп'ютеризація (або «діджиталізація», англ. digitalization) – це комплекс робіт, що розробляє, впроваджує, обслуговує, вдосконалює і змінює видах людської діяльності традиційні технології на ефективніші комп'ютеризовані технології.

Впровадження КС (комп'ютеризованих систем) спрямовано на вирішення завдань пов'язаних із повним циклом ведення діловодства, зменшення частини паперового (традиційного) документообігу та збільшення ефективності аналітичних робіт та зменшення часу їх виконання.

Сучасні інструменти створення програмного забезпечення (ПЗ) дозволяють створити потужні, ефективні, швидкі та «дружелюбні до користувача» (англ. user-friendly interface) застосунки для вирішення, фактично, будь-яких задач.

Однією з систем, котрі часто впроваджуються у навчальний процес, є електронний класний журнал успішності та відвідуваності, який діджиталізує процес контролю за успішністю та відвідуваністю, створюючи копію записів традиційного паперового журналу, захищаючи його від помилок заповнення, даючи тримати під контролем процес оцінювання та факти відвідування занять, а також забезпечує можливість проведення заходів аналітичного спрямування з отриманими даними.

Метою даного проекту є побудова комп'ютеризованої системи «Класний журнал» для ДВНЗ «Нововолинський електромеханічний коледж», що передбачає розробку бази даних та застосунку для діджиталізації процесу

					КС КРБ 123.213.00.00 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

ведення класного журналу.

Основними завданнями проекту є:

- 1) дослідження особливостей організації обліку процесу оцінювання та відвідуваності у вищому навчальному закладі;
- 2) інформаційний аналіз предметної області та виділення інформаційних об'єктів проекту;
- 2) побудова логічної моделі даних системи;
- 3) проектування реляційної бази даних;
- 4) вибір СУБД;
- 5) вибір середовища розробки застосунків (IDE);
- 6) розробка застосунку;
- 7) тестування системи, внесення корективів;
- 8) створення документації до КС щодо порядку організації роботи з нею;
- 9) кінцеве налаштування та введення в експлуатацію (за ініціативи керівництва).

Оскільки, ця комп'ютеризовано-інформаційна система буде розроблена конкретно для потреб ДВНЗ «Нововолинський електромеханічний коледж», то її впровадження у навчальну структуру надасть низку переваг її використання, таких як:

- конкретний необхідний набір функціоналу;
- можливість зміни та модифікації системи залежно від змінних потреб у питаннях організації навчального процесу.

					КС КРБ 123.213.00.00 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1 СУЧАСНІ МЕТОДИ АВТОМАТИЗАЦІЇ КОНТРОЛЮ НАВЧАЛЬНОГО ПРОЦЕСУ

1.1 Організація навчального процесу з використанням електронного журналу

Будь-який заклад освіти має великий обсяг документообігу. Викладачі кожного дня витрачають час на заповнення класного журналу.

Високий рівень комп'ютеризовано-інформаційних технологій сьогодні, надає все більшій кількості електронної документації набути офіційного статусу, витісняючи фізичні джерела. Тому, комп'ютеризовано-інформаційні технології дозволяють реалізувати систему додаткового контролю успішності та відвідуваності здобувачів освіти, діджиталізуючи даний процес в навчальному закладі. Одним із основних документів, що ведеться у закладах освіти є класний журнал.

Класний журнал – основний державний документ реєстрації відвідуваності і успішності учнів та студентів в усіх закладах освіти. Ведення окремого журналу для кожного класу або групи кожним вчителем/викладачем обов'язково. Класний журнал є його аналогом у віртуальному просторі, що легко може набути фізичного втілення

Класний електронний журнал – це комп'ютеризовано-інформаційний інструмент, що діджиталізує процес ведення традиційного класного журналу.

Так як класний електронний журнал в ідеалі виступає повноцінною комп'ютеризовано-інформаційною системою, а комп'ютеризовано-інформаційна система є сукупністю рішень для обробки та збереження інформації, щоб забезпечення інформаційні потреби користувачів, то всю інформацію про

					КС КРБ 123.213.00.00 ПЗ			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		<i>Дзюбак Ю.Ю.</i>			<i>Сучасні методи автоматизації контролю навчального процесу</i>	<i>Лім.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Перевір.</i>		<i>Луцик Н.С.</i>					10	8
<i>Реценз.</i>						<i>ТНТУ, каф. КС, гр. СІс-43</i>		
<i>Н. Контр.</i>		<i>Тиш Є.В.</i>						
<i>Затверд.</i>		<i>Осухівська Г.М.</i>						

навчальний процес досить заповнити один раз, а в майбутньому лише доповнювати або виправляти ці дані при необхідності і використовувати їх за потреби.

1.2 Огляд сучасних програмних засобів для розв'язання задачі організації ведення електронного журналу

На комерційному рівні є велика кількість варіантів реалізації схожих систем. В західних країнах діджиталізація навчального процесу є розвинутішою, чим в навчальних закладах України. Причиною є тільки початок розвитку інформаційних технологій нашої країни, у тому числі через гостру необхідність віддаленої освіти. Це і стимулює діджиталізацію освітньої діяльності, введення комп'ютерних інтерфейсів доступу до інформаційних систем.

Оскільки, в Україні ступінь розвитку діджиталізації освітньої діяльності тільки починає набирати обертів, тому кількість подібних сервісів ще не така значна, як за кордоном. При цьому, багато установ, які вводять дистанційне навчання і сервіси автоматизації навчального процесу, є відділеннями зарубіжних інститутів і середніх навчальних закладів.

З огляду на виклики та нагальні потреби, які постали перед системою освіти України в умовах протидії пандемії COVID-19 та військовим часом, а також, з метою покращення управління системою освіти, автоматизоване ведення документообігу закладів освіти Міністерством освіти і науки України було поставлено завдання ДНУ «Інститут освітньої аналітики» (наказ МОН № 781 від 09.06.2020) здійснити розробку державних безкоштовних електронних журналів та щоденників Electronic Journal [1].

До подібних реалізацій в Україні, також, варто віднести програмне забезпечення котре розробляється приватними компаніями. Наприклад, як кордоном так і у нашій країні доволі популярною є освітня система Moodle [2] – це навчальна платформа об'єднання всіх учасників навчального процесу в

					КС КРБ 123.213.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

спеціалізовану систему, що утворює своєрідне навчальне середовище, у якій існують свої реалізації електронних класних журналів.

Іншими вдалими комерційними реалізаціями КС «Класний журнал» в Україні є:

– «Електронні щоденники та журнали» на платформі e-schools.info [3] – сервіс від ТОВ «Київська Енергетична Агенція» за підтримки Міністерства освіти і науки України надає установам освіти України можливість зробити впевнений крок вперед, почавши використовувати сучасний сервіс електронних щоденників і журналів (рис. 1.1);

Тема уроку	Вересень																															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
1. Заново Максим	7	8,43	7				11					7																				
2. Заново Юлія	4	9	10	10								7																				
3. Байдак Микола	8	8,2	8					8				8																				
4. Баранів Ганна	4	8,75	8	8																												
5. Богданів Олександр	7	8,27	8	7				8	10																							
6. Богданів Анастас	7	7,43	8	8																												
7. Боравіло Олександр	8	8,38	8																													
8. Гонца Катерина	11	8,24	8	8																												
9. Горіна Карина	8	8,28	8	8																												
10. Григорський Віталій	10	8,2	8	8																												
11. Дубіш Алла	7	8	8	7																												
12. Жовтківська Валентина	8	8,33	8	8																												
13. Жолтенко Сергій	7	8,29	8	8																												
14. Пилипівська Світлана	8	8,38	8	8																												
15. Навин Роман	8	7,21	8	8																												

Рисунок 1.1 – Приклад сторінки журналу успішності учнів від E-schools.info

– проект «Щоденник» від ТОВ «Щоденник» [4] – всеукраїнська безкоштовна освітня мережа «Щоденник.ua» – це унікальне електронне середовище для вчителів, учнів та їх батьків. Щоденник надає різноманітні функції для школярів, учителів і батьків (рис. 1.2);



Рисунок 1.2 – Приклад сторінки електронного журналу від «Щоденник.ua»

– сайт ukrschools.com.ua із можливостями порталу – комерційний проект (річна підписка), створений для всіх учасників навчального процесу. Візуал системи максимально схожий до традиційних класного журналу та щоденника. Всі розділи «Електронного журналу» зручні та інтуїтивно зрозумілі (рис. 1.3).



Рисунок 1.3 – Інструменти електронного журналу від ukrschools.ua

Вищі начальні заклади в Україні зазвичай не практикують впровадженнь комерційних програмних продуктів, а впроваджують власні або замовні розробки, інформація про опис чи структуру яких не попадає у загальний доступ.

Змн.	Арк.	№ докум.	Підпис	Дата

Ось декілька ВНЗ, де впроваджені подібні рішення:

- Київський національний економічний університет [5];
- Буковинський державний медичний університет [6];
- Національної академії Державної прикордонної служби України імені Богдана Хмельницького [7];
- Національний педагогічний університет імені М.П. Драгоманова [8].

1.3 Обґрунтування доцільності створення комп'ютеризованої системи «Класний журнал» під потреби окремого вищого навчального закладу

Доцільність створення системи такого плану під потреби окремого, конкретно виділеного, вищого навчального закладу має дві сторони – сторону переваг створення такої системи і, водночас, сторону проблем, котрі необхідно буде вирішити задля того, аби така система могла з'явитися та успішно функціонувати.

Переваги створення КС для конкретного вищого навчального закладу:

- система, на відміну від універсальних аналогів, буде побудована конкретно за побажаннями замовників, з урахуванням усіх необхідних особливостей у своїй структурі та функціоналі;
- система буде унікальною;
- створення такої системи студентами, котрі і навчаються у навчальному закладі, для якого буде створюватися дана КС, або хоча б їх залучення до створення такого продукту у певній мірі підвищить престижність даного навчального закладу та їх мотивацію та конкурентоздатність на ринку праці;
- у разі створення системи власними зусиллями на її розробку піде мінімум матеріальних ресурсів;
- створену систему можна буде вдосконалювати та функціонально доповнювати власними силами;

					КС КРБ 123.213.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

– створену систему можна буде зробити універсальною або модифікувати під конкретні потреби іншого навчального закладу і у такому випадку, ймовірно, її можна буде комерціалізувати.

Недоліки (проблеми) створення КС для конкретного вищого навчального закладу:

- складність створення подібної системи власними силами;
- великі затрати часу при створенні подібної системи власними силами;
- при замовленні подібної унікальної системи у професіоналів головним недоліком буде ціна, бо якісний унікальний продукт з супроводом відповідно коштує;
- вірогідність невдачі та недостатньої досконалості створеної системи власними силами (відсутність досвіду та професійності у виконавців) або навіть при замовленні у професіоналів;
- значні витрати часу та зусиль на налагодження системи та її тестування.

Проте, зважаючи на особливості навчального процесу в кожному вищому навчальному закладі (зважаючи на отриману самостійність), все ж доцільним є створення власного програмного продукту із можливістю наступної модифікації та адаптації до змінних потреб.

Відповідно, реалізація проекту КС «Класний журнал» для конкретного навчального закладу втілюється в рамках конкретного призначення, цілей та завдань.

Призначення проекту КС «Класний журнал» – ведення автоматизованого обліку успішності та відвідуваності здобувачів освіти. КС призначена для виконання наступних функцій:

- введення і редагування даних успішності студентів;
- введення і редагування даних відвідуваності студентами занять;
- введення і редагування списків студентів, груп, викладачів, предметів та інших інформаційних об'єктів системи;

					КС КРБ 123.213.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

- оперативний доступ до необхідної інформації щодо організації навчального процесу у розрізі «Класного журналу»;
- можливість формування необхідної аналітичної інформації у розрізі «Класного журналу»;
- можливість формування фізичного журналу на основі даних введених у систему «Класного журналу».

Комп'ютеризовану систему «Класний журнал» передбачається використовувати у ДВНЗ «Нововолинський електромеханічний коледж» з орієнтацію на забезпечення максимальної універсальності щодо організації навчального процесу у подібних вищих навчальних закладах.

Основними цілями створення КС є:

- підвищення рівня ефективності виконання процесів організації навчального процесу шляхом скорочення непродуктивних і дублюючих операцій та операцій, що виконуються «вручну»;
- забезпечення зручності введення, редагування, отримання, використання та відображення інформації;
- врахування особливостей організації навчального процесу та можливість забезпечення її адаптації за умов його зміни;
- пришвидшення пошуку інформації у системі обліку успішності та відвідуваності студентів шляхом її переведення на електронні основи;
- організація розрізненого доступу користувачів системи до визначеного функціоналу та відповідної інформації;
- забезпечення оперативного контролю адміністрацією організації та перебігу навчального процесу шляхом моніторингу інформаційної системи навчального закладу.

Кінцевою метою даного проекту є створення комп'ютеризованої системи «Класний журнал» для ДВНЗ «Нововолинський електромеханічний коледж», що передбачає розробку бази даних та застосунок для діджиталізації процесу ведення журналу успішності та відвідуваності здобувачів освіти.

					КС КРБ 123.213.00.00 ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

Для досягнення цієї мети необхідно розв'язати ряд взаємозалежних послідовних задач (етапів розробки проекту):

- 1) інформаційний аналіз предметної області та виділення інформаційних об'єктів проекту;
- 2) побудова логічної моделі даних системи;
- 3) проектування реляційної бази даних;
- 4) вибір СУБД;
- 5) вибір середовища розробки застосунків (IDE);
- 6) розробка застосунку;
- 7) тестування системи, внесення корективів;
- 8) створення документації до КС щодо порядку організації роботи з нею;
- 9) кінцеве налаштування та введення в експлуатацію (за ініціативи керівництва).

Дані кроки будуть реалізовані у наступних розділах роботи.

					КС КРБ 123.213.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

РОЗДІЛ 2 ПРОЕКТУВАННЯ КОМП'ЮТЕРИЗОВАНОЇ СИСТЕМИ «КЛАСНИЙ ЖУРНАЛ»

2.1 Аналіз предметної області та виділення інформаційних об'єктів

У даній роботі головним інформаційним об'єктом є класний журнал (або журнал групи) як основний документ обліку успішності та відвідуваності студентів. Таким чином, предметну область цієї сфери діяльності в рамках навчального процесу необхідно характеризувати особливостями інформаційних потоків зі своїми наборами об'єктів, що утворюють основу такої інформаційну систему як класний журнал.

Журнал обліку роботи академічної групи та викладачів (далі – журнал) – обов'язковий документ, в якому фіксуються результати навчальних досягнень студентів (учнів), відвідування ними занять, виконання навчальних програм тощо.

Наявність журналу на занятті обов'язкова. У позанавчальний час журнал зберігається в навчальній частині (денному відділенні).

Основу опису предметної області становлять вказівки щодо ведення обліку навчальних занять згідно виписки з Положення про ведення журналів навчальних занять ДВНЗ «НЕМК», зокрема:

1) Ведення журналу здійснюється викладачами коледжу та кураторами груп. Викладачі несуть відповідальність за правильність і достовірність записів;

2) Записи в журналах проводяться виключно державною мовою, з предметів іноземних мов можна частково записувати зміст заняття та домашнє завдання мовою, що вивчає предмет, з інформатично-комп'ютерних предметів – допускається варіант заповнення англійською мовою;

					КС КРБ 123.213.00.00 ПЗ			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		<i>Дзюбак Ю.Ю.</i>			<i>Проектування комп'ютеризованої системи «Класний журнал»</i>	<i>Лім.</i>	<i>Арк.</i>	<i>Аркуші</i>
<i>Перевір.</i>		<i>Луцик Н.С.</i>					18	14
<i>Реценз.</i>						<i>ТНТУ, каф. КС, гр. СІс-43</i>		
<i>Н. Контр.</i>		<i>Тиш Є.В.</i>						
<i>Затверд.</i>		<i>Осухівська Г.М.</i>						

3) Записи вносяться синім чорнилом акуратно і уважно. Будь-які виправлення на сторінках журналу проводити не можна, в разі помилки поряд вноситься виправлений варіант, що засвідчується потім підписом і скріплюється печаткою відповідним представником адміністрації;

4) Куратор групи заповнює розділ «Облік відвідування»;

5) Відсутність студента на занятті позначається літерою «н» і здійснюється в кінці заняття, у разі спізнення ставиться літера «с», якщо студент спізнився і отримав оцінку, то оцінка ставиться через риску дробу, наприклад «с/5». Якщо студент відпрацював пропущене заняття, у відповідну клітинку до літери «н» через дріб ставиться оцінка, наприклад, «н/5»;

6) Головними видами оцінювання навчальних досягнень здобувачів освіти є поточне та підсумкове (тематичне, модульне, семестрове, річне), державна підсумкова атестація, залік, екзамен. Тематичне оцінювання здійснюється у відповідності до робочих навчальних програм не рідше як двічі на семестр;

7) Незадовільна Семестрова оцінка, Залік, Екзамен може підлягати коригуванню. Скоригована оцінка виставляється без дати у колонку з надписом Скоригована поруч із колонкою I семестр, II семестр, Залік, Екзамен. Задовільна оцінка може бути скоригована тільки в особливих випадках і тільки з дозволу директора коледжу.

Основою журналу обліку навчальних занять є його зміст, іншими словами, навантаження групи на період заданого навчального року (рис. 2.1).

ЗМІСТ

№ з/п	Назва навчальної дисципліни	Години		Прізвище та ініціали викладача	Сторінки

Рисунок 2.1 – Сторінка «Зміст» журналу обліку навчальних занять

Згідно «Змісту» (навантаження групи) далі будуть формуватися сторінки дисциплін, котрі складаються з «правої» (рис. 2.2) та «лівої» частин (рис. 2.3).

На правій сторінці дисципліни ведеться облік успішності та відвідуваності студентів за проведеними заняттями, основними характеристиками яких є дата, вид та тема заняття.

Назва дисципліни _____
 Вид навчальних занять _____
(лекції, лабораторії та практики)

№ з/п	Місяць, число Прізвище та ініціали студента	Оцінка знань і відвідування																			

Рисунок 2.2 – «Права» частина сторінки дисципліни

Ці характеристики прописуються у «лівій» частині сторінки дисципліни (рис. 2.3).

Викладач _____
(прізвище та ініціали)

№ з/п	Дата	Кількість годин	Тема заняття	Домашнє завдання	Підпис викладача

Рисунок 2.3 – «Ліва» частина сторінки дисципліни

Згідно цих складових журналу обліку навчальних занять можна сформуванати список необхідних інформаційних об'єктів предметної області та перелік функцій котрі від цих об'єктів будуть прямозалежними, що необхідні для відтворення даного журналу у електронному виді (табл. 2.1).

Відповідно доцільним є розробити інтерфейси для адміністрації навчального закладу, викладачів, кураторів та студентів. Крім того, планується додавання можливості друку заповнених «сторінок журналу» і зшивання з них готових журналів.

Але, оскільки, основою інформаційної системи є викладацький інтерфейс та пов'язані з ним функції, то найбільше уваги тут буде приділено йому.

Таблиця 2.1 – Опис інформаційних об'єктів та функціональних блоків моделі предметної області

Список інформаційних об'єктів	<p>Викладачі Групи та підгрупи Студенти Дисципліни Види занять Навчальні роки Семестри Системи оцінювання Оцінки Відмітки відвідуваності Навантаження окремих груп Сторінки журналу (сформовані з навантаження занять та реалізовані облік успішності й відвідуваності студентів)</p>
Перелік функцій, де будуть використані інформаційні об'єкти	<p>Етап А: введення необхідних для ведення журналу довідкових даних адміністратором Етап Б: ведення викладачем журналу Б1: створення заняття викладачем згідно навантаження Б2: заповнення даних про заняття Б3: заповнення даних про відвідуваність та успішність заняття Етап В: використання введених даних (наочне відображення, друк і т. д.)</p>

2.2 Побудова логічної моделі та проектування реляційної бази даних

Відповідно до проаналізованих у попередньому підрозділі даних щодо

опису предметної області, виділених інформаційних об'єктів та функціональних блоків моделі предметної області проектується логічна модель інформаційної системи (рис. 2.4).

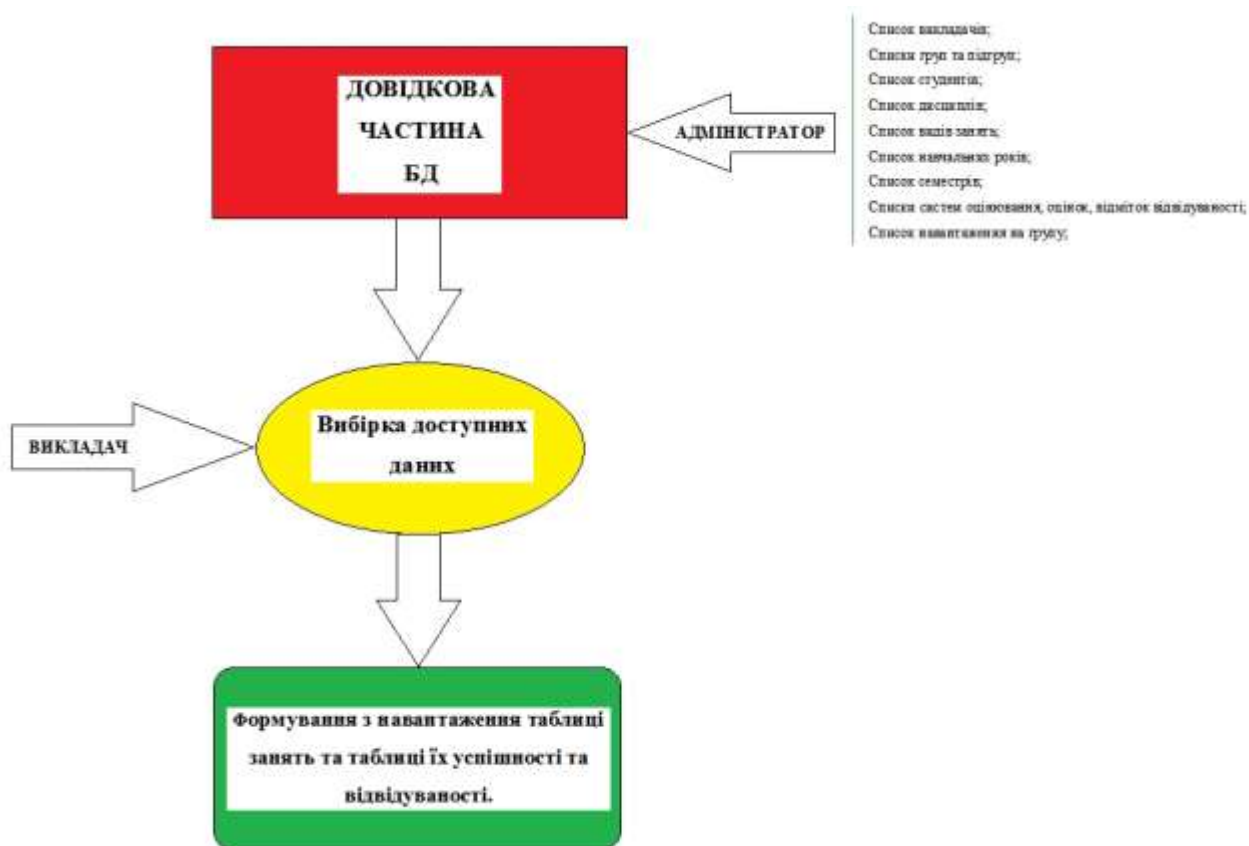


Рисунок 2.4 – Логічна модель інформаційної системи

Створене уявлення про інформаційні об'єкти та функціональні блоки моделі предметної області разом із продуманою логічною моделлю інформаційної системи дають можливість приступити до наступного етапу, а саме проектування реляційної бази даних.

За допомогою глибокого аналізу предметної області та дослідження ведення навчального процесу шляхом експериментів та проб було спроектовано модель бази даних у Microsoft Access 2010 (рис. 2.5).

Особливості предметної області та обмеження, що вплинули на проектування БД:

- 1) Студент є студентом лише однієї групи. Якщо одна й та ж людина

на паралель, вона є іншим студентом, тому знову вноситься в систему. Зумовлено це необхідністю чіткої ідентифікації студента за номером залікової книжки;

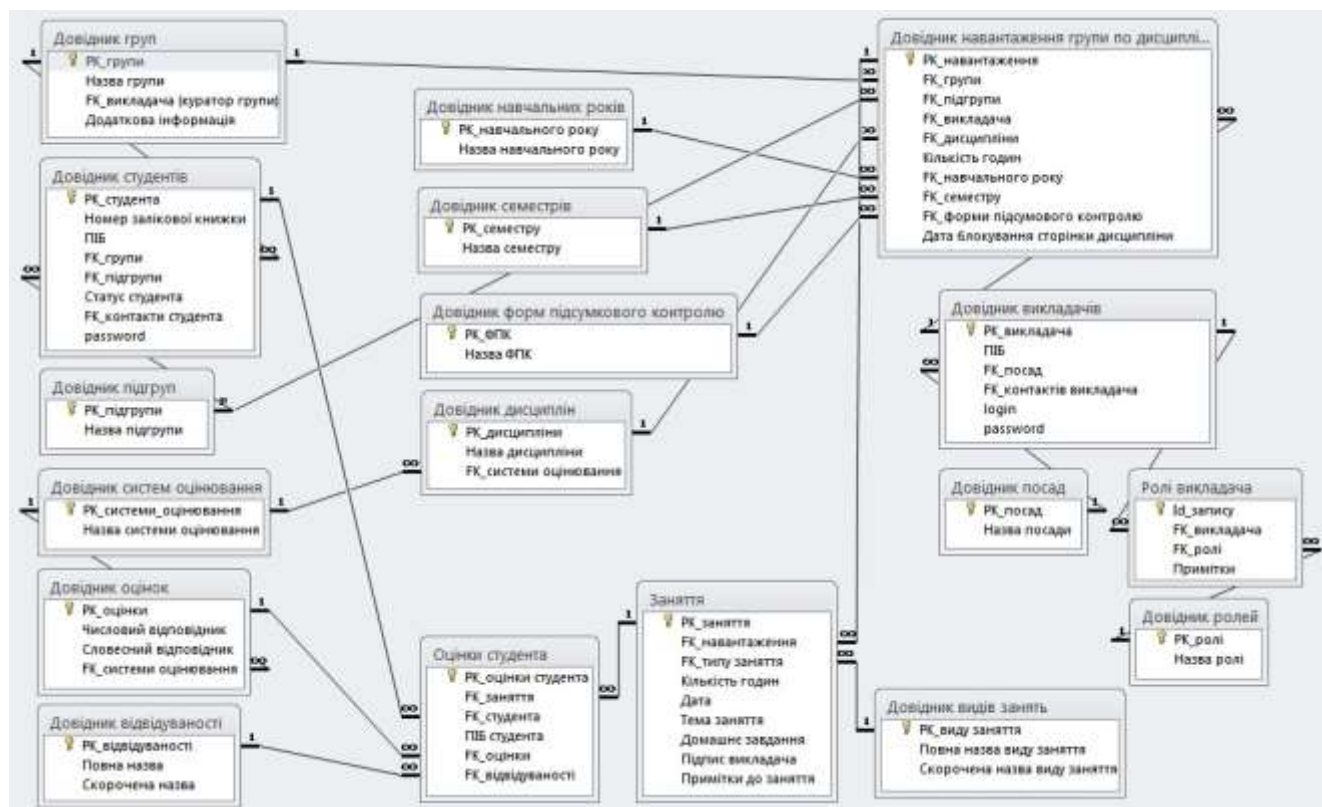


Рисунок 2.5 – Логічна модель бази даних

2) Оскільки у коледжі передбачений поділ студентів на підгрупи при їх кількості ≥ 15 осіб лише при проведенні ЛР, то доцільним є створення властивості «підгрупа», що характеризує приналежності студента до підгрупи згідно навантаження. Наприклад, усі студенти апіорі належать до групи 1-КТ-16, проте на дисципліні «Інформатика (ЛР)» за умови наявності ≥ 15 осіб, студенти належать також до підгруп 1 чи 2, що є додатковою властивістю студента. Відповідно для дисципліни «Інформатика» у навантаженні відсутньою (незаповненою) буде властивість «Підгрупа», а для дисципліни «Інформатика (ЛР)» за умови поділу студентів – властивість «Підгрупа» заповнюється;

3) У перспективі викладачі одночасно в системі можуть виконувати декілька ролей, залежно від ролі їм буде доступний той чи інший інтерфейс;

4) Студенти на дисциплінах можуть поділятися на підгрупи, тоді у журналі для цієї дисципліни виділяється така кількість сторінок, скільки підгруп у цій групі. Це призводить до того, що одну і ту ж дисципліну у рамках різних підгруп можуть вести різні викладачі. Водночас у разі тимчасової заміни викладача іншим для нього доступу до сторінки групи не буде. Це можна забезпечити з дозволу адміністрації. Помітка про проведення заняття іншим викладачем зазначається у комірці «Підпис», де буде вказуватися прізвище викладача, що провів заміну;

5) Дата блокування заповнення інформації по дисципліні повинна зазначатися індивідуально для кожної сторінки у журналі;

6) Вибір викладача згідно навантаження повинен проводитися для кожної «сторінки» у журналі;

7) Довідник навантаження групи за підгрупами використовується для відображення навчального плану через конкретне навантаження та для перевірки кількості годин для дисципліни у даній підгрупі групи із фактично вчитаними. Коли така кількість співпадає, з'являтиметься повідомлення про вчитку годин;

8) Якщо дисципліна викладається у двох і більше семестрах, то вона фактично приймається як дві різні дисципліни, тобто «Математика» з властивістю «I семестр» і «Математика» з властивістю «II семестр» (для цього існуватиме довідкова таблиця «Довідник семестрів»). Аналогічно окремими дисциплінами є «Бухгалтерський облік» та «Бухгалтерський облік (Курсова)» або «Інформатика» та «Інформатика (ЛР)» тощо;

9) Заняття може бути «тарифікованим» чи «нетарифікованим», відповідно за замовчуванням кількість годин кожного заняття 2, за потреби вноситься інше число (починається з нуля, якщо «нетарифіковане» заняття, типу «тематична», «скоригована» і може бути дробовим, адже на години екзамену впливає кількість студентів);

10) Будь-яка система оцінювання містить у своєму наборі ще додаткові дві універсальні оцінки «зарах.», «незарах.», адже окремі види виконуваних

					КС КРБ 123.213.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		24

робіт не обов'язково повинні бути «оцінені» у типовому розумінні цього слова;

11) Оскільки у БД буде міститися немаленька кількість довідкових таблиць, необхідно було вирішити проблему із видаленням із них непотрібних (неактуальних записів). Вирішити цю проблему допоможе рішення на кшталт рішень у корпоративних БД, тобто введенням у таблиці додаткового поля «existence» (з англ. «наявність», «актуальність»), яка характеризується властивістю «1» або «0» та означає, що об'єкт наявний (актуальний) або видалений (неактуальний).

2.3 Вибір системи управління базами даними та середовища розробки програмної частини

У ролі сервера баз даних був обраний MySQL Server 8.0, бо MySQL – є однією з найпопулярніших систем керування базами даних. Використовується для створення веб-сайтів та застосунків, має підтримку різноманітних мов програмування.

Відповідно, під час підбору СУБД вибір пав на MySQL Workbench, тому що це продукт корпорації Oracle, як і MySQL Server, тому вибір саме цієї СУБД гарантує повну сумісність та «розуміння» СУБД, сервера та самої бази даних. Крім того, розробник даного проекту має досвід користування саме MySQL Server та Workbench.

MySQL Workbench (рис. 2.6) – це уніфіковане ПЗ для архітекторів, розробників та адміністраторів баз даних. MySQL Workbench забезпечує моделювання даних, розробку на мові SQL та комплекс інструментів для адміністрування та налаштування сервера і його користувачів, а також, резервного копіювання та інше. MySQL Workbench доступний у Windows, Linux та Mac OS X.

Особливості та переваги MySQL Workbench:

– дозволяє адміністратору баз даних, розробнику або архітектору даних

					КС КРБ 123.213.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

візуально проектувати, моделювати, генерувати та керувати базами даних. Він включає все, що потрібно для створення складних моделей EER-діаграм, прямої та зворотної інженерії, а також забезпечує ключові функції для виконання складних завдань управління змінами та документацією, які зазвичай вимагають багато часу та зусиль;



Рисунок 2.6 – Головне вікно MySQL Workbench

– має зручні візуальні інструменти для створення, виконання та оптимізації запитів SQL. Редактор SQL здійснює виділення синтаксису кольорів, автоматичне завершення виразів, повторне використання фрагментів SQL та історію виконання SQL. Панель підключень до бази даних дозволяє ефективно керувати стандартними підключеннями до бази даних. Оглядач інформаційних об'єктів здійснює миттєвий доступ до схеми бази даних та інших об'єктів;

– включає візуальну консоль для легкого адміністрування середовищ MySQL та отримання кращої видимості в базах даних. Розробники та адміністратори баз даних можуть використовувати візуальні інструменти для налаштування серверів, адміністрування та налаштування сервера і його

користувачів, а також, резервного копіювання та перевірки стану баз даних;

– надає набір інструментів для підвищення продуктивності MySQL-додатків. Адміністратори баз даних можуть швидко переглядати основні показники ефективності за допомогою інформаційної панелі ефективності. Звіти ефективності забезпечують просту ідентифікацію та доступ до точок доступу вводу-виводу, значення операторів SQL тощо. Крім того, за допомогою одного кліку розробники можуть побачити, де оптимізувати свій запит за допомогою вдосконаленого та простого у використанні інструменту Visual Explain Plan;

– пропонує повне, просте у використанні рішення для міграції Microsoft SQL Server, Microsoft Access, Sybase ASE, PostgreSQL та інших таблиць, об'єктів і даних RDBMS в MySQL. Розробники та адміністратори баз даних можуть швидко та легко конвертувати існуючі програми для роботи на MySQL як на Windows, так і на інших платформах. Міграція також підтримує апгрейд застарілих версій MySQL до найсвіжіших версій.

Для ролі середовища розробки програмної частини було обрано продукт компанії JetBrains – IntelliJ IDEA (рис. 2.7), котрий, на даний момент, вважається найкращим середовищем розробки для Java.



Рисунок 2.7 – Вікно завантаження IntelliJ IDEA

					КС КРБ 123.213.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		27

IntelliJ IDEA – одне з тих середовищ розробки, яка користується повагою у справжніх професіоналів. Платформа орієнтована, в першу чергу, на Java-кодинг, але її універсальність дозволяє працювати з різними мовами. Звичайно, тут немає такого масованого набору плагінів, як у будь-якого open-source конкурента, але зате в їх якості можна бути впевненим. Якщо порівнювати це середовище з операційними системами, то IntelliJ буде те ж саме, що Mac OS – закрита, злегка консервативна, але й надійна.

Деякі із особливостей та переваг IntelliJ IDEA:

- глибокий аналіз коду – IntelliJ IDEA аналізує код в пошуках зв'язків між символами у всіх файлах і на всіх мовах, що використовуються в проєкті. На основі цього аналізу IntelliJ IDEA надає допомогу при написанні коду, зручну навігацію, перевірку помилок в коді і, звичайно, рефакторинги;

- ергономічне середовище – кожна складова IntelliJ IDEA розроблена для зручності і ефективності користування. Основний даної IDE – ніщо не повинно відволікати розробника від творчого процесу;

- великий набір вбудованих інструментів «з коробки» для розробників для максимального спрощення робочого процесу: декомпілятор, переглядач байт-коду, FTP і багато іншого;

- розробка на різних мовах, підтримка різних технологій та фреймворків – крім Java, IntelliJ IDEA «з коробки» надає першокласну підтримку провідних фреймворків (JVM і інших) і популярних мов. Крім того, має вбудовані інструменти для роботи з різними технологіями та бібліотеками.

2.4 Вибір програмно-апаратних рішень для реалізації серверної частини

Оскільки, розроблювана комп'ютеризована система не потребуватиме серйозних обчислювальних потужностей для свого функціонування, то сенсу в придбанні нового окремого фізичного сервера для неї немає.

Тому, для ролі фізичного сервера було обрано наявний HP ProLiant

					КС КРБ 123.213.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		28

DL160 G6 (рис. 2.8 - 2.9) – це швидкий, високопродуктивний (табл. 2.2) і недорогий стійковий сервер прекрасно оснащений для виконання складних завдань. DL160 G6 – це хороше рішення для роботи з додатками, що базуються на виробничо-комерційних обчисленнях з інтенсивним використанням пам'яті.



Рисунок 2.8 – Сервер HP ProLiant DL160 G6



Рисунок 2.9 – Сервер HP ProLiant DL160 G6 у серверній коледжу

Змн.	Арк.	№ докум.	Підпис	Дата

КС КРБ 123.213.00.00 ПЗ

Арк.

29

Таблиця 2.2 – Короткі технічні характеристики обраного сервера

Технічні характеристики HP ProLiant DL160 G6	
Процесор(и)	2x Intel Xeon L5640, 2.266 GHz, Hexa Core
Материнська плата	HP ProLiant DL160 G6 (LGA 1366)
RAM	32 GB DDR3
HDD	1 TB
БЖ	600 Вт
ОС	Windows Server 2012 R2 Standard Server

Даний сервер є вже налаштованим і використовується у навчальному процесі як термінальний сервер. Ця роль не дає великого навантаження на нього і не перешкоджає створення йому ще одної ролі – забезпечення серверної частини для комп'ютеризованої системи «Електронний журнал».

Проте, щоб відділити ці дві взаємозалежні ролі було прийнято рішення зробити «сервер на сервері» за допомогою системи віртуалізації VMware Workstation Pro (рис. 2.10), на яку буде встановлено окрему операційну систему і налаштовано базу даних для «Електронного журналу». Крім того, віртуалізація серверної частини для комп'ютеризованої системи надасть властивість мобільності – у випадку зміни фізичного сервера не буде необхідності перевстановлювати ОС із налаштованим середовищем для БД.



Рисунок 2.10 – Головне вікно VMware Workstation Pro

VMware Workstation – система віртуалізації ОС від компанії VMware для різних платформ, що дає можливість запустити на ПК кілька ОС одночасно. Кожна віртуальна машина може мати свою, іншу операційну систему, роду Microsoft Windows чи Linux.

VMware Workstation є особливо корисною коли необхідно тестувати клієнт-серверні середовища.

Для ролі серверної операційної системи було обрано Windows Server 2012 R2 Standard Server – серверна версія операційної системи від Microsoft.

Windows Server є висококласною серверною операційною системою, на основі котрої побудована велика кількість найбільших центрів обробки даних, що дає означає наявність широких можливостей системи для підприємств будь-якого розміру по всьому світу. Windows Server 2012 включає великий набір вдосконалених та абсолютно нових функцій, що дозволяють вдосконалювати IT-середовища віртуалізації даних та хмарних обчислень. У Windows Server 2012 введено інновації в функцію віртуалізації, загальних мережових технологій, систем зберігання даних і зручності роботи.

					КС КРБ 123.213.00.00 ПЗ	Арк.
						31
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОЕКТУ

3.1 Створення бази даних

При створенні БД необхідно враховувати три основні складові моделі даних реляційної БД:

- структуру даних;
- операції, що виконуються над даними;
- обмеження, від яких залежить цілісність даних.

Головна структура даних в реляційній моделі – таблиці, що в реляційній теорії називаються «відношеннями». Власне від терміну «відношення» (англ. «relation») і сформувалася назва моделі – реляційна.

Логічна модель бази даних відображає інформаційно-логічну модель досліджуваної предметної області.

Інформаційні об'єкти моделі даних відображається відповідною реляційною таблицею. Структура цієї таблиці визначається реквізитами відповідного інформаційного об'єкта, де кожне поле відповідає одному з реквізитів об'єкту.

Головні реквізити об'єкту формують унікальний ключ реляційної таблиці. Для усіх полів таблиці (поля) задаються типи, розміри даних і інші властивості. Записи таблиці відповідають складовим об'єктам і формуються при ініціалізації таблиці.

Створення бази даних на розробленій раніше логічній моделі відбувалося у середовищі MySQL Workbench (рис. 3.1).

Таблиці та зв'язки створювалися за логічною моделлю БД. Загалом схема БД, на якій присутні всі зв'язки, поля, розміри полів та типи даних має

					КС КРБ 123.213.00.00 ПЗ			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		<i>Дзюбак Ю.Ю.</i>			<i>Реалізація проекту</i>	<i>Лім.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Перевір.</i>		<i>Луцик Н.С.</i>					32	27
<i>Реценз.</i>						<i>ТНТУ, каф. КС, гр. СІс-43</i>		
<i>Н. Контр.</i>		<i>Тиш Є.В.</i>						
<i>Затверд.</i>		<i>Осухівська Г.М.</i>						

такий вид (рис. 3.2):



Рисунок 3.1 – Створена БД із списком таблиць

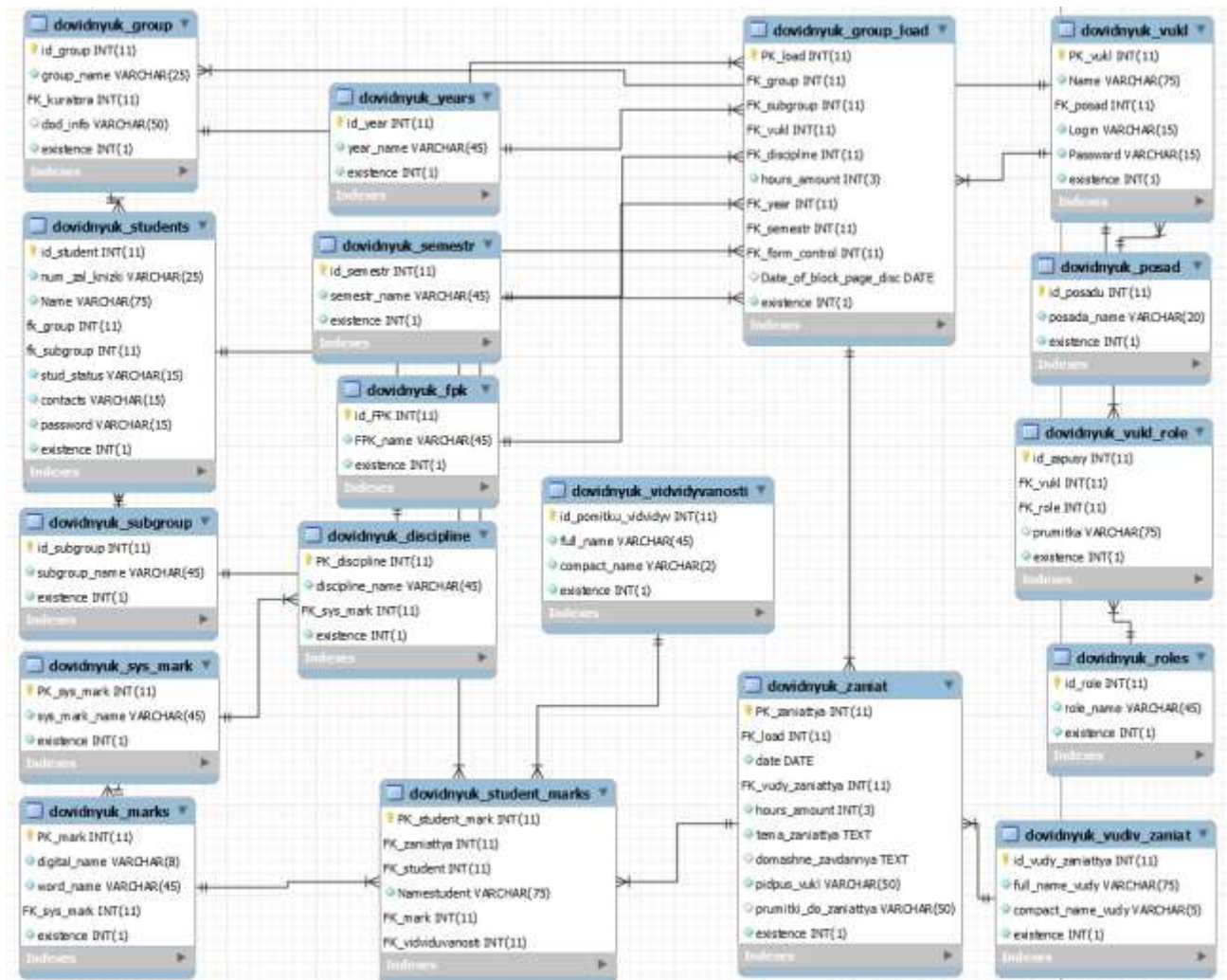


Рисунок 3.2 – EER-діаграма створеної БД у MySQL Workbench

3.2 Опис методів та ресурсів для розробки інформаційної системи

Теорія архітектури «клієнт-сервер» виникла з розвитком уявлень про процеси обробки даних і розподілені обчислювальні процеси та стала уніфікованим поняттям про взаємозв'язок складових обчислювальних системах в мережах, серед яких можна виділити:

- клієнтська сторона, що є джерелом запитів;
- серверна сторона, що є джерелом відповідей.

Взаємодія між учасниками даної архітектури у мережі відбувається за допомогою визначеного раніше протоколу.

В концепції даної архітектури мається на увазі, що крім хостування центральної БД сервер бази даних повинен займатись виконанням основного об'єму обробки даних. Запит клієнта запускає процес пошуку даних на сервері. Знайдені дані передаються по мережі від сервера до клієнта. За специфіку «клієнт-серверної» відповідає мова запитів SQL (Structured Query Language).

Запит до бази даних виконує клієнт, але виконується вона на сервері. Результат повертається клієнту по мережі (рис. 3.3).

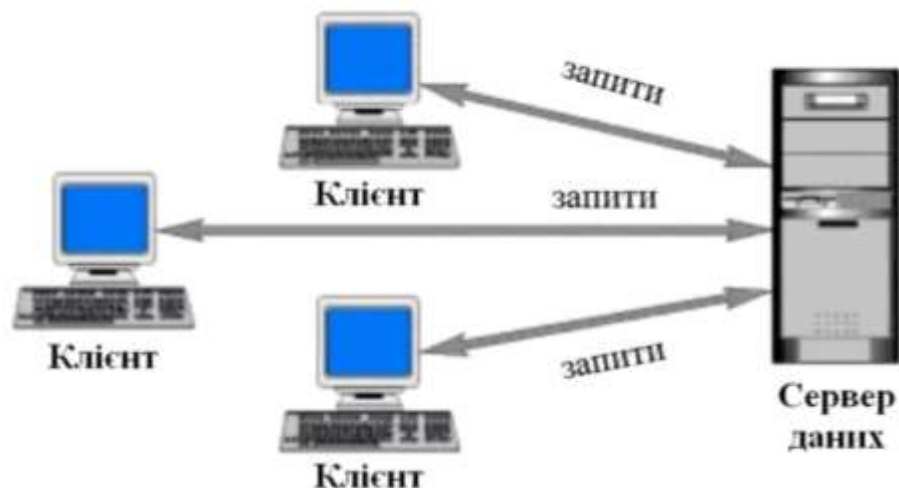


Рисунок 3.3 – «Клієнт-серверна» технологія

Цей процес складається з шести етапів:

- 1) Клієнт запитує дані;
- 2) Запит транслюється в SQL;
- 3) SQL-запит передається по мережі на сервер;
- 4) Сервер бази даних здійснює процес пошуку;
- 5) Необхідні записи повертаються клієнтові;
- 6) Дані пред'являються користувачу.

Використання «клієнт-серверної» технології має реальні переваги, такі як:

- відносно невелика вартість системи;
- достатні обчислювальні можливості;
- легке налаштування під конкретні завдання;
- менші вимоги до обсягу оперативної пам'яті у клієнтів;
- менші вимоги до обсягу дискового простору у клієнтів;
- сервери можуть зберігати великий обсяг даних;
- спрощується процес резервного копіювання даних;
- простіший процес керування всією системою і контролем її безпеки.

Для розробки самого застосунку основним інструментом була мова програмування Java.

Java – це мова програмування і обчислювальна платформа, розроблена компанією Sun Microsystems у 1995 році. Вона розвинулась із скромних початків, щоб забезпечити живлення значної частини сучасного цифрового світу, забезпечуючи надійну платформу, на якій побудовано багато послуг і програм. Нові, інноваційні продукти та цифрові послуги, розроблені для майбутнього, також продовжують покладатися на Java.

У той час як більшість сучасних програм Java об'єднують середовище виконання Java та програму разом, все ще існує багато програм і навіть деякі веб-сайти, які не працюватимуть, якщо не встановлено Java для настільного комп'ютера.

Для розробки необхідно встановити Java Runtime Environment (JRE).

					КС КРБ 123.213.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35

JRE містить у собі віртуальну машину Java (JVM), основні класи платформи Java та базові бібліотеки Java. JRE – це частина програмного забезпечення Java, що виконується під час виконання програм написаних на Java. Також необхідними є інструменти розробки – Java Development Kit (JDK).

Основою для створення програми на мові Java стала платформа JavaFX.

JavaFX – це платформа на основі мови Java, що розроблена для створення застосунків із розвиненим графічним інтерфейсом. Вона може бути використана для створення настільних застосунків (що запускаються з ОС), для розробки веб-додатків типу RIA, що запускаються в браузерах, а також, для мобільних застосунків. JavaFX є заміною застарілої графічної бібліотеки Swing. Кнокурентами даної платформа є Adobe Flash, Microsoft Silverlight і ін. схожі системами.

Безпосередньо для розробки застосунку були використані описані нижче бібліотеки (драйвери).

MySQL Connector Java (driver або external library) – бібліотека, що забезпечує підключення застосунку до бази даних MySQL та надає певні інструменти для налаштування створюваного підключення.

Connection – забезпечує з'єднання (сеанс) із певною базою даних. Виконуються оператори SQL і результати повертаються в контексті з'єднання. База даних об'єкта Connection може надавати інформацію, що описує її таблиці, підтримувану граматику SQL, її збережені процедури, можливості цього з'єднання і так далі.

DriverManager – основний клас для управління набором драйверів JDBC. У рамках своєї ініціалізації клас DriverManager завантажує класи драйверів, на які посилається система jdbc.drivers. Це дозволяє користувачеві налаштовувати драйвери JDBC, що використовуються їхніми додатками.

ResultSet – таблиця даних, що представляє набір результатів бази даних, яка зазвичай генерується шляхом виконання оператора, який запитує базу даних. Об'єкт ResultSet має вказівник поточного рядка даних. Спочатку

					КС КРБ 123.213.00.00 ПЗ	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

курсор розташовується перед першим рядком. Наступний метод переміщує курсор до наступного рядка, і оскільки він повертає false, коли в об'єкті ResultSet більше немає рядків, його можна використовувати в циклі while для перебору набору результатів.

Statement – об'єкт, який використовується для виконання статичного оператора SQL і повертає отримані результати. За замовчуванням одночасно може бути відкритий лише один об'єкт ResultSet на об'єкті Statement. Отже, якщо зчитування одного об'єкта ResultSet чергується з читанням іншого, кожен повинен бути згенерований різними об'єктами Statement. Усі способи виконання в інтерфейсі Statement неявно закривають поточний об'єкт ResultSet Statement, якщо він існує.

SQLException – виняток, який надає інформацію про помилку доступу до бази даних або інші помилки.

ObservableList – список, що дає можливість стежити за змінами, якщо вони відбуваються.

FXCollections – службовий class, який складається з статичних методів, які є копією java.util. У методів обгортки (як synchronizedObservableList або emptyObservableList) є точно та ж сама функціональність як у методів в наборі з виключенням, що повертають ObservableList і є тому придатними для методів, які вимагають ObservableList на ввіді. Службові методи тут, головним чином, з причин продуктивності.

ArrayList – найпопулярніший вид списків. ArrayList – автоматично розширюваний масив. Ви можете працювати з масивом, але при цьому не використовуються квадратні дужки.

JFoenix – бібліотека Java з відкритим кодом, яка реалізує дизайн матеріалів Google за допомогою компонентів Java.

iTextPDF – є однією з найбільш повно задокументованих і найбільш гнучких бібліотек для роботи з файлами PDF у світі (для мов Java і .NET), що дозволяє інтегрувати функції PDF не тільки в робочі процедури, але й у додатки, процеси та продукти.

					КС КРБ 123.213.00.00 ПЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

Фактично, створювана програма є CRUD-застосунком.

CRUD – аббревіатура, що означає чотири базові функції, котрі використовуються при роботі з базами даних: створення (англ. «create»), читання («read»), модифікація («update»), видалення («delete»). Приклад базового CRUD-застосунку на основі JavaFX та MySQL був знайдений на ютуб-каналі «kenLovesToCode» [9]. Деякі матеріали із даного прикладу, в модифікованому варіанті, стали основою довідкових форм застосунку.

3.3 Розробка інтерфейсів застосунку для користувачів системи

Першочергово було створено структуру («скелет») проекту (рис. 3.4).

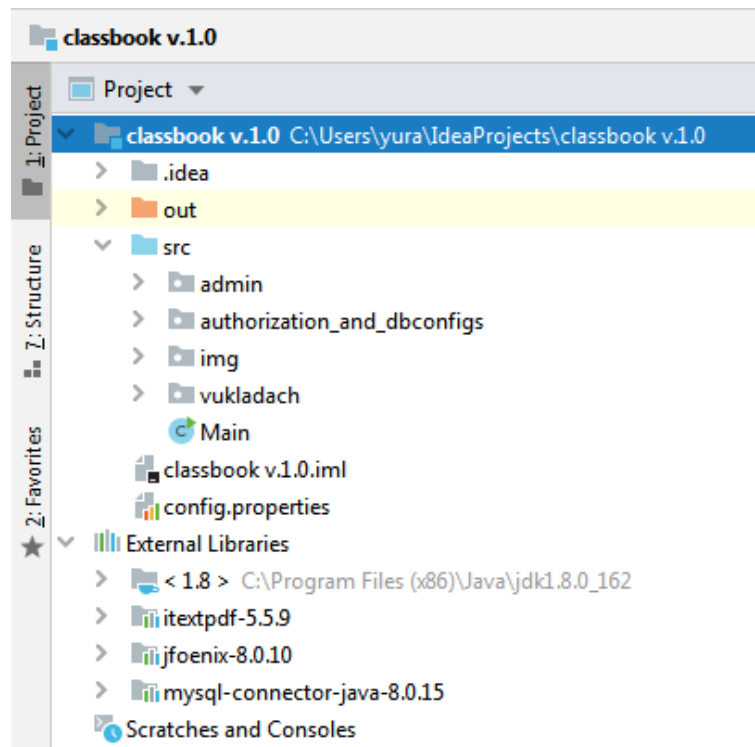


Рисунок 3.4 – Стартова структура програмного проекту

Наступним кроком було створення класу DBConnection (рис. 3.5), котрий і відповідає за зв'язок із базою даних.

```

public class DBConnection {
    private final String dburl;
    private Connection connect;
    private String host;
    private String port;
    private String db;
    public DBConnection() {
        FileInputStream fis;
        Properties property = new Properties();
        String path = System.getProperty("user.dir");
        try {fis = new FileInputStream(path+"/config.properties");
            property.load(fis);
            host = property.getProperty("db.host");
            port = property.getProperty("db.port");
            db = property.getProperty("db.name");
        } catch (IOException e) {
            System.err.println("Error: Config file is not allowed!");
        }
        dburl =
        "jdbc:mysql://" + host + ":" + port + "/" + db + "?allowPublicKeyRetrieval=true&useUnicode=true&useJDBCCompliantTimezoneShift=true&useLegacyDateTimeCode=false&serverTimezone=UTC&useSSL=false";
    }

    public Connection getConnection() {
        try {
            connect = DriverManager.getConnection(dburl, username, password);
        } catch (Exception e) { e.printStackTrace(); }
        return connect;
    }

    public void close(Connection connect, PreparedStatement pstmt,
    ResultSet rs) {
        try {if(connect != null)connect.close();
            if(pstmt != null)pstmt.close();
            if(rs != null)rs.close();
        } catch (Exception e) { e.printStackTrace(); }}

    public void close(Connection connect, PreparedStatement pstmt)
    { try { close(connect, pstmt, null);
        } catch (Exception e) { e.printStackTrace(); } }

    public void close(PreparedStatement pstmt) {
        try { close(null, pstmt, null);
        } catch (Exception e) { e.printStackTrace(); } }
}

```

Рисунок 3.5 – Лістинг коду класу DBConnection

Важливим у даній реалізації є те, що дані доступу до БД зберігаються не напряму у кодї, а у property-файлі (файлі властивостей) – котрий призначений, щоб зберігати будь-які статичні дані необхідні проекту, наприклад, логін і

					КС КРБ 123.213.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		39

пароль до БД.

Після створення класу DBConnection було створення форм та дизайну для адмінпанелі застосунку (рис. 3.6).



Рисунок 3.6 – Адмінпанель застосунку

Загалом для створення функціоналу адмінпанелі застосунку використовувалося два основних типи форм: повнорозмірна (рис. 3.7) та підформа (рис. 3.8), зразки яких приведено панелями «Керування навантаженням» та «Керування навчальними роками».



Рисунок 3.7 – Повнорозмірна панель «Керування навантаженням»

Змн.	Арк.	№ докум.	Підпис	Дата

КС КРБ 123.213.00.00 ПЗ

Арк.

40

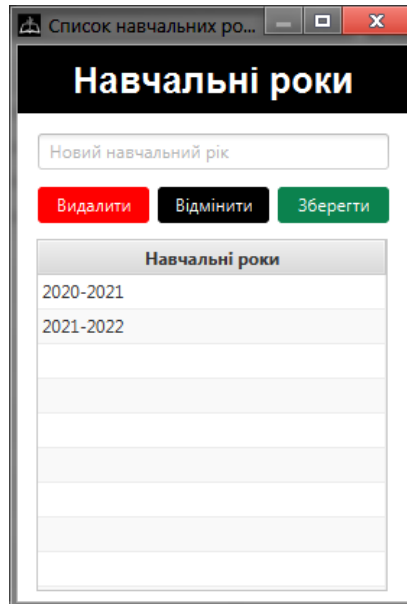


Рисунок 3.8 – Підформа «Керування навчальними роками»

Усі інші панелі створені за аналогічними моделями форм. Кожна панель прив’язана до своєї таблиці або декількох пов’язаних таблиць у БД.

Для кожної форми у програмі створюється три індивідуальні класи: FXML-клас (де описаний зовнішній вид форми та прописані id компонентів), клас-контролер форми (котрий відповідає за методи-події, які будуть відбуватися при роботі з формою) та клас-відправник/приймач (що відповідає за змінні, в які відбувається зчитування даних з БД та через які відбувається передача даних у БД), а також дописуються необхідні методи у спільний для всіх форм клас DataAccessObject, що відповідає за безпосередній контакт із БД (зчитування, збереження).

Для розробки загального стилю застосунку та створення фонових зображень використовувалися фоторедактори та власна фантазія. Фронтенд форм створювався в редакторі для JavaFX – Scene Builder від компанії Gluon, що дозволяє легко створювати елементи керування інтерфейсу JavaFX, діаграми, фігури та контейнери, щоб швидко і якісно прототипувати інтерфейси користувача.

Після створення адмінпанелі було створено дизайн та функціонал головного вікна авторизації у систему (рис. 3.9).



Рисунок 3.9 – Головне вікно авторизації у систему

Також були розроблені та додані функції перевірки ідентифікаторів, що вводяться у поля, і зроблені візуальні підказки, наприклад, коли одне з полів (або й обидва) заповнено неправильно, символи заповненого поля підсвітяться червоним, а те, що заповнено вірно, – зеленим (рис. 3.10).

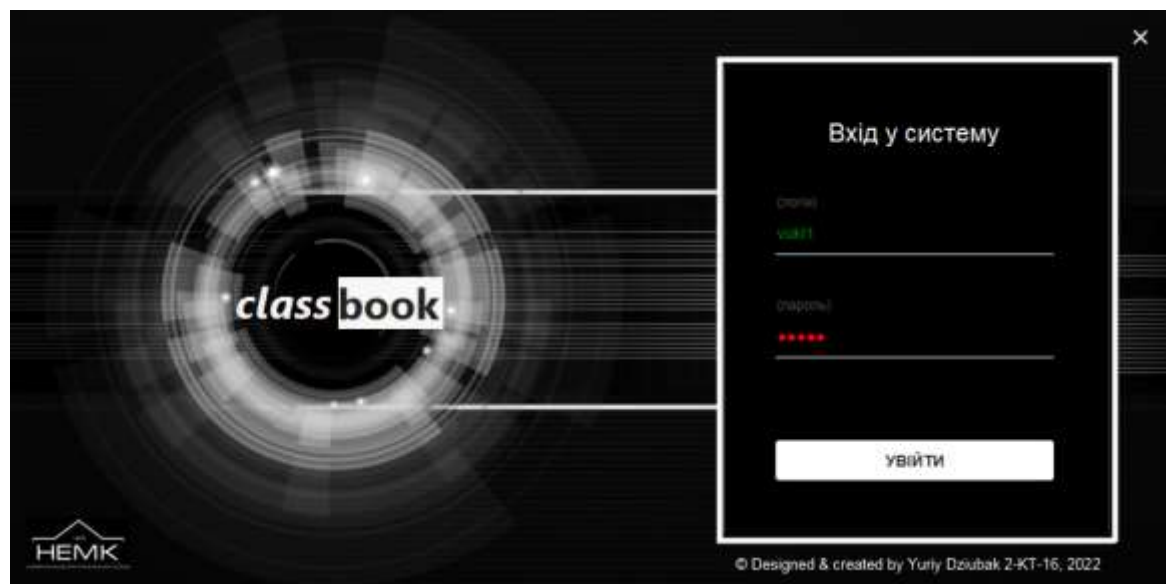


Рисунок 3.10 – Головне вікно авторизації у систему з неправильно введеним користувачем паролем

При розробці було враховано можливість перевірки програмою статусу з'єднання із сервером (рис. 3.11).

					КС КРБ 123.213.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		42



Рисунок 3.11 – Головне вікно авторизації у систему при відсутності з'єднання з сервером

Після реалізації цього етапу було створено стартову панель персонального акаунта користувача системи. На даній панелі відображається ПШБ авторизованого користувача і доступні йому ролі (рис. 3.12) для поточного сеансу. Було реалізовано такі ролі: головна роль – «Викладач», суть якої аналогічна ролі викладача при заповненні традиційного паперового журналу і додаткові ролі – «Представник адміністрації навчального закладу» та «Куратор».

Додаткові ролі відрізняються тим, що «Представник адміністрації навчального закладу» може обирати будь-яку групу, рік, семестр та дисципліну і переглядати дані по них, а «Куратор» може переглядати всі дані тільки своєї групи, але обидві ці ролі не мають можливості внесення нових даних (крім приміток) та зміни вже існуючих даних у відомостях, що доступні їм до перегляду.

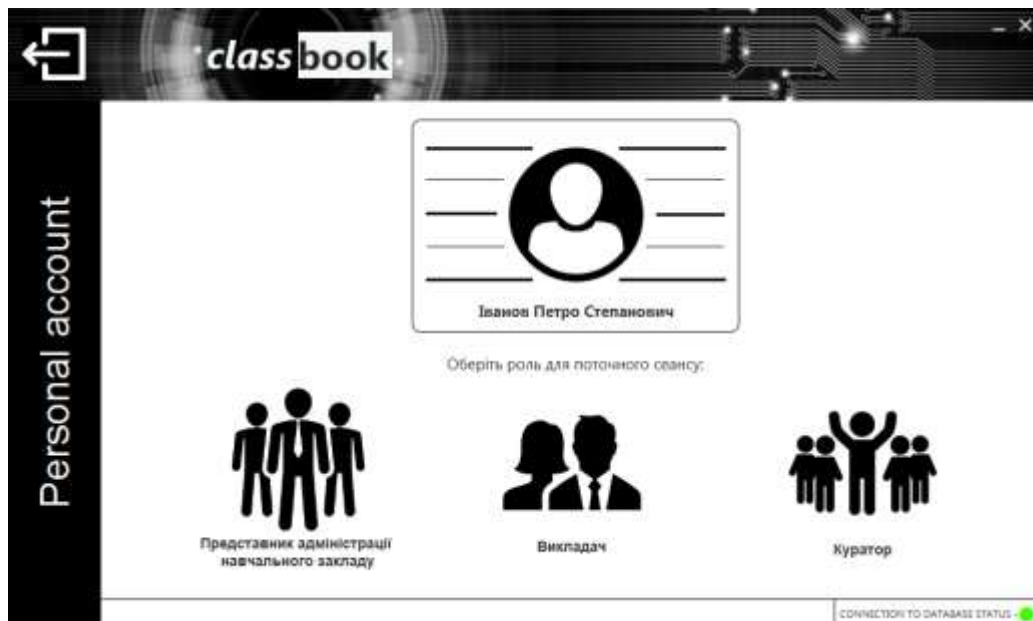


Рисунок 3.12 – Панель персонального акаунту користувача

Наступною була реалізована панель вибору викладачем необхідних доступних даних із довідника навантаження для створення заняття (рис. 3.13).



Рисунок 3.13 – Панель вибору викладачем даних для створення заняття

Далі було створено форму «Керування заняттям», що є фактично «правою» стороною паперового класного журналу (рис. 3.14-3.16).

The screenshot shows the 'classbook' interface. The main area is a table with columns: №, Дата проведення, Вид заняття, Кількість годин, Тема заняття, Домашнє завдання, Підпис, and Прямийк. The table is currently empty, showing 'No content in table'. On the left, there are form fields for:

- Дата проведення заняття: [empty]
- Вид заняття: [dropdown menu]
- Кількість годин: [input field]
- Тема заняття: [text area]
- Домашнє завдання: [text area]
- Підпис: [input field]
- Прямийк: [input field]

 Buttons include 'Очистити поля' (Clear fields) and 'Зберегти зміни' (Save changes). At the bottom, there are status indicators: 'Кількість год. до вичиток: 104', 'Перейти до оцінювання/відвідуваності заняття' (blue button), 'ФПК: екзамен', and 'CONNECTION TO DATABASE STATUS: [green dot]'.

Рисунок 3.14 – Форма «Керування заняттям»

This screenshot shows the same form as Figure 3.14, but with data entered. The form fields are filled as follows:

- Дата проведення заняття: 12.06.2020
- Вид заняття: Л
- Кількість годин: 2
- Тема заняття: Тема 1.
- Домашнє завдання: Домашнє завдання 1.
- Підпис: Іванко

 The 'Зберегти зміни' button is now green, indicating a successful save. The rest of the interface, including the empty table and status indicators, remains the same.

Рисунок 3.15 – Заповнення форми «Керування заняттям»

Керування заняттями
дисципліна: Програмування

Дата проведення заняття:

Вид заняття:

Кількість годин:

Тема заняття:

Домашнє завдання:

Підпис:

Примітки:

Очистити поля

Зберегти зміни

Кількість год. до вичитки: 104

Перейти до оцінювання/відвідуваності заняття

ФПК: вкранен

CONNECTION TO DATABASE STATUS: ●

№	Дата проведення	Вид заняття	Кількість годин	Тема заняття	Домашнє завдання	Підпис	Примітки
1	2020-06-12	Л	2	Тема 1.	Домашнє завдання 1.	Іванов	

Рисунок 3.16 – Заповнена інформація про одне заняття у формі «Керування заняттями»

Наступним етапом було створення форми оцінювання заняття (рис. 3.17-3.18). Щоб перейти до форми оцінювання заняття, необхідно виділити потрібне заняття у таблиці та натиснути синю кнопку «Перейти до оцінювання/відвідуваності заняття».

Керування оцінюванням заняттями
дисципліна: Програмування

Оцінка:

Відмітка відвідування:

Зберегти оцінку/відвідуваність вибраного студента

Відмінити вибір (очистити поля)

Згенерувати "ліву сторінку журналу" за даними дисципліни

CONNECTION TO DATABASE STATUS: ●

№	ПІБ студента	Оцінка	Відвідуваність
1	Бойчук У.І.		
2	Корольов І.В.		
3	Зубайко М.У.		

Рисунок 3.17 – Форма оцінювання заняття

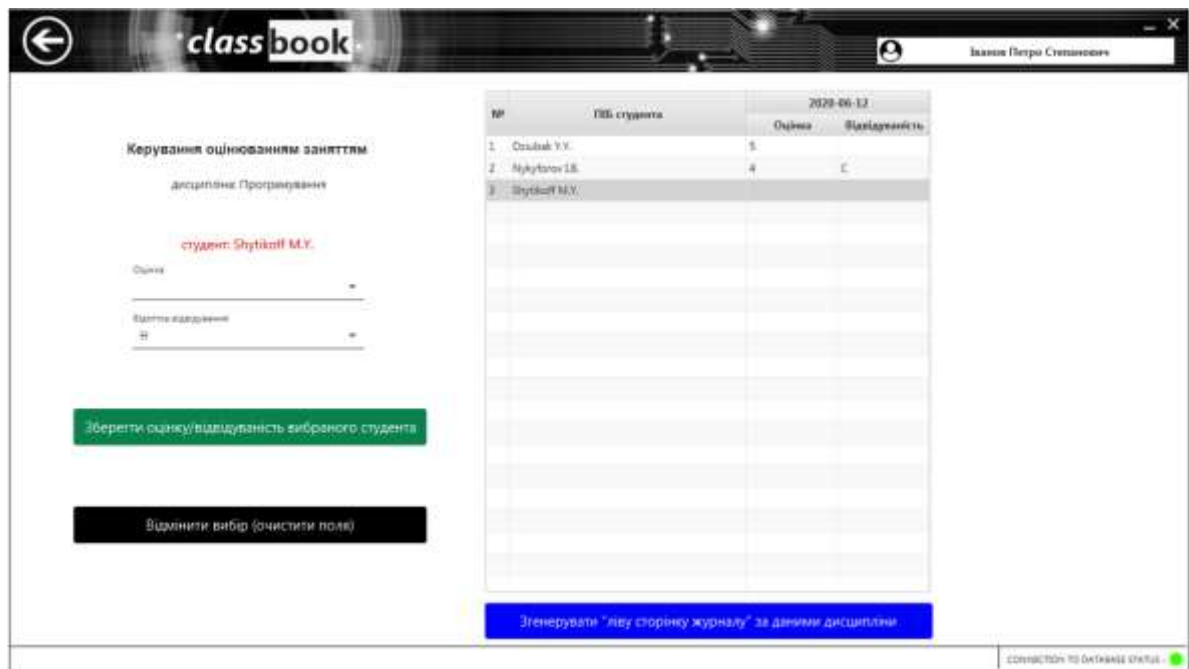


Рисунок 3.18 – Приклад оцінювання/проставлення оцінок та відміток відвідуваності вибраного студента

Для зручнішого візуального відображення відомості з усієї дисципліни було створено ще одну форму (рис. 3.19), що, по факту, відображає «ліву» сторону журналу.

ПІБ студента	2020-06-12	2020-06-12
Dziubak Y.Y.	5	
Nykyforov I.B.	4	C
Shytikoff M.Y.		H

Рисунок 3.19 – Згенерована форма обліку оцінювання та відвідуваності студента

Також, для більш зручного використання створеного застосунку формами можна користуватися одночасно кількома вікнами (рис. 3.20):



Рисунок 3.20 – Приклад паралельного використання двох форм

Загалом програмна структура проекту набула такого вигляду (рис. 3.21):

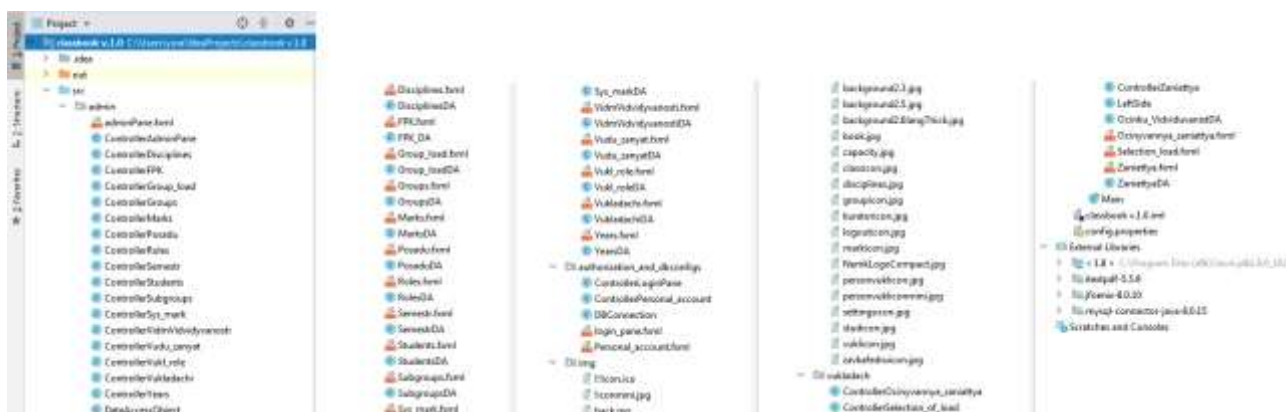


Рисунок 3.21 – Програмна структура проекту

Наступним етапом була реалізація можливості друку заповнених даних, що дасть можливість створення «фізичного» класного журналу для зшивання та розміщення в архіві навчального закладу.

3.4 Реалізація функції друку сторінок журналу на основі заповнених даних

Для реалізації функції друку сторінок журналу необхідно було спочатку

розробити алгоритм формування документів придатних для друку із даних заповнених в БД. Крім того, також розробити алгоритм формування правильного шаблону даних документів.

Для реалізації даних алгоритмів було обрано iTextPDF – це бібліотека Java, за допомогою якої можна розробляти програми Java, які створюють, конвертують та обробляють документи PDF.

Оскільки, друкувати потрібно «ліві» та «праві» сторінки журналу (рис. 2.2 - 2.3), то необхідно розробити два шаблони для формованих документів. Самі функції, котрі відповідають за формування шаблону, його заповнення та зберігання було розміщено у класах, котрі відповідають за заповнення даних «лівої» та «правої» сторінок журналу. Так було зроблено тому, що ці класи формують таблиці для відображення заповнених раніше даних і є фактично цифровими версіями «правої» та «лівої» сторінок журналу, що означає те, що алгоритм правильного формування даних сторінок не потрібно буде дублювати для створення друкопридатних документів, бо раціональніше використати бібліотечні інструменти для зчитування даних, вже зформованих, цифрових таблиць (рис. 3.20).

Алгоритм формування правої сторінки наведено в Додатку Б у методі AddToPDF() класу ControllerZaniattya.

Цей алгоритм є простішим ніж той що формує «ліву» сторінку, бо «права» сторінка має фіксовану кількість стовпців, на відміну від «лівої» – кількість стовпців котрої залежить від кількості годин дисципліни, яка є змінною. Тобто, «ліва» сторінка, фактично, заповнюється у горизонтальній площині, а «права» у вертикальній.

Дана особливість серйозно ускладнює алгоритм заповнення шаблону «лівої» сторінки. Крім того, необхідно враховувати, що кількість занять на обох сторінках має бути однаковою – якщо під час формування на одній із сторінок закінчився вільний простір, то інша сторінка також автоматично має переходити на новий лист документу.

Всі особливості було враховано і розроблено абсолютно унікальний

					КС КРБ 123.213.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		49

алгоритм заповнення сформованого (за прикладом шаблону «правої») шаблону «лівої» сторінки, котрий наведено в Додатку Б у методі AddToPDF() класу LeftSide.

На форми «Керування заняттям» та «Керування оцінюванням заняття» було додано кнопки (рис. 3.22), що викликають вище описані методи, результатом натискання яких є генерація PDF-документів «лівої» та «правої» сторінок журналу (рис. 3.23).



Рисунок 3.22 – Кнопки для генерації PDF-документів додані на форми

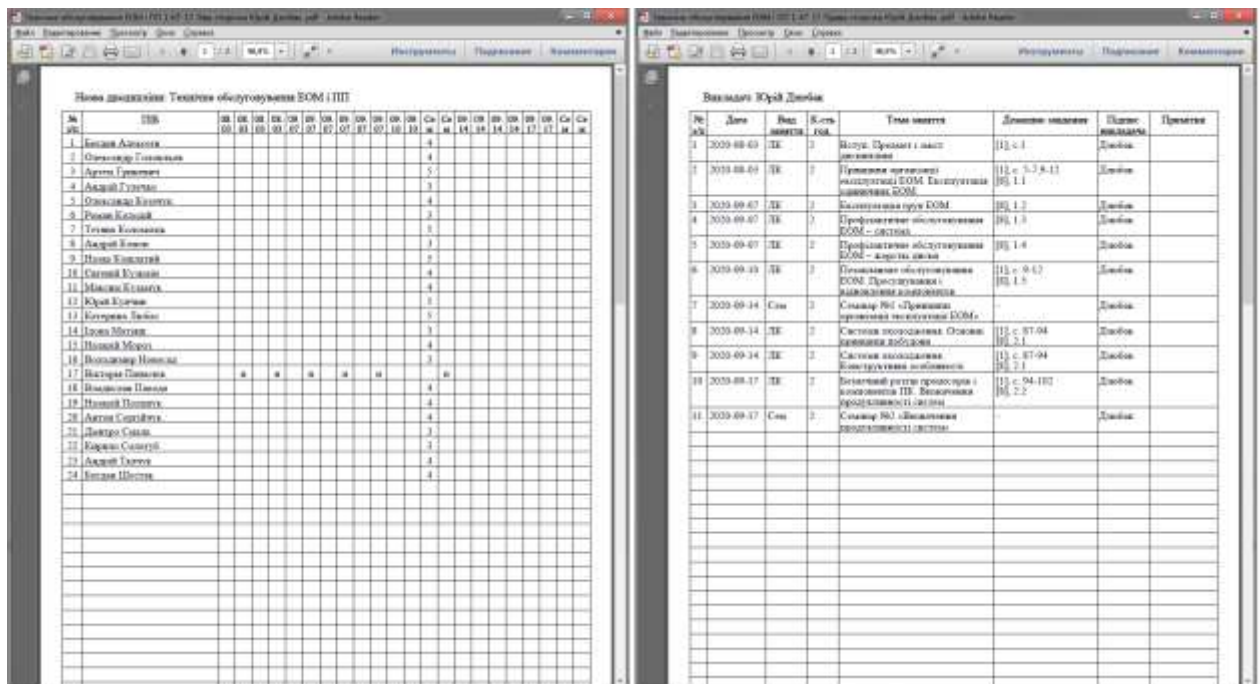


Рисунок 3.23 – Згенеровані PDF-документи «лівої» та «правої» сторінки

3.5 Збірка та компіляція програми у виконуваний файл

Збірка та компіляція програми у виконуваний файл були виконані за допомогою інтегрованих засобів середовища розробки IntelliJ IDEA. Для цього необхідно було перейти у пункт меню Project Structure, де обрати Artifacts – Add – JavaFx Application – From module 'classbook v.1.0' (рис. 3.24).

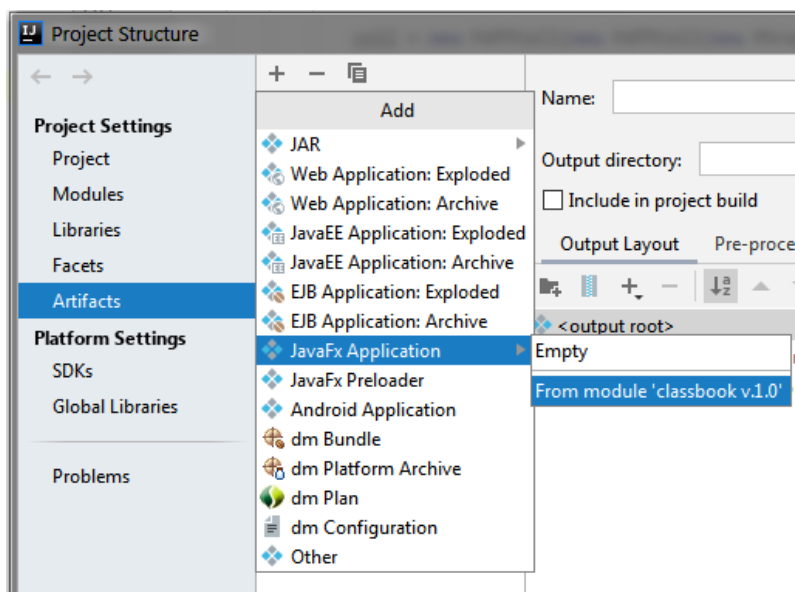


Рисунок 3.24 – Перші кроки для збирання програми

Далі необхідно було вказати, які саме вихідні файли проекту необхідні для збирання (рис. 3.25).

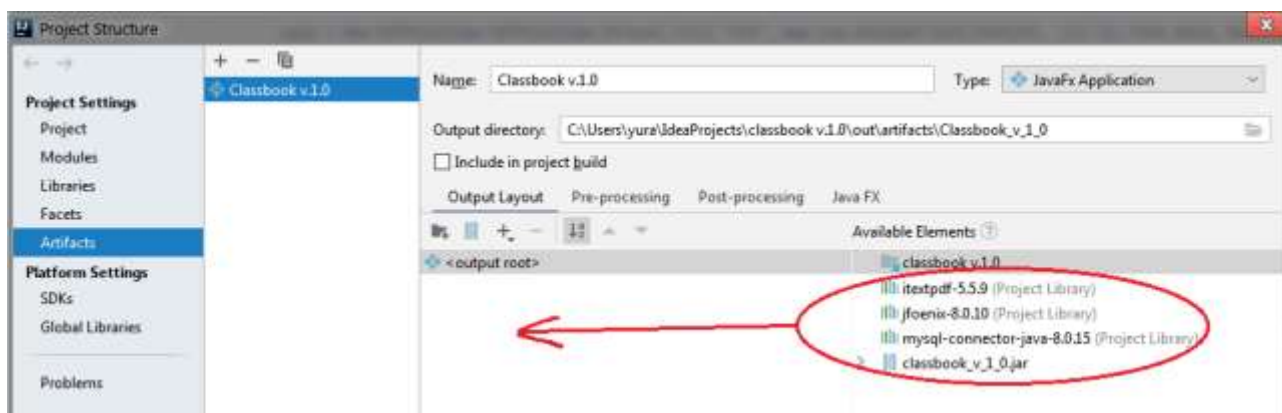


Рисунок 3.25 – Перенос вихідних файлів проекту для формування зібраної програми

Змн.	Арк.	№ докум.	Підпис	Дата

Потім необхідно було задати необхідні параметри і вказати файл іконки застосунку у меню параметрів JavaFx (рис. 3.26).

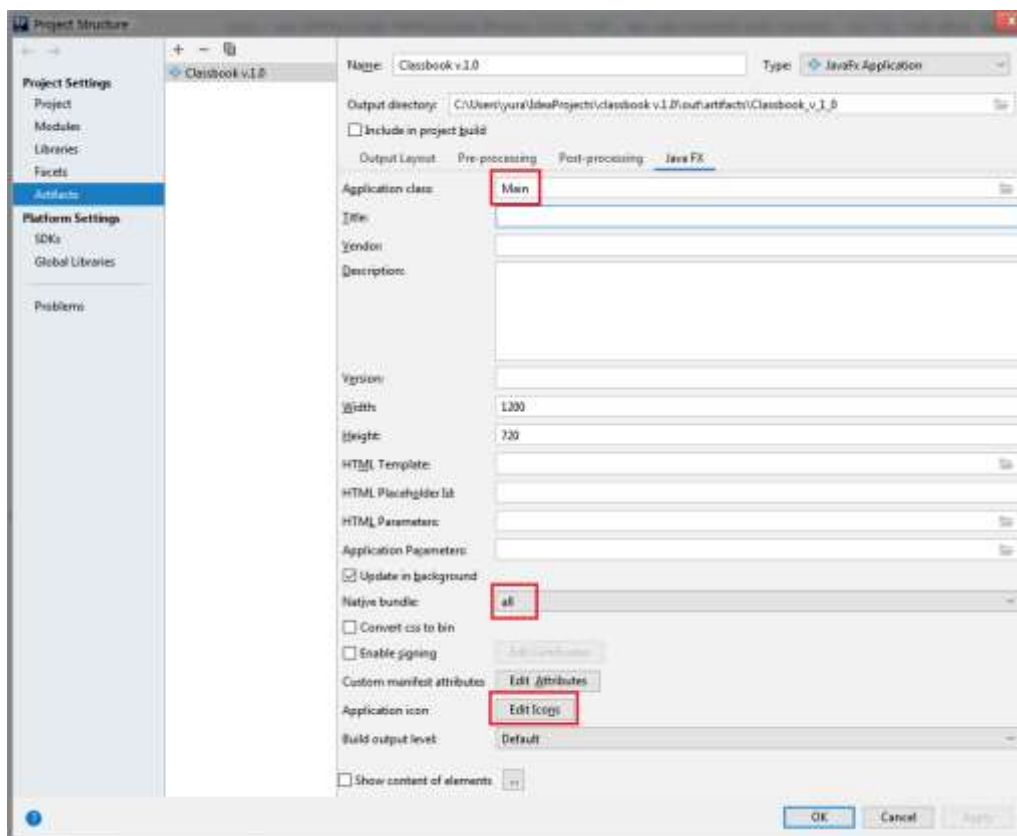


Рисунок 3.26 – Необхідні параметри у меню параметрів JavaFx

Після чого перейти у Build – Build Artifacts та виконати Build (рис. 3.27).

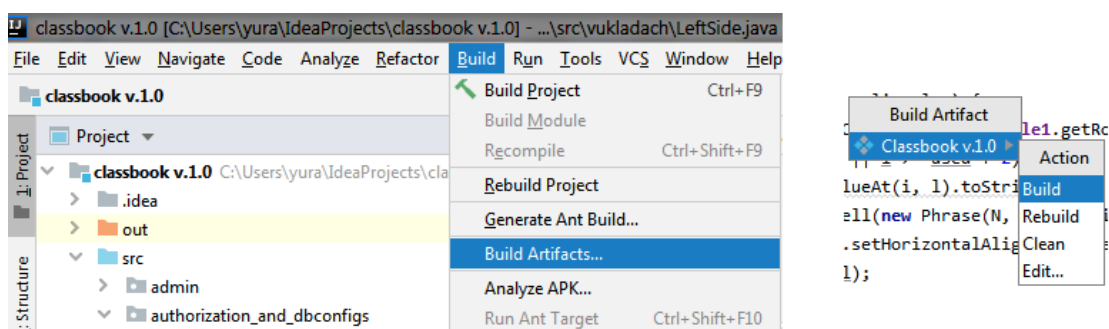


Рисунок 3.27 – Build Artifact

Тоді, якщо виконання Build Artifact пройде без помилок, у теці із проектом в папці out – artifacts з'явиться зібрана програма із виконуваним файлом (рис. 3.28), для функціонування якої із кореня папки проекту у папку

app необхідно ще перенести файл із конфігурацією доступу до БД (рис. 3.29). Після виконання даних маніпуляцій програма буде придатною до використання на будь-якому ПК із ОС Windows 7+.

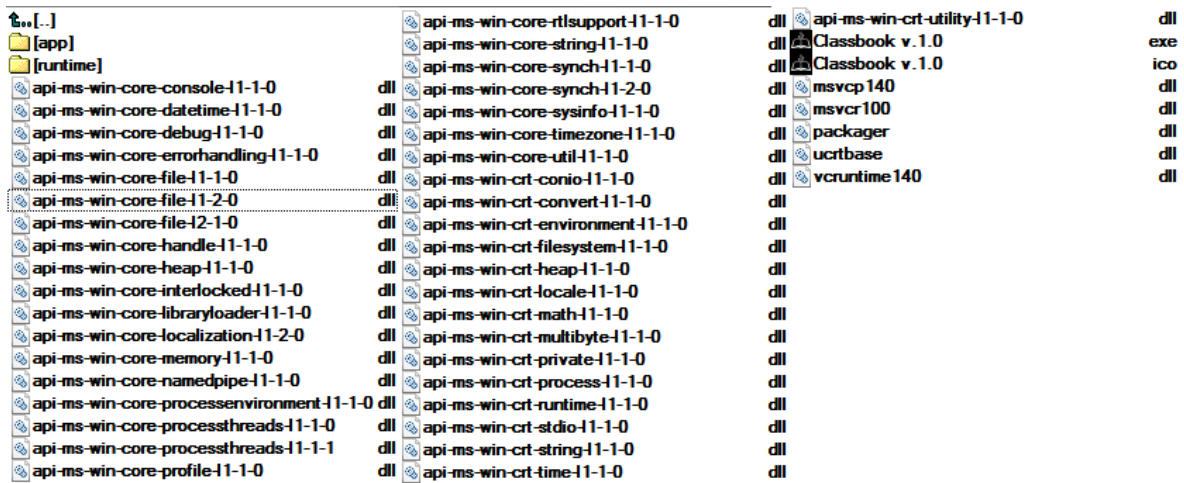


Рисунок 3.28 – Структура зібраної програми із виконуваним файлом

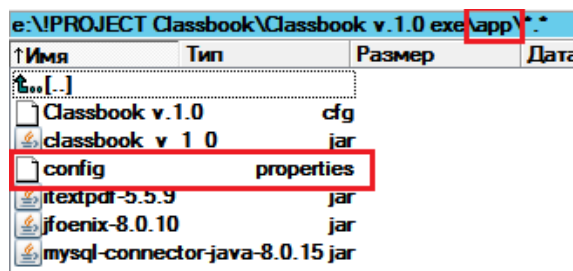


Рисунок 3.29 – Доданий файл із конфігурацією доступу до БД

3.6 Встановлення спеціалізованої ОС на виділений сервер та налаштування на ньому СУБД

Спершу, в уже існуючу ОС на сервері, було встановлено систему віртуалізації VMware Workstation (рис. 3.30), в середовищі якої, потім, було створено і налаштовано нову віртуальну машину (рис. 3.31), на яку буде встановлюватись ОС для сервера комп'ютеризованої системи.



Рисунок 3.30 – Головне вікно встановленої VMware Workstation Pro

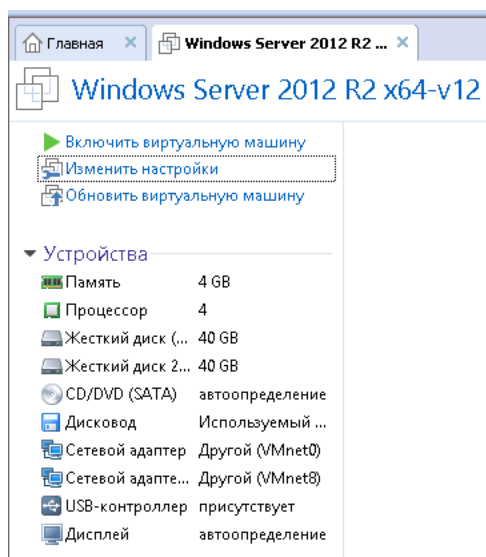


Рисунок 3.31 – Створена і налаштована нова віртуальна машина

Наступним етапом було встановлення на віртуальну машину Windows Server 2012 R2 (рис. 3.32) і базове налаштування його сервісів і мережі.

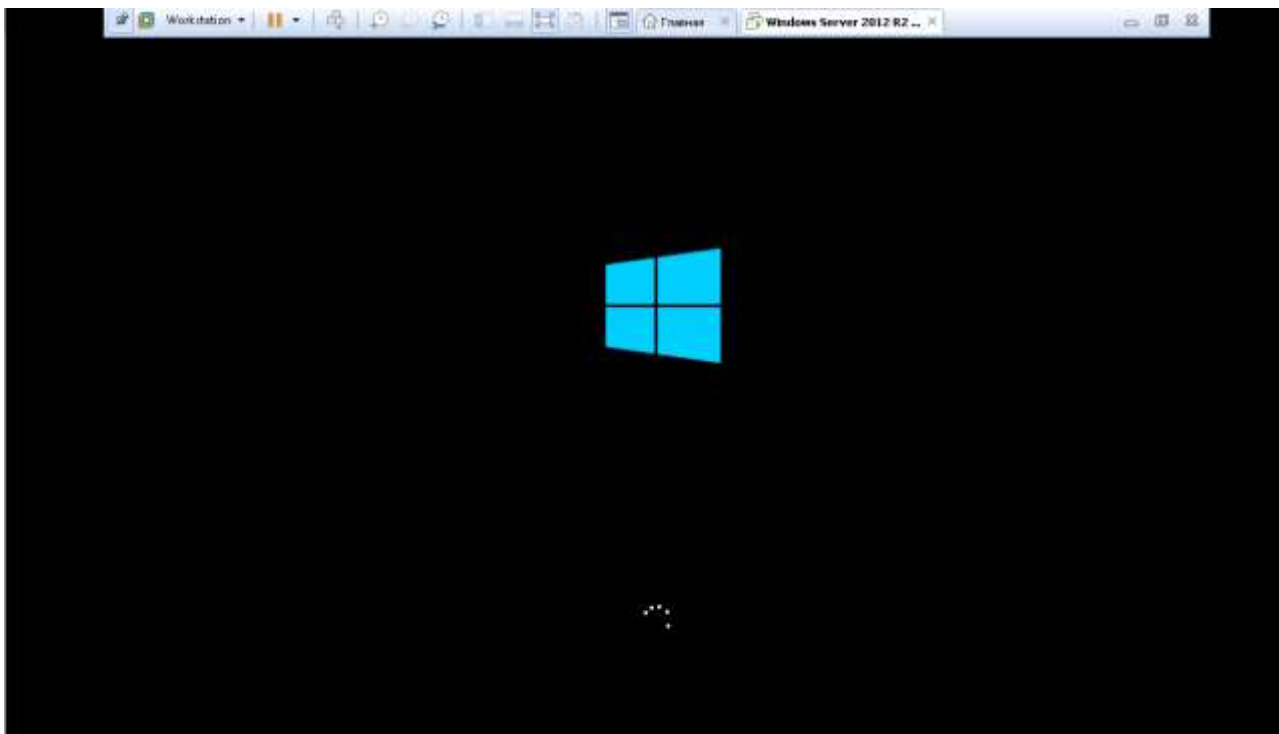


Рисунок 3.32 – Завершальний етап встановлення ОС Windows Server 2012 R2

Далі було встановлено компоненти MySQL: сам MySQL сервер, СУБД WorkBench, Notifier та Connectors. Потім було імпортовано dump із базою даних для комп'ютеризованої системи (рис. 3.33).

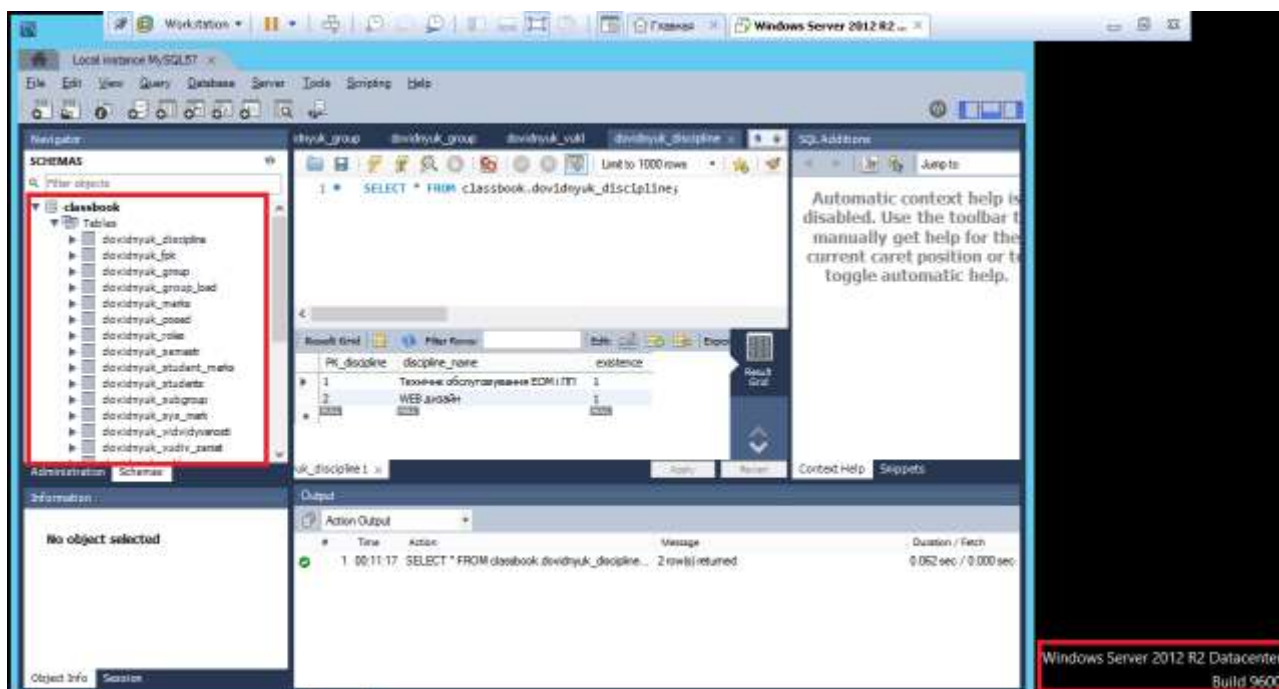


Рисунок 3.33 – Встановлені компоненти MySQL із імпортованою БД у ОС Windows Server 2012 R2

Останнім етапом було створення користувача БД у середовищі MySQL і надання йому права користування базою даних наступними командами (рис. 3.34).

```
CREATE USER 'classbookuser'@'%' IDENTIFIED BY '*****';  
GRANT ALL PRIVILEGES ON classbook.* to 'classbookuser';
```

Рисунок 3.34 – Лістинг SQL-скриптів

Після здійснення усіх необхідних налаштувань було проведено перевірку зв'язку клієнтського застосунку із свіжоналаштованим сервером (рис. 3.35).



Рисунок 3.35 – Успішно встановлений зв'язок клієнтського застосунку із серверною частиною

3.7 Тестування та налагодження комп'ютеризованої системи

У ході тестування, котре відбувалося як і безпосередньо під час розробки, так і після повного створення функціоналу застосунку, було виявлено та виправлено кілька «багів», а також підкориговано та модифіковано певні моменти:

1. Було вирішено проблему, що полягала у тому, що при видаленні

					КС КРБ 123.213.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		56

записів із довідкової таблиці відбувався зсув id записів, що призводило до некоректного вибору Foreign Key для запитів і елементів Combobox. Вирішено проблему за допомогою додавання в усі таблиці поля «активність» з властивістю boolean (варіанти: «1» – використовується, «0» – видалено). Відповідно замість фізичного видалення запис буде приховуватись (рішення схоже на таке, що використовується у корпоративних БД), про що вже згадувалося у підрозділі 2.2.

Крім того, при «видаленні» до основної назви, що видаляється, додається позначка *(DELETED) (рис. 3.36).

	id_year	year_name	existence
▶	1	/	1
	2	2020-2020	1
	3	2019-2020(DELETED)	0
*	NULL	NULL	NULL

Рисунок 3.36 – Демонстрація появи помітки для запису, що визначений на видалення

2. У ранній версії БД існувало три поля «Прізвище», «Ім'я» та «По батькові» для викладачів і студентів. Було вирішено об'єднати ці поля в одне – «ПІБ», оскільки окремо ці поля ніде не використовуються, крім того, стало зручніше використовувати ці дані як довідкові.

3. Додано функціональні кнопки до усіх довідкових підформ (рис. 3.37):

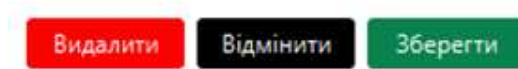


Рисунок 3.37 – Зовнішній вигляд функціональних кнопок підформ

4. Було виявлено та вирішено два «баги» з редагуванням записів:

1) у той момент, коли якийсь запис вибирався для редагування, редагувався і зберігався, залишався зв'язок із полями введення, і, якщо одразу після виконаних вище описаних дій була спроба введення нових даних у очищені поля, то змінювалося те, що було щойно відредаговано. Проблему

вирішено додаванням рядка коду, який забезпечує після завершення операції зберігання «вимикання» умови можливості помилкового редагування, що випадково спрацьовувало;

2) некоректно відображались ід записів у Combobox, коли вибиралось редагування існуючого запису, що провокувало помилку ключа. Проблему вирішено додатковим ініціюванням елементів Combobox у методі, де описано редагування.

5. Було вирішено прибрати модель відображення «лівої» сторінки журналу у вигляді таблиці JTable, а лишити тільки генерацію PDF-документа для перегляду зведеної відомості, через більш правильну модель відображення даної інформації у варіанті згенерованого PDF-документу. При цьому генерація таблиці JTable просто відбуватиметься приховано, бо вона все одно необхідна як шаблон для PDF-документу.

Це далеко не повний список того, що модифікувалося і правилося під час розробки, тому що проект дуже стрімко розвивався і продовжуватиме розвиватися й надалі.

Введення ІС в повноцінну експлуатацію планується після розробки веб-версії із безпековими політиками для користування системою через Інтернет та інтерфейсу для студентів. Також запланований запис відео щодо роз'яснення основних функціональних можливостей системи для проведення навчання серед персоналу коледжу і студентів.

Версія комп'ютеризованої системи, що описана у даній роботі є пілотним тестовим проектом, котрий є базою для подальшого розвитку і модернізації даної системи.

					КС КРБ 123.213.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		58

РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Працездатність людини – оператора

Під працездатністю людини розуміють можливість її виконувати роботу з необхідною якістю та в установлений час. Працездатність людини залежить як від зовнішніх чинників, так і від внутрішнього стану (внутрішні чинники).

До зовнішніх чинників належать: кількість та форма отриманої інформації, зручність робочого місця, характер взаємостосунків в колективі, вплив чинників середовища існування.

До внутрішніх чинників належать: рівень підготовки, тренованість людини та її емоційна стійкість.

У процесі роботи людина переживає різні функціональні стани, які зумовлюють різні рівні її працездатності.

На рисунку 4.1 наведено зміни функціонального стану та якості роботи людини у процесі одного трудового циклу (зміни).

Виділяють 4 фази працездатності: пристосування до праці, стійкої працездатності, субкомпенсації, втоми. Тривалість усіх фаз та усього циклу роботи залежить від рівня підготовки людини до роботи.

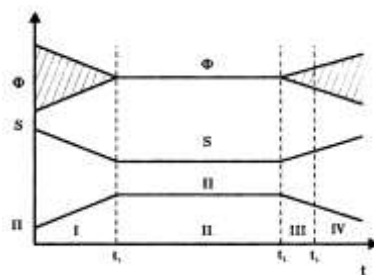


Рисунок 4.1 – Фази працездатності

КС КРБ 123.213.00.00 ПЗ				
Змн.	Арк.	№ докум.	Підпис	Дата
Розроб.		Дзюбак Ю.Ю.		
Перевір.		Луцик Н.С.		
Консульт.		Лазарюк В.В.		
Н. Контр.		Тиш Є.В.		
Затверд.		Осухівська Г.М.		
Безпека життєдіяльності, основи охорони праці				
		Літ.	Арк.	Аркуші
		59	8	
ТНТУ, каф. КС, гр. СІс-43				

Ф – показник функціонального стану;

Б – помилки роботи;

П – продуктивність праці.

Фаза пристосування до праці (0 – 1) – це час, протягом якого людина адаптується до майбутніх умов праці. Основний показник поступово досягає свого встановленого значення. Тривалість періоду пристосування організму до умов праці залежить від багатьох чинників, серед яких основними є інтенсивність роботи (чим інтенсивніша робота, тим цей період коротший) та рівень готовності людини до майбутньої роботи.

Значного скорочення фази пристосування до праці можна досягти за рахунок попередньої підготовки людини до роботи (виконання фізичних вправ, адаптації зору, слуху та ін.) та шляхом посиленого навчального навантаження. Суть останнього полягає в тому, що оператор перед початком роботи проводить короткочасне тренування щодо розв'язання однієї чи кількох задач підвищеної складності.

Фаза стійкої працездатності ($t_1 - t_2$) характеризується найвищою якістю праці при оптимальних рівнях функціонування фізіологічних систем організму. Тривалість цього періоду залежить від інтенсивності роботи. Чим інтенсивніша праця, тим коротший цей період. Найоптимальніша динамічна робота, коли цей період може бути в десятки разів довшим, ніж при статичній діяльності.

На процес стійкої працездатності великий вплив справляють емоції. Негативні (страх, невпевненість, поганий настрій) знижують працездатність. Позитивні (впевненість, спокій, бадьорий настрій) значно продовжують період стійкої працездатності.

Продовження періоду стійкої працездатності можна забезпечити:

- оптимальним рівнем напруги психофізіологічних функцій;
- комфортними умовами праці;
- правильним поєднанням режимів праці та відпочинку;
- емоційним розвантаженням;

					КС КРБ 123.213.00.00 ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

– використанням тонізуючих напоїв (кава, чай), фармакологічних засобів, зокрема препаратів рослинного походження (вітаміни, препарати, які впливають на енергетичні та метаболічні процеси);

– інформуванням людини про наслідки її діяльності, наглядом та контролем її роботи.

Практичний досвід свідчить, що вживання легких стимуляторів допомагає знизити сонливість, сприяє підвищенню працездатності на короткий період. Однак активні стимулятори на відповідальних видах робіт здатні викликати негативний ефект – погіршується самопочуття, знижується рухливість та швидкість реакцій. Поширене серед населення вживання транквілізаторів, викликаючи заспокоєння та запобігаючи розвитку неврозів, може знизити психічну активність, сповільнити реакції, спричинити апатію та сонливість.

Фаза субкомпенсації ($t_2 - t_3$) розглядається як початок розвитку втоми. В цей період якість праці ще зберігається на високому рівні, але тільки за рахунок перенапруги відповідних функцій організму.

Фаза втоми (з моменту у характеризується чітко вираженим зниженням якості роботи при подальшому погіршенні функціонального стану людини. Об'єктивними показниками втоми є зміна частоти пульсу, дихання, зорової та слухової чутливості.

Наступною фазою життєдіяльності людини повинна бути фаза відновлення працездатності (відпочинку), яка може тривати від 3 до 5 хвилин; 60 — 90 хв. і навіть декілька діб.

Раціональний режим праці та відпочинку передбачає дотримання певної тривалості безперервної роботи на персональному комп'ютері і перерв, регламентованих з урахуванням тривалості робочої зміни, виду трудової діяльності.

Для попередження передчасної стомлюваності операторів ПК рекомендується організувати робочу зміну шляхом чергування робіт з використанням персонального комп'ютера і без нього.

					КС КРБ 123.213.00.00 ПЗ	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

При виникненні у працюючих з ПК зорового дискомфорту та інших несприятливих суб'єктивних відчуттів, незважаючи на дотримання санітарно-гігієнічних і ергономічних вимог, рекомендується застосовувати індивідуальний підхід з обмеженням часу роботи з ПК.

У випадках, коли характер роботи вимагає постійної взаємодії з ВДТ (набір текстів або введення даних тощо) з напругою уваги та зосередженості, при виключенні можливості періодичного перемикавання на інші види трудової діяльності, не пов'язані з ПК, рекомендується організація перерв на 10 -15 хвилин через кожні 45-60 хвилин роботи. Тривалість безперервної роботи з ВДТ без регламентованого перерви не повинна перевищувати однієї години.

Сумарний час регламентованих перерв залежить від тривалості роботи, виду та категорії трудової діяльності з використанням ПК (табл. 4.1).

Таблиця 4.1 – Сумарний час регламентованих перерв залежно від тривалості роботи, виду та категорії трудової діяльності з ПК

Категорія роботи з ПК	Сумарний час регламентованих перерв, хв	
	при 8-годинній зміні	при 12-годинній зміні
I	50	80
II	70	110
III	90	140

Під час регламентованих перерв з метою зниження нервово-емоційного напруження, стомлення зорового аналізатора, усунення впливу гіподинамії та гіпокінезії, запобігання розвитку позотонічної (статичного) втоми доцільно виконувати спеціально розроблені комплекси вправ.

4.2 Вимоги ергономіки до організації робочого місця оператора ПК

Для збереження працездатності й попередження розвитку захворювань

опорно-рухового апарату операторів ПК необхідно організувати для них робочі місця, що відповідають вимогам ДСТУ.

Виконання цих вимог показані на рисунку 4.2 і приведено конструктивні особливості встановлюваних робочих столів і стільців, що забезпечують можливість індивідуального регулювання відповідно росту працюючих і створення для них зручної пози.

При правильній організації робочого місця продуктивність праці оператора зростає на 8 – 20%.

Відповідно до ДСТУ конструкція робочого місця й розташування всіх його елементів повинна відповідати антропометричним, фізичним і психологічним вимогам. Велике значення має також характер роботи. Зокрема, при організації робочого місця оператора ПК повинні бути дотримані наступні основні умови:

- оптимальне розміщення устаткування, що входить до складу робочого місця;
- достатній робочий простір, що дозволяє здійснювати всі необхідні рухи й переміщення;
- необхідно природне й штучне освітлення для виконання поставлених завдань;
- рівень акустичного шуму не повинен перевищувати допустимого значення.

Головними елементами робочого місця оператора є письмовий стіл і крісло. Основним робочим положенням є положення сидячи.

Робоча поза сидячи викликає мінімальне стомлення оператора ПК. Раціональне планування робочого місця передбачає чіткий порядок і сталість розміщення предметів, засобів праці й документації. Те, що потрібно для виконання робіт частіше, розташовано в зоні легкої досяжності робочого простору.

Моторне поле – простір робочого місця, у якому можуть здійснюватися рухові дії людини.

					КС КРБ 123.213.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		63

Максимальна зона досяжності рук – це частина моторного поля робочого місця, обмеженого дугами, описуваними максимально витягнутими руками при русі їх у плечовому суглобі.

Оптимальна зона – частина моторного поля робочого місця, обмеженого дугами, описуваними передпліччями при русі в ліктьових суглобах з опорою в точці ліктя й з відносно нерухомим плечем.

Зони досяжності рук у горизонтальній площині зображено на рисунку 4.2.

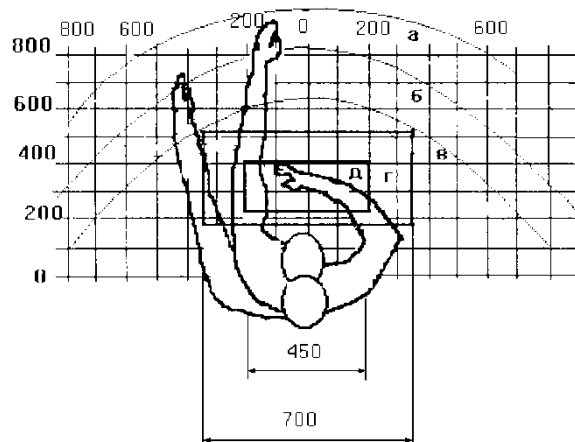


Рисунок 4.2 – Зони досяжності рук у горизонтальній площині

- а - зона максимальної досяжності;
- б - зона досяжності пальців при витягнутій руці;
- в - зона легкої досяжності долоні;
- г - оптимальний простір для грубої ручної роботи;
- д - оптимальний простір для тонкої ручної роботи.

Розглянемо оптимальне розміщення предметів праці й документації в зонах досяжності рук:

- монітор розміщується в зоні а (у центрі);
- клавіатура - у зоні г/д;
- системний блок розміщується в зоні б (ліворуч);
- принтер перебуває в зоні а (праворуч);

Змн.	Арк.	№ докум.	Підпис	Дата

КС КРБ 123.213.00.00 ПЗ

Арк.

64

– документація - у зоні легкої досяжності долоні - в (ліворуч) - література й документація, необхідна при роботі;

– висота стола повинна бути обрана з урахуванням можливості сидіти вільно, у зручній позі, при необхідності опираючись на підлокітники;

– нижня частина стола повинна бути сконструйована так, щоб користувач ПК міг зручно сидіти, не був змушений підтискати ноги;

– поверхня стола повинна мати властивості, що виключають появу відблисків у полі зору користувача;

– конструкція стола повинна передбачати наявність висувних ящиків (не менш 3 для зберігання документації, літератури, особистих речей).

Параметри робочого місця вибираються відповідно до антропометричних характеристик. При використанні цих даних у розрахунках варто виходити з максимальних антропометричних характеристик.

При роботі в положенні сидячи рекомендуються наступні параметри робочого простору:

– ширина - не менш 700мм;

– глибина - не менш 400мм;

– висота робочої поверхні стола над підлогою 700-750мм.

Оптимальними розмірами стола є:

– висота 710мм;

– довжина стола 1300мм;

– ширина стола 650мм.

Під робочою поверхнею має бути передбачений простір для ніг:

– висота - не менш 600мм;

– ширина - не менш 500мм;

– глибина - не менш 400мм.

Важливим елементом робочого місця оператора ПК є крісло. При проектуванні крісла виходять із того, що при будь-якому робочому положенні оператора його поза повинна бути фізіологічно правильною, тобто положення частин тіла повинне бути оптимальним. Для задоволення вимог фізіології, що

					КС КРБ 123.213.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		65

впливають із аналізу положення тіла людини в положенні сидячи, конструкція робочого сидіння повинна задовольняти наступним основним вимогам:

- допускати можливість зміни положення тіла, тобто забезпечувати вільне переміщення корпусу й кінцівок тіла;
- допускати регулювання висоти залежно від росту працюючої людини (у межах від 400 до 550мм);
- радіус кривизни в горизонтальній площині 400мм;
- кут нахилу спинки повинен змінюватися в межах 90-110° до площини сидіння.

					КС КРБ 123.213.00.00 ПЗ	Арк.
						66
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

Метою даної дипломної роботи була розробка комп'ютеризованої системи для обліку успішності та відвідування занять здобувачами освіти. Фізичним сервером для проекту став HP ProLiant DL160 G6 з ОС Windows Server 2012 R2, реалізація логічної серверної частини відбувалася на основі бази даних MySQL, програмна реалізація відбувалася на мові Java (платформа JavaFX) у середовищі IntelliJ IDEA.

За час виконання проекту було здобуто колосальну кількість досвіду, навичок та знань, що знадобляться для розробки майбутніх проектів. Вдалося попрацювати як і у колективі (при проектуванні КС разом із керівником дипломного проекту і представниками адміністрації навчального закладу), так і самостійно з потужними інструментами та методами для розробки програмних продуктів, а також, засобами для налаштування апаратної складової проекту.

Комп'ютеризована система розроблялась так, щоб максимально задовільнити потреби замовника (коледжу), а також, передбачалась можливість використання її і в інших подібних навчальних закладах.

Даний проект має величезні та практично безмежні перспективи і можливості щодо розвитку, вдосконалення та модифікації, частина з яких буде використана для реалізації нових функцій у найближчому майбутньому, після чого проект буде введено у повноцінну експлуатацію.

					КС КРБ 123.213.00.00 ПЗ	Арк.
						67
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Київський національний економічний університет імені Вадима Гетьмана. Журнал успішності студентів [Електронний ресурс]. – Режим доступу: kneu.edu.ua/ua/Information_for/students/jurnal/.
2. Офіційний веб-сайт освітнього ресурсу Moodle. Головна [Електронний ресурс]. – Режим доступу: moodle.org/?lang=uk.
3. Офіційний веб-сайт платформи E-schools. Електронні щоденники та журнали [Електронний ресурс]:. – Режим доступу: e-schools.info/e-service.
4. Офіційний веб-сайт компанії «Щоденник». Щоденник.ua [Електронний ресурс]. – Режим доступу: company.shodennik.ua service.
5. Офіційний веб-сайт Буковинського державного медичного університету. Електронний журнал успішності [Електронний ресурс]. – Режим доступу: www.bsmu.edu.ua/studentu/elektronniy-zhurnal-uspishnosti/.
6. Діденко О. В., Купрієнко Д. А. Інформаційні технології і засоби навчання: електронний журнал обліку успішності слухачів (курсантів, студентів) як засіб раціоналізації навчально-виховного процесу [Електронний ресурс]. – Том 47, №3. – 2015. – Режим доступу: www.google.com.ua/url?sa=t&rct=j&q=&esrc=s&source=web&cd=13&cad=rja&uact=8&ved=2ahUKEwj6pPCn9sToAhUml4sKHdz6CLsQFjAMegQIAxAB&url=https%3A%2F%2Fjournal.iitta.gov.ua%2Findex.php%2Fitlt%2Farticle%2Fview%2F1226%2F924&usg=AOvVaw3dg-f9VyOC4CDFskb-naVc.
7. Офіційний веб-сайт Національного педагогічного університету імені М.П. Драгоманова. Пс-журнал успішності-web [Електронний ресурс]. – Режим доступу: ntu.npu.edu.ua.
8. Прохоренко М. А. Основи Java / М. А. Прохоренко. – СПб.: БХВ-Петербург, 2017. –704 с.
9. Ютуб-канал «kenLovesToCode». Цикл відео JavaFX CRUD [Електронний ресурс]. – Режим доступу: www.youtube.com/channel/UCopWb4iPes0SDAVD1n7CQmw.

					КС КРБ 123.213.00.00 ПЗ	Арк.
						68
Змн.	Арк.	№ докум.	Підпис	Дата		

10. Бедрій Я. І. Безпека життєдіяльності. – К.: Київ, Кондор, 2009. 286 с.
11. Геврик Є.О. Охорона праці. – К.: Ельга, Ніка-Центр, 2003. 280 с.

					<i>КС КРБ 123.213.00.00 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		69

Додаток А.
Технічне завдання

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії

Кафедра комп'ютерних систем та мереж

“Затверджую”

Завідувач кафедри КС

_____Осухівська Г.М.

“___” _____2022 р

КОМП'ЮТЕРИЗОВАНА СИСТЕМА ОБЛІКУ УСПІШНОСТІ ТА
ВІДВІДУВАННЯ ЗАНЯТЬ ЗДОБУВАЧАМИ ОСВІТИ

ТЕХНІЧНЕ ЗАВДАННЯ

на 12 листках

Вид робіт:

Кваліфікаційна робота

На здобуття освітнього ступеня «Бакалавр»

Спеціальність 123 «Комп'ютерна інженерія»

«УЗГОДЖЕНО»

Керівник кваліфікаційної роботи

_____к.т.н., доц. Луцик Н.С.

«___» _____2022 р.

«ВИКОНАВЕЦЬ»

Студентка групи СІс-44

_____Дзюбак Ю.Ю.

«___» _____2022 р.

Тернопіль 2022

1 Загальні відомості

1.1 Повна назва та її умовне позначення

Повна назва теми кваліфікаційної роботи: «Комп'ютеризована система обліку успішності та відвідування занять здобувачами освіти».

Умовне позначення кваліфікаційної роботи: КС КРБ 123.213.00.00

1.2 Виконавець

Студент групи СІс-43, факультету комп'ютерно-інформаційних систем і програмної інженерії, кафедри комп'ютерних систем та мереж, Тернопільського національного технічного університету імені Івана Пулюя, Дзюбак Юрій Юрійович.

1.3 Підстава для виконання роботи

Підставою для виконання кваліфікаційної роботи є наказ по університету (№ 4.7-180 від 23.03.2022 р.)

1.4 Планові терміни початку та завершення роботи

Плановий термін початку виконання кваліфікаційної роботи – 23.03.2022 р.

Плановий термін завершення виконання кваліфікаційної роботи – 22.06.2022 р.

1.5 Порядок оформлення та пред'явлення результатів роботи

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

Пред'явлення проміжних результатів роботи з виконання кваліфікаційної роботи здійснюється у відповідності до графіку, затвердженого керівником роботи.

Попередній захист кваліфікаційної роботи відбувається при готовності роботи на 90% , наявності пояснювальної записки та графічного матеріалу.

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

2 Призначення і цілі створення системи

2.1 Призначення системи

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

При проектуванні комп'ютеризованої системи обліку успішності та відвідування занять здобувачами освіти необхідно дослідити існуючу апаратну інфраструктуру, спроектувати схему бази даних і клієнтське програмне забезпечення для зручного використання програмного забезпечення системи.

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

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

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

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

2.2 Мета створення системи

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

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

- інформаційний аналіз предметної області та виділення інформаційних об'єктів проекту;
- побудова логічної моделі даних системи;
- проектування реляційної бази даних;
- вибір СУБД;
- вибір середовища розробки застосунків (IDE);
- розробка застосунку;
- тестування системи, внесення корективів;
- створення документації до КС щодо порядку організації роботи з нею;
- кінцеве налаштування та введення в експлуатацію (за ініціативи керівництва).

2.3 Характеристика об'єкту

2.3.1 Основні задачі та функції об'єкту

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

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

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

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

Функції прикладного програмного додатку полягають у:

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

3 Вимоги до системи

3.1 Вимоги до системи в цілому

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

Вимогами до комп'ютерної мережі закладу освіти є:

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

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

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

3.1.1 Вимоги до структури та функціонування системи

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

- захищеність;
- надійність;
- продуктивність;
- відновлюваність та резервування.

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

1. Функціональність:

- формування довідників про типи навчальних занять і назви предметів;
- створення довідників груп, студентів, учителів;
- можливість введення, редагування та оновлення інформації про проведені заняття;
- здатність відображати успішність учнів/студентів;
- наявність засобів для впорядкування даних за відповідними полями бази даних;
- наявність механізмів управління авторизованим доступом;
- можливість створення звітів успішності здобувачів освіти;
- облік навантаження викладачів/учителів.

2. Продуктивність:

- час реакції на дії користувача до 1 с;
- час відповіді системи до 1 с;
- оптимальне використання ресурсів RAM;
- оптимальність використання ресурсів жорсткого диску;
- прозорість та доступність використання функціональності програмного забезпечення;
- налаштування політики прав доступу до системи обліку та контролю навчальних занять;
- гнучкість та масштабованість структури програмного забезпечення

комп'ютеризованої системи.

3. Надійність:

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

4. Зручність використання:

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

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

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

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

3.1.3 Вимоги по діагностуванню системи

Діагностика комп'ютеризованої системи обліку успішності та відвідування занять здобувачами освіти відбувається згідно регламенту та відповідного розкладу,

і передбачає перевірку параметрів працездатності компонентів локальної комп'ютерної мережі та прикладного програмного забезпечення.

3.1.4 Перспективи розвитку, модернізація системи

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

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

3.1.5 Вимоги до надійності системи

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

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

3.1.6 Вимоги до функцій та задач, які виконує система

Основними вимогами і функціями комп'ютеризованої системи обліку успішності та відвідування занять здобувачами освіти є:

- забезпечення комунікаційної інфраструктури при взаємодії клієнтського

програмного забезпечення з системою керування базами даних;

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

3.1.7 Вимоги до апаратного забезпечення

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

1. Активне комутаційне обладнання мережі.
2. Сервер баз даних:
 - процесор з тактовою частотою не менше 2,1 ГГц з 8 ядрами та 16 потоками;
 - розмір оперативної пам'яті – не менше 16 ГБ;
 - об'єм жорсткого диску - не менше 1 Тб.
3. Робочі станції:
 - процесор з тактовою частотою не менше 2,0 ГГц;
 - об'єм оперативної пам'яті - не менше 2 ГБ;
 - об'єм жорсткого диску – не менше 80 Гб.

3.1.8 Вимоги до програмного забезпечення

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

Операційна система робочих станцій рекомендована до використання – Windows 7+..

4 Вимоги до документації

Документація повинна відповідати вимогам ЄСКД та ДСТУ.

Комплект документації повинен складатись з:

- пояснювальної записки;
 - графічного матеріалу:
1. Логічна модель спроектованої бази даних.
 2. EER-діаграма створеної бази даних.
 3. Класова структура програмного забезпечення комп'ютеризованої системи.
 4. Алгоритм роботи програмного забезпечення комп'ютеризованої системи.

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

5 Стадії та етапи проектування

Таблиця 1 – Стадії та етапи виконання кваліфікаційної роботи бакалавра

№ етапу	Назва етапу виконання кваліфікаційної роботи	Термін виконання
1	Розробка та аналіз вимог технічного завдання	23.03-31.03
2	Аналіз організаційної структури закладу освіти	25.03-02.04
3	Проектування схеми бази даних та архітектури програмного забезпечення	02.04-14.04
4	Розробка програмного забезпечення комп'ютеризованої системи	14.04-15.05
5	Налаштування апаратної складової комп'ютеризованої системи	16.05-20.05
6	Тестування і виправлення помилок комп'ютеризованої системи	20.05-01.06
7	Безпека життєдіяльності, основи охорони праці	02.06-06.06
8	Оформлення кваліфікаційної роботи	06.06-12.06
9	Попередній захист кваліфікаційної роботи	12.06-17.06
10	Захист кваліфікаційної роботи	22.06-24.06

6 Додаткові умови виконання кваліфікаційної роботи

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

Додаток Б.

Лістинг програмного додатку

Main.java

```
import javafx.application.Application;
import javafx.event.EventHandler;
import javafx.fxml.FXMLLoader;
import javafx.geometry.Rectangle2D;
import javafx.scene.Scene;
import javafx.scene.image.Image;
import javafx.scene.input.MouseEvent;
import javafx.stage.Screen;
import javafx.stage.Stage;
import javafx.stage.StageStyle;
import authorization_and_dbconfigs.ControllerLoginPane;
public class Main extends Application {

    private FXMLLoader loader;
    private double xOffset;
    private double yOffset;

    @Override
    public void start(Stage primaryStage) {

        try {

            loader = new FXMLLoader();
            loader.setLocation(getClass().getResource("authorization_and_dbconfigs/login_pane.fxml"));
            ControllerLoginPane controller = new ControllerLoginPane();
            loader.setController(controller);
            loader.load();
            Scene scene = new Scene(loader.getRoot());
            scene.setOnMousePressed(new EventHandler<MouseEvent>()
            {
                @Override
                public void handle(MouseEvent event) {
                    xOffset = primaryStage.getX() -
event.getScreenX();
                    yOffset = primaryStage.getY() -
event.getScreenY();
                }
            });
            scene.setOnMouseDragged(new EventHandler<MouseEvent>()
            {
                @Override
                public void handle(MouseEvent event) {
                    primaryStage.setX(event.getScreenX() +
xOffset);
                    primaryStage.setY(event.getScreenY() +
yOffset);
                }
            });
        }
    }
}
```

```

        }
    });
    primaryStage.initStyle(StageStyle.UNDECORATED);
    primaryStage.setScene(scene);
    primaryStage.setResizable(false);
    primaryStage.setTitle("classbook v.1.0");
    primaryStage.getIcons().add(new
Image("/img/iconmini.jpg"));
    primaryStage.show();
    Rectangle2D primScreenBounds =
Screen.getPrimary().getVisualBounds();
    primaryStage.setX((primScreenBounds.getWidth() -
primaryStage.getWidth()) / 2);
    primaryStage.setY((primScreenBounds.getHeight() -
primaryStage.getHeight()) / 2);

        } catch(Exception e) {
            e.printStackTrace();
        }
    }
    public static void main(String[] args) {
        launch(args);
    }
}

```

DBConnection.java

```

package authorization_and_dbconfigs;
import java.io.FileInputStream;
import java.io.IOException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.Properties;
public class DBConnection {
    private final String dburl;
    private final String username = "*****";
    private final String password = "*****";
    private Connection connect;
    private String host;
    private String port;
    private String db;
    public DBConnection() {
        FileInputStream fis;
        Properties property = new Properties();
        String path = System.getProperty("user.dir");
        try {
            fis = new FileInputStream(path+"/config.properties");
            property.load(fis);
            host = property.getProperty("db.host");
            port = property.getProperty("db.port");

```

```

        db = property.getProperty("db.name");
    } catch (IOException e) {
        System.err.println("Error: Config file is not
allowed!");
    }
    dburl =
"jdbc:mysql://" + host + ":" + port + "/" + db + "?allowPublicKeyRetrieval=true&useUnicode=true&useJDBCCompliantTimezoneShift=true&useLegacyDate
timeCode=false&serverTimezone=UTC&useSSL=false";
    }
    public Connection getConnection() {
        try {
            connect = DriverManager.getConnection(dburl, username,
password);
        } catch (Exception e) {
            e.printStackTrace();
        }
        return connect;
    }
    public void close(Connection connect, PreparedStatement pstmt,
ResultSet rs) {
        try {
            if (connect != null)
                connect.close();
            if (pstmt != null)
                pstmt.close();
            if (rs != null)
                rs.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    public void close(Connection connect, PreparedStatement pstmt)
{
        try {
            close(connect, pstmt, null);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    public void close(PreparedStatement pstmt) {
        try {
            close(null, pstmt, null);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

DataAccessObject.java

package admin;

```

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import authorization_and_dbconfigs.DBConnection;
import vukladach.Ocinku_VidviduvanistDA;
import vukladach.ZaniattyaDA;
public class DataAccessObject {
    private DBConnection database = new DBConnection();
    private ResultSet rs;
    private PreparedStatement pstmt;
    private Connection connect;
    public DataAccessObject() {
    }
    public void saveData(String query) {
        try {
            connect = database.getConnection(); // get connection
            pstmt = connect.prepareStatement(query);
            pstmt.executeUpdate();
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            database.close(connect, pstmt, null);
        }
    }

    ////////////VUKLADACHI//////////
    public ObservableList<PosaduDA> getPosadaData(String query) {
        ObservableList<PosaduDA> list =
FXCollections.observableArrayList();
        try {
            connect = database.getConnection();
            pstmt = connect.prepareStatement(query);
            rs = pstmt.executeQuery();
            while(rs.next()) {
                list.add(new PosaduDA(rs.getInt(1),
rs.getString(2)));
            }
        } catch (Exception e) {
            e.printStackTrace();
        }

        return list;
    }

    public ObservableList<RolesDA> getRolesData(String query) {
        ObservableList<RolesDA> list =
FXCollections.observableArrayList();
        try {
            connect = database.getConnection();
            pstmt = connect.prepareStatement(query);
            rs = pstmt.executeQuery();

```

```

        while(rs.next()) {
            list.add(new RolesDA(rs.getInt(1),
rs.getString(2)));
        }
    }catch(Exception e) {
        e.printStackTrace();
    }

    return list;
}

    public ObservableList<VukladachiDA> getVuklAccountsData(String
query){
        ObservableList list = FXCollections.observableArrayList();
        try {
            connect = database.getConnection();
            pstmt = connect.prepareStatement(query);
            rs = pstmt.executeQuery();
            while(rs.next()) {
                list.add(new VukladachiDA(rs.getInt(1),
rs.getString(2), rs.getString(3), rs.getString(4),
rs.getString(5)));
            }
        }catch(Exception e) {
            e.printStackTrace();
        }

        return list;
    }

    public ObservableList<Vukl_roleDA>
getVukl_roleAccountsData(String query){
        ObservableList list = FXCollections.observableArrayList();
        try {
            connect = database.getConnection();
            pstmt = connect.prepareStatement(query);
            rs = pstmt.executeQuery();
            while(rs.next()) {
                list.add(new Vukl_roleDA(rs.getInt(1),
rs.getString(2), rs.getString(3), rs.getString(4)));
            }
        }catch(Exception e) {
            e.printStackTrace();
        }

        return list;
    }

    public ObservableList<String> getPosadaComboBox(String query){
        ObservableList list = FXCollections.observableArrayList();
        try {
            connect = database.getConnection();
            pstmt = connect.prepareStatement(query);

```

```

        rs = pstmt.executeQuery();
        while(rs.next()) {
            list.add(rs.getString(1));
        }
    }catch(Exception e) {
        e.printStackTrace();
    }

    return list;
}

public ObservableList<String> getRoleComboBox(String query){
    ObservableList list = FXCollections.observableArrayList();
    try {
        connect = database.getConnection();
        pstmt = connect.prepareStatement(query);
        rs = pstmt.executeQuery();
        while(rs.next()) {
            list.add(rs.getString(1));
        }
    }catch(Exception e) {
        e.printStackTrace();
    }

    return list;
}

//////////GROUPS//////////
public ObservableList<GroupsDA> getGroupsAccountsData(String
query){
    ObservableList list = FXCollections.observableArrayList();
    try {
        connect = database.getConnection();
        pstmt = connect.prepareStatement(query);
        rs = pstmt.executeQuery();
        while(rs.next()) {
            list.add(new GroupsDA(rs.getInt(1),
rs.getString(2), rs.getString(3), rs.getString(4)));
        }
    }catch(Exception e) {
        e.printStackTrace();
    }

    return list;
}

public ObservableList<String> getKuratorComboBox(String
query){
    ObservableList list = FXCollections.observableArrayList();
    try {
        connect = database.getConnection();
        pstmt = connect.prepareStatement(query);
        rs = pstmt.executeQuery();

```



```

        while(rs.next()) {
            list.add(rs.getString(1));
        }
    }catch(Exception e) {
        e.printStackTrace();
    }

    return list;
}

//////////STUDENTS//////////
public ObservableList<StudentsDA> getStudAccountsData(String
query){
    ObservableList list = FXCollections.observableArrayList();
    try {
        connect = database.getConnection();
        pstmt = connect.prepareStatement(query);
        rs = pstmt.executeQuery();
        while(rs.next()) {
            list.add(new StudentsDA(rs.getInt(1),
rs.getString(2), rs.getString(3), rs.getString(4),
rs.getString(5), rs.getString(6), rs.getString(7),
rs.getString(8)));
        }
    }catch(Exception e) {
        e.printStackTrace();
    }

    return list;
}

public ObservableList<SubgroupsDA> getSubgroupData(String
query){
    ObservableList<SubgroupsDA> list =
FXCollections.observableArrayList();
    try {
        connect = database.getConnection();
        pstmt = connect.prepareStatement(query);
        rs = pstmt.executeQuery();
        while(rs.next()) {
            list.add(new SubgroupsDA(rs.getInt(1),
rs.getString(2)));
        }
    }catch(Exception e) {
        e.printStackTrace();
    }

    return list;
}

public ObservableList<String> getSubgroupComboBox(String
query){
    ObservableList list = FXCollections.observableArrayList();

```

```

    try {
        connect = database.getConnection();
        pstmt = connect.prepareStatement(query);
        rs = pstmt.executeQuery();
        while(rs.next()) {
            list.add(rs.getString(1));
        }
    }catch(Exception e) {
        e.printStackTrace();
    }

    return list;
}

public ObservableList<String> getGroupComboBox(String query){
    ObservableList list = FXCollections.observableArrayList();
    try {
        connect = database.getConnection();
        pstmt = connect.prepareStatement(query);
        rs = pstmt.executeQuery();
        while(rs.next()) {
            list.add(rs.getString(1));
        }
    }catch(Exception e) {
        e.printStackTrace();
    }

    return list;
}

...
}

```

ControllerZaniattya.java

```

package vukladach;
import admin.DataAccessObject;
import com.itextpdf.text.*;
import com.itextpdf.text.pdf.BaseFont;
import com.itextpdf.text.pdf.PdfPCell;
import com.itextpdf.text.pdf.PdfPTable;
import com.itextpdf.text.pdf.PdfWriter;
import com.jfoenix.controls.JFXButton;
import com.jfoenix.controls.JFXDatePicker;
import javafx.beans.property.ReadOnlyObjectWrapper;
import javafx.event.EventHandler;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.fxml.Initializable;
import javafx.geometry.Rectangle2D;
import javafx.scene.Scene;

```

```

import javafx.scene.control.*;
import javafx.scene.control.Label;
import javafx.scene.control.TextArea;
import javafx.scene.control.TextField;
import javafx.scene.image.Image;
import javafx.scene.image.ImageView;
import javafx.scene.input.MouseButton;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.AnchorPane;
import javafx.scene.layout.Pane;
import javafx.scene.shape.Circle;
import javafx.stage.Screen;
import javafx.stage.Stage;
import javafx.stage.StageStyle;
import authorization_and_dbconfigs.ControllerPersonal_account;
import authorization_and_dbconfigs.DBConnection;
import javax.swing.*;
import java.awt.*;
import java.awt.Font;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.net.URL;
import java.sql.Connection;
import java.sql.Date;
import java.sql.ResultSet;
import java.sql.Statement;
import java.time.LocalDate;
import java.time.format.DateTimeFormatter;
import java.util.*;

public class ControllerZaniattya extends Component implements
Initializable {

    ...

    private void AddToPDF() throws IOException, DocumentException
    {

        String path = "";
        JFileChooser j = new JFileChooser();
        j.setFileSelectionMode(JFileChooser.DIRECTORIES_ONLY);
        int x = j.showSaveDialog(this);

        if (x == JFileChooser.APPROVE_OPTION) {
            path = j.getSelectedFile().getPath();
        }

        Document doc = new Document();
        String docname =
ControllerSelection_of_load.getpDiscname()+"
"+ControllerSelection_of_load.getpGroupname()+" Права сторінка
"+VuklName+".pdf";

```

```

        PdfWriter.getInstance(doc, new FileOutputStream(path +
docname));
        doc.open();
        BaseFont bf =
BaseFont.createFont("c:/windows/fonts/times.ttf",
BaseFont.IDENTITY_H, BaseFont.EMBEDDED);

        Chunk chunk = new Chunk("Викладач: "+ VuklName, new
com.itextpdf.text.Font(bf, 12, Font.BOLD, BaseColor.BLACK));

        Paragraph paragraph = new Paragraph(0);
        paragraph.setSpacingAfter(10);
        paragraph.add(chunk);
        paragraph.setAlignment(Element.ALIGN_LEFT);

        PdfPTable tbl1 = new PdfPTable(8);

        tbl1.setTotalWidth(550);
        tbl1.setLockedWidth(true);

        tbl1.setWidths(new int[]{5, 17, 12, 9, 45, 30, 15, 20});
        for (int i = 0; i < 49; i++) {
            for (int k = 0; k < 8; k++) {
                tbl1.addCell(new PdfPCell(new Phrase(" ", new
com.itextpdf.text.Font(bf, 11, Font.TRUETYPE_FONT,
BaseColor.BLACK))));
            }
        }

        int used = 0;
        int n = 0, height = 0;

        do {
            doc.add(paragraph);
            PdfPTable tbl = new PdfPTable(8);

            tbl.setTotalWidth(550);
            tbl.setLockedWidth(true);

            tbl.setWidths(new int[]{5, 17, 12, 9, 45, 30, 15,
20});

            PdfPCell cell = new PdfPCell(new PdfPCell(new
Phrase("№"+"\\n"+"з/п", new com.itextpdf.text.Font(bf, 11,
Font.BOLD, BaseColor.BLACK))));
            cell.setHorizontalAlignment(Element.ALIGN_CENTER);
            tbl.addCell(cell);

            cell = new PdfPCell(new PdfPCell(new
Phrase("Дата", new com.itextpdf.text.Font(bf, 11, Font.BOLD,
BaseColor.BLACK))));
            cell.setHorizontalAlignment(Element.ALIGN_CENTER);
            tbl.addCell(cell);

```

```

        cell = new PdfPCell(new PdfPCell(new
Phrase("Вид"+"\\n"+"заняття", new com.itextpdf.text.Font(bf, 11,
Font.BOLD, BaseColor.BLACK)));
        cell.setHorizontalAlignment(Element.ALIGN_CENTER);
tbl.addCell(cell);

        cell = new PdfPCell(new PdfPCell(new Phrase("К-
сть"+"\\n"+"год.", new com.itextpdf.text.Font(bf, 11, Font.BOLD,
BaseColor.BLACK)));
        cell.setHorizontalAlignment(Element.ALIGN_CENTER);
tbl.addCell(cell);

        cell = new PdfPCell(new PdfPCell(new Phrase("Тема
заняття", new com.itextpdf.text.Font(bf, 11, Font.BOLD,
BaseColor.BLACK)));
        cell.setHorizontalAlignment(Element.ALIGN_CENTER);
tbl.addCell(cell);

        cell = new PdfPCell(new PdfPCell(new
Phrase("Домашнє завдання", new com.itextpdf.text.Font(bf, 11,
Font.BOLD, BaseColor.BLACK)));
        cell.setHorizontalAlignment(Element.ALIGN_CENTER);
tbl.addCell(cell);

        cell = new PdfPCell(new PdfPCell(new
Phrase("Підпис"+"\\n"+"викладача", new com.itextpdf.text.Font(bf,
11, Font.BOLD, BaseColor.BLACK)));
        cell.setHorizontalAlignment(Element.ALIGN_CENTER);
tbl.addCell(cell);

        cell = new PdfPCell(new PdfPCell(new
Phrase("Примітки", new com.itextpdf.text.Font(bf, 11, Font.BOLD,
BaseColor.BLACK)));
        cell.setHorizontalAlignment(Element.ALIGN_CENTER);
tbl.addCell(cell);

        for (int i = used; i < 9999; i++)
        {
            if (i < tblview.getItems().size() &&
used+11>=tblview.getItems().size()) {
                String N =
tblview.getColumns().get(1).getCellObservableValue(i).getValue().t
oString();

                String D =
tblview.getColumns().get(2).getCellObservableValue(i).getValue().t
oString();

                String V =
tblview.getColumns().get(3).getCellObservableValue(i).getValue().t
oString();

                String K =
tblview.getColumns().get(4).getCellObservableValue(i).getValue().t

```

```

oString();
                String T =
tblview.getColumns().get(5).getCellObservableValue(i).getValue().t
oString();
                String Dz =
tblview.getColumns().get(6).getCellObservableValue(i).getValue().t
oString();
                String P =
tblview.getColumns().get(7).getCellObservableValue(i).getValue().t
oString();
                String Pr =
tblview.getColumns().get(8).getCellObservableValue(i).getValue().t
oString();

                tbl.addCell(new PdfPCell(new Phrase(N, new
com.itextpdf.text.Font(bf, 11, Font.TRUETYPE_FONT,
BaseColor.BLACK))));
                tbl.addCell(new PdfPCell(new Phrase(D, new
com.itextpdf.text.Font(bf, 11, Font.TRUETYPE_FONT,
BaseColor.BLACK))));
                tbl.addCell(new PdfPCell(new Phrase(V, new
com.itextpdf.text.Font(bf, 11, Font.TRUETYPE_FONT,
BaseColor.BLACK))));
                tbl.addCell(new PdfPCell(new Phrase(K, new
com.itextpdf.text.Font(bf, 11, Font.TRUETYPE_FONT,
BaseColor.BLACK))));
                tbl.addCell(new PdfPCell(new Phrase(T, new
com.itextpdf.text.Font(bf, 11, Font.TRUETYPE_FONT,
BaseColor.BLACK))));
                tbl.addCell(new PdfPCell(new Phrase(Dz,
new com.itextpdf.text.Font(bf, 11, Font.TRUETYPE_FONT,
BaseColor.BLACK))));
                tbl.addCell(new PdfPCell(new Phrase(P, new
com.itextpdf.text.Font(bf, 11, Font.TRUETYPE_FONT,
BaseColor.BLACK))));
                tbl.addCell(new PdfPCell(new Phrase(Pr,
new com.itextpdf.text.Font(bf, 11, Font.TRUETYPE_FONT,
BaseColor.BLACK))));

                n++;
                height = (int) (height +
tbl.getRowHeight(n));
        }
        else if (i < used+11 &&
used+11<tblview.getItems().size()){
                String N =
tblview.getColumns().get(1).getCellObservableValue(i).getValue().t
oString();

                String D =
tblview.getColumns().get(2).getCellObservableValue(i).getValue().t
oString();

                String V =

```

```

tblview.getColumns().get(3).getCellObservableValue(i).getValue().t
oString();
        String K =
tblview.getColumns().get(4).getCellObservableValue(i).getValue().t
oString();
        String T =
tblview.getColumns().get(5).getCellObservableValue(i).getValue().t
oString();
        String Dz =
tblview.getColumns().get(6).getCellObservableValue(i).getValue().t
oString();
        String P =
tblview.getColumns().get(7).getCellObservableValue(i).getValue().t
oString();
        String Pr =
tblview.getColumns().get(8).getCellObservableValue(i).getValue().t
oString();

tbl.addCell(new PdfPCell(new Phrase(N, new
com.itextpdf.text.Font(bf, 11, Font.TRUETYPE_FONT,
BaseColor.BLACK))));
tbl.addCell(new PdfPCell(new Phrase(D, new
com.itextpdf.text.Font(bf, 11, Font.TRUETYPE_FONT,
BaseColor.BLACK))));
tbl.addCell(new PdfPCell(new Phrase(V, new
com.itextpdf.text.Font(bf, 11, Font.TRUETYPE_FONT,
BaseColor.BLACK))));
tbl.addCell(new PdfPCell(new Phrase(K, new
com.itextpdf.text.Font(bf, 11, Font.TRUETYPE_FONT,
BaseColor.BLACK))));
tbl.addCell(new PdfPCell(new Phrase(T, new
com.itextpdf.text.Font(bf, 11, Font.TRUETYPE_FONT,
BaseColor.BLACK))));
tbl.addCell(new PdfPCell(new Phrase(Dz,
new com.itextpdf.text.Font(bf, 11, Font.TRUETYPE_FONT,
BaseColor.BLACK))));
tbl.addCell(new PdfPCell(new Phrase(P, new
com.itextpdf.text.Font(bf, 11, Font.TRUETYPE_FONT,
BaseColor.BLACK))));
tbl.addCell(new PdfPCell(new Phrase(Pr,
new com.itextpdf.text.Font(bf, 11, Font.TRUETYPE_FONT,
BaseColor.BLACK))));
    }
    else if (tbl1.getTotalHeight() >
tbl.getTotalHeight()){
tbl.addCell(new PdfPCell(new Phrase(" ",
new com.itextpdf.text.Font(bf, 11, Font.TRUETYPE_FONT,
BaseColor.BLACK))));
tbl.addCell(new PdfPCell(new Phrase(" ",
new com.itextpdf.text.Font(bf, 11, Font.TRUETYPE_FONT,
BaseColor.BLACK))));
tbl.addCell(new PdfPCell(new Phrase(" ",

```

```

new com.itextpdf.text.Font(bf, 11, Font.TRUETYPE_FONT,
BaseColor.BLACK)));
        tbl.addCell(new PdfPCell(new Phrase(" ",
new com.itextpdf.text.Font(bf, 11, Font.TRUETYPE_FONT,
BaseColor.BLACK))););
        tbl.addCell(new PdfPCell(new Phrase(" ",
new com.itextpdf.text.Font(bf, 11, Font.TRUETYPE_FONT,
BaseColor.BLACK))););
        tbl.addCell(new PdfPCell(new Phrase(" ",
new com.itextpdf.text.Font(bf, 11, Font.TRUETYPE_FONT,
BaseColor.BLACK))););
        tbl.addCell(new PdfPCell(new Phrase(" ",
new com.itextpdf.text.Font(bf, 11, Font.TRUETYPE_FONT,
BaseColor.BLACK))););
        tbl.addCell(new PdfPCell(new Phrase(" ",
new com.itextpdf.text.Font(bf, 11, Font.TRUETYPE_FONT,
BaseColor.BLACK))););
    }
}

doc.add(tbl);
doc.newPage();

used = used + 11;
n = 0;

if (tblview.getItems().size() - used <=0){
    break;
}

} while (true);

doc.close();
Desktop desktop = Desktop.getDesktop();
desktop.open(new File(path+docname));

}

...

}

```

LeftSide.java

```

package vukladach;

import authorization_and_dbconfigs.DBConnection;
import com.itextpdf.text.*;
import com.itextpdf.text.pdf.BaseFont;
import com.itextpdf.text.pdf.PdfPCell;
import com.itextpdf.text.pdf.PdfPTable;
import com.itextpdf.text.pdf.PdfWriter;

```



```

import java.awt.Font;
import java.awt.Desktop;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.net.URL;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;

import javax.swing.*;
import javax.swing.table.DefaultTableCellRenderer;
import javax.swing.table.TableColumn;

public class LeftSide extends JPanel {

    private JFrame frame1;
    private JTable table1;

    private DBConnection database;
    private Connection connection;

    public LeftSide() throws IOException, DocumentException {

        frame1 = new JFrame();
        frame1.setTitle(ControllerSelection_of_load.getpDiscname()
+ " " + ControllerSelection_of_load.getpGroupname());

        try {

            database = new DBConnection();
            connection = database.getConnection();
            Statement statement = connection.createStatement();
            Statement statement2 = connection.createStatement();

            String sql0 = "SELECT FK_student FROM
dovidnyuk_student_marks WHERE FK_zaniattya='" +
ControllerZaniattya.getpID_zaniattya() + "' ";
            statement.execute(sql0);
            ResultSet result = statement.getResultSet();
            int id = 0;
            while (result.next()) {
                id++;
            }

            String sql1 = "SELECT PK_zaniattya FROM
dovidnyuk_zaniat WHERE FK_load='" +
ControllerSelection_of_load.getpID_load() + "' ";
            statement.execute(sql1);

```

```

result = statement.getResultSet();
int id2 = 2;
while (result.next()) {
    id2 = id2 + 2;
}

String[][] data = new String[id][id2];
ArrayList<String> dates = new ArrayList<String>();
dates.add("№");
dates.add("ПИБ студента");
String sqlx = "SELECT a.FK_vudy_zaniattya, a.date,
s.compact_name_vudy FROM dovidnyuk_zaniat as a JOIN
dovidnyuk_vudiv_zaniat as s ON
s.id_vudy_zaniattya=a.FK_vudy_zaniattya " +
"WHERE a.FK_load='" +
ControllerSelection_of_load.getpID_load() + "' ORDER BY
PK_zaniattya";

statement.execute(sqlx);
result = statement.getResultSet();
String name, compactname, name_v;
int lenght;
int id_v;
while (result.next()) {
    id_v = result.getInt("FK_vudy_zaniattya");
    name_v = result.getString("compact_name_vudy");

    if (id_v > 2) {
        dates.add(name_v);
        dates.add(name_v);
    }

    else {
        name = result.getString("date");
        lenght = name.length();
        compactname = name.substring(5, lenght);
        compactname = compactname.replace("-",
", ".");

        dates.add(compactname);
        dates.add(compactname);
    }
}

Object[] columnNames = dates.toArray();
table1 = new JTable(data, columnNames);

table1.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);
DefaultTableCellRenderer centerRend = new
DefaultTableCellRenderer();
centerRend.setHorizontalAlignment(JLabel.CENTER);
table1.setDefaultRenderer(Object.class, centerRend);

```

```

TableColumn a = table1.getColumnModel().getColumn(0);
a.setPreferredWidth(20);
TableColumn b = table1.getColumnModel().getColumn(1);
b.setPreferredWidth(150);

String sql;

    sql = "SELECT a.Namestudent FROM
dovidnyuk_student_marks as a " +
        "JOIN dovidnyuk_students as s ON
a.FK_student=s.id_student JOIN dovidnyuk_marks as d ON
a.FK_mark=d.PK_mark JOIN dovidnyuk_group as g ON
s.fk_group=g.id_group " +
        "JOIN dovidnyuk_vidvidyvanosti as f ON
a.FK_vidviduvanosti=f.id_pomitku_vidvidyv WHERE a.FK_zaniattya='"
+ ControllerZaniattya.getpID_zaniattya() + "' " +
        "AND g.group_name='" +
ControllerSelection_of_load.getpGroupname() + "' ORDER BY
a.FK_student";

statement.execute(sql);
ResultSet rec = statement.getResultSet();
int i = 0, n = 1;

while (rec.next()) {
    data[i][0] = Integer.toString(n);
    data[i][1] = rec.getString("Namestudent");
    i++;
    n++;
}

int j = 2, k = 3;

do {

    String sql3 = "SELECT PK_zaniattya FROM
dovidnyuk_zaniat WHERE FK_load='" +
ControllerSelection_of_load.getpID_load() + "' ORDER BY
PK_zaniattya"; //////////////!!!!!!!
    statement.execute(sql3);
    result = statement.getResultSet();
    int PK;
    while (result.next()) {
        PK = result.getInt("PK_zaniattya");
        String sql2;

        sql2 = "SELECT d.digital_name, f.compact_name
FROM dovidnyuk_student_marks as a " +
            "JOIN dovidnyuk_students as s ON
a.FK_student=s.id_student JOIN dovidnyuk_marks as d ON

```

```

a.FK_mark=d.PK_mark JOIN dovidnyuk_group as g ON
s.fk_group=g.id_group " +
        "JOIN dovidnyuk_zaniat as h ON
a.FK_zaniattya=h.PK_zaniattya JOIN dovidnyuk_group_load as k ON '"
+ ControllerSelection_of_load.getpID_load() + "'=k.PK_load JOIN
dovidnyuk_discipline as j ON k.FK_discipline=j.PK_discipline " +
        "JOIN dovidnyuk_vidvidyvanosti as f ON
a.FK_vidviduvanosti=f.id_pomitku_vidvidyv WHERE h.FK_load='" +
ControllerSelection_of_load.getpID_load() + "' AND
h.PK_zaniattya='" + PK + "' AND g.group_name='" +
ControllerSelection_of_load.getpGroupname() + "' ORDER BY
a.FK_student";

```

```

        statement2.execute(sql2);
        ResultSet rec2 = statement2.getResultSet();
        i = 0;
        while (rec2.next()) {
            data[i][j] =
rec2.getString("digital_name");
            data[i][k] =
rec2.getString("compact_name");
            i++;
        }
        k = k + 2;
        j = j + 2;
    }
}

```

```
break;
```

```
} while (true);
```

```

rec.close();
result.close();
statement.close();
statement2.close();
connection.close();

```

```

} catch (SQLException el) {
    el.printStackTrace();
}

```

```

JScrollPane spanel = new JScrollPane(table1);
frame1.add(spanel);
URL iconURL =
getClass().getResource("/img/iconmini.jpg");
ImageIcon icon = new ImageIcon(iconURL);
frame1.setIconImage(icon.getImage());
frame1.setSize(1200, 470);
frame1.setVisible(false);
AddToPDF();
}

```

```

private void AddToPDF() throws IOException, DocumentException
{
    String path = "";
    JFileChooser j = new JFileChooser();
    j.setFileSelectionMode(JFileChooser.DIRECTORIES_ONLY);
    int x = j.showSaveDialog(this);

    if (x == JFileChooser.APPROVE_OPTION) {
        path = j.getSelectedFile().getPath();
    }

    Document doc = new Document();
    String docname =
ControllerSelection_of_load.getpDiscname()+"
"+ControllerSelection_of_load.getpGroupname()+" Ліва сторінка
"+ControllerZaniattya.getpVuklName()+".pdf";
    PdfWriter.getInstance(doc, new
FileOutputStream(path+docname));
    doc.open();
    BaseFont bf =
BaseFont.createFont("c:/windows/fonts/times.ttf",
BaseFont.IDENTITY_H, BaseFont.EMBEDDED);

    Chunk chunk = new Chunk("Назва дисципліни: " +
ControllerSelection_of_load.getpDiscname(), new
com.itextpdf.text.Font(bf, 12, Font.BOLD, BaseColor.BLACK));

    Paragraph paragraph = new Paragraph(0);
    paragraph.setSpacingAfter(10);
    paragraph.add(chunk);

    doc.add(paragraph);

    PdfPTable tbl = new PdfPTable(24);

    tbl.setTotalWidth(550);
    tbl.setLockedWidth(true);

    tbl.setWidths(new float[]{3, 20, (float) 2.5, (float) 2.5,
(float) 2.5, (float) 2.5, (float) 2.5, (float) 2.5, (float) 2.5,
(float) 2.5, (float) 2.5, (float) 2.5, (float) 2.5,
(float) 2.5, (float) 2.5, (float) 2.5, (float) 2.5, (float)
2.5, (float) 2.5, (float) 2.5, (float) 2.5, (float) 2.5, (float)
2.5, (float) 2.5, (float) 2.5});

    PdfPCell cell = new PdfPCell(new PdfPCell(new PdfPCell(new Phrase("№" +
"\n" + "з/п", new com.itextpdf.text.Font(bf, 10, Font.BOLD,
BaseColor.BLACK))));
    cell.setHorizontalAlignment(Element.ALIGN_CENTER);
    tbl.addCell(cell);

    cell = new PdfPCell(new PdfPCell(new PdfPCell(new Phrase("ПІБ", new

```

```

com.itextpdf.text.Font(bf, 11, Font.BOLD, BaseColor.BLACK)));
    cell.setHorizontalAlignment(Element.ALIGN_CENTER);
    cell.setNoWrap(true);
    tbl.addCell(cell);

    String N;

    for (int l = 2; l < 24; l++) {
        if (l < table1.getColumnCount()) {
            N = table1.getColumnName(l);
            cell = new PdfPCell(new PdfPCell(new Phrase(N, new
com.itextpdf.text.Font(bf, 10, Font.BOLD, BaseColor.BLACK))));
            cell.setHorizontalAlignment(Element.ALIGN_CENTER);
            tbl.addCell(cell);
        }
        else {
            cell = new PdfPCell(new Phrase(" ", new
com.itextpdf.text.Font(bf, 11, Font.TRUETYPE_FONT,
BaseColor.BLACK)));
            tbl.addCell(cell);
        }
    }

    for (int i = 0; i < 49; i++) {
        for (int l = 0; l < 24; l++) {
            if (l < table1.getColumnCount() && i <
table1.getRowCount()) {
                N = table1.getValueAt(i, l).toString();
                cell = new PdfPCell(new Phrase(N, new
com.itextpdf.text.Font(bf, 11, Font.TRUETYPE_FONT,
BaseColor.BLACK)));
                if (l != 1)
cell.setHorizontalAlignment(Element.ALIGN_CENTER);
                tbl.addCell(cell);
            }
            else {
                cell = new PdfPCell(new Phrase(" ", new
com.itextpdf.text.Font(bf, 11, Font.TRUETYPE_FONT,
BaseColor.BLACK)));
                tbl.addCell(cell);
            }
        }
    }

    doc.add(tbl);

    if (table1.getColumnCount() > 24){

        int used = 22, b = 0, g = 0;
        int lim;

        do {

```

```

lim = table1.getColumnCount() - used - 2;
if (lim>=22){
    lim = 22;
}
else lim = lim + 2;

doc.newPage();
doc.add(paragraph);

PdfPTable tbl1 = new PdfPTable(24);

tbl1.setTotalWidth(550);
tbl1.setLockedWidth(true);

tbl1.setWidths(new float[]{3, 20, (float) 2.5,
(float) 2.5, (float) 2.5, (float) 2.5, (float) 2.5,
(float) 2.5, (float) 2.5, (float) 2.5, (float) 2.5, (float) 2.5,
(float) 2.5, (float) 2.5, (float) 2.5, (float) 2.5, (float) 2.5,
(float) 2.5, (float) 2.5, (float) 2.5});

cell = new PdfPCell(new PdfPCell(new Phrase("№" +
"\n" + "з/п", new com.itextpdf.text.Font(bf, 10, Font.BOLD,
BaseColor.BLACK))));
cell.setHorizontalAlignment(Element.ALIGN_CENTER);
tbl1.addCell(cell);

cell = new PdfPCell(new PdfPCell(new Phrase("ПИБ",
new com.itextpdf.text.Font(bf, 11, Font.BOLD, BaseColor.BLACK))));
cell.setHorizontalAlignment(Element.ALIGN_CENTER);
cell.setNoWrap(true);
tbl1.addCell(cell);

for (int l = used + 2; l < used + 2 + 22; l++) {
    if (l < table1.getColumnCount()) {
        N = table1.getColumnName(l);
        cell = new PdfPCell(new PdfPCell(new
Phrase(N, new com.itextpdf.text.Font(bf, 10, Font.BOLD,
BaseColor.BLACK))));
cell.setHorizontalAlignment(Element.ALIGN_CENTER);
tbl1.addCell(cell);
    } else {
        cell = new PdfPCell(new Phrase(" ", new
com.itextpdf.text.Font(bf, 11, Font.TRUETYPE_FONT,
BaseColor.BLACK))));
tbl1.addCell(cell);
    }
}

for (int i = 0; i < 49; i++) {
    for (int l = 0; l < used + 2 + lim; l++) {

```

```

        if (l < table1.getColumnCount() && i <
table1.getRowCount()) {
            if (lim > b && l < 2 || l >= used + 2)
{
                N = table1.getValueAt(i,
l).toString();
                cell = new PdfPCell(new Phrase(N,
new com.itextpdf.text.Font(bf, 11, Font.TRUETYPE_FONT,
BaseColor.BLACK)));
                if (l != 1)
cell.setHorizontalAlignment(Element.ALIGN_CENTER);
                tbl1.addCell(cell);
                b++;
                if (b == 24) b=0;
            } else if (b == lim && 24 - lim > g) {
                cell = new PdfPCell(new Phrase("
", new com.itextpdf.text.Font(bf, 11, Font.TRUETYPE_FONT,
BaseColor.BLACK)));
                tbl1.addCell(cell);
                g++;
            } else if (24 - lim == g) {
                b = 0;
                g = 0;
                l = -1;
            }
            } else if (table1.getColumnCount() > 1 &&
tbl.getRows().size() > tbl1.getRows().size()) {
                cell = new PdfPCell(new Phrase(" ",
new com.itextpdf.text.Font(bf, 11, Font.TRUETYPE_FONT,
BaseColor.BLACK)));
                tbl1.addCell(cell);
            }
        }
    }

    doc.add(tbl1);

    used = used + 22;
    b = 0; g = 0;

    if (table1.getColumnCount() - used - 2 <= 0) {
        break;
    }

} while (true);
}

doc.close();
frame1.dispose();
Desktop desktop = Desktop.getDesktop();
desktop.open(new File(path+docname));
}
}

```