Міністерство освіти і науки України
Криворізький національний університет
Кафедра моделювання та програмного забезпечення

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти бакалавра

з напряму підготовки 121 «Інженерія програмного забезпечення»

На тему: *Розробка програмного забезпечення обліку та контролю успішності школярів*

Засвідчую, що в цій кваліфікаційній роботі немає запозичень із праць інших авторів без відповідних посилань.
Студент гр. ІПЗ-20-2
_____ *Колтунова Є.Є.*

Керівник кваліфікаційної роботи _____ */ Гриценко А. М. /*

Завідувач кафедри _____ */ Стрюк А. М. /*

Кривий Ріг
2024

Криворізький національний університет

Факультет: *Інформаційних технологій*

Кафедра: *Моделювання та програмного забезпечення*

Освітньо-кваліфікаційний рівень: *бакалавр*

Спеціальність: *121 "Інженерія програмного забезпечення"*

ЗАТВЕРДЖУЮ
Зав. кафедри
_____*Стрюк А. М.*
«___»_____ 2024_ р.

ЗАВДАННЯ
на кваліфікаційну роботу

студента групи ІПЗ-20-2 Колтунова Єлизавета Євгенівна

1. Тема: *Розробка програмного забезпечення обліку та контролю успішності школярів* затверджено наказом по КНУ №___ від «__» лютого 2024 р.

2. Термін подання студентом закінченого проекту «10» червня 2024 р.

3. Вихідні дані по роботі: *Пояснювальна записка: 69 сторінок, 34 рисунків, 2 додатки, 19 використаних у роботі джерел*

4. Зміст пояснювальної записки (перелік питань, що їх треба розробити): .

5. Перелік графічного демонстраційного матеріалу: Приклади існуючих аналогів програм. Функціональна схема. . Генералізована структура бази даних. Інтерфейси роботи програми програмного застосунку.

Календарний план:

№	Найменування етапів кваліфікаційної роботи	Термін виконання етапів роботи
1	*Формулювання мети та задач роботи*	*13.02.2024-20.02.2024*
2	*Аналіз існуючих розробок та інформаційних джерел предметної області*	*21.02.2024-09.03.2024*
3	*Визначення вимог до програмного продукту*	*10.03.2024-18.03.2024*
4	*Розробка функціональної схеми, її опис*	*19.03.2024-23.03.2024*
5	*Розробка узагальненої структури бази даних*	*24.03.2024-31.03.2024*
6	*Розробка алгоритмів роботи*	*01.04.2024-14.04.2024*
7	*Розробка програмного забезпечення*	*15.04.2024-13.05.2024*
8	*Тестування програмного забезпечення*	*14.05.2024-20.05.2024*
9	*Оформлення пояснювальної записки*	*21.05.2024-12.06.2024*
10	*Розробка демонстраційних матеріалів*	*13.06.2024-17.06.2024*

Дата видачі завдання: «_____»_____ 2024 р.
Студент _____ Колтунова Є.Є.
Керівник роботи _____ Гриценко А. М.

РЕФЕРАТ

АВТОМАТИЗОВАНА СИСТЕМА, ОСВІТНІЙ ПРОЦЕС, ОБЛІК НАВЧАННЯ, АВТОМАТИЗАЦІЯ ДОКУМЕНТООБІГУ, ПРОГРАМУВАННЯ, PYTHON.

Пояснювальна записка: 69 с., 34 рис., 9 джерел.

В кваліфікаційній роботі розроблено та реалізовано програмний продукт автоматизованої інформаційно-довідкової системи обліку та контролю успішності школярів.

Виконано аналітичний огляд існуючих систем контролю успішності навчання. Розглянуто недоліки та переваги відомих розробок та технологій, представлених широкому загалу. Визначено задання розробки автоматизованої системи, що передбачає її універсальність, гнучкість, та відповідність вимогам, особливу увагу приділено простоті використання та підтримки.

Реалізація інформаційної системи виконана з врахуванням невисоких вимог до апаратних засобів, для її функціонування.

Візуальний інтерфейс програмного продукту простий, розроблений з врахуванням того, що користувач швидко те легко оволодів функціоналом системи. Розроблено ключові алгоритми роботи сценаріїв системи, ключовим з яких є алгоритм побудови динамічної таблиці журналу в pdf-файл.

Для розробки програмного продукту обліку та контролю успішності школярів обрано одну з найбільш поширеніших мов програмування – Java. В Розробку проекту виконано в середовищі IntelliJ IDEA.

ABSTRACT

AUTOMATED SYSTEM, EDUCATIONAL PROCESS, LEARNING ACCOUNTING, DOCUMENTATION AUTOMATION, PROGRAMMING, PYTHON.

Explanatory note: 69 pp., 34 pictures, 9 sources.

In the qualification work, the software product of the automated information and reference system for accounting and monitoring the performance of schoolchildren was developed and implemented.

An analytical review of the existing systems for monitoring the success of education was carried out. The disadvantages and advantages of known developments and technologies presented to the general public are considered. The task of developing an automated system, which provides for its universality, flexibility, and compliance with requirements, is defined, special attention is paid to ease of use and support.

The implementation of the information system is made taking into account the low requirements for hardware for its operation.

The visual interface of the software product is simple, designed taking into account the fact that the user quickly and easily mastered the functionality of the system. The key algorithms for the operation of system scripts have been developed, the key of which is the algorithm for building a dynamic log table in a pdf file.

One of the most common programming languages - Java - was chosen for the development of a software product for accounting and monitoring the success of schoolchildren. The project was developed in the IntelliJ IDEA environment.

ЗМІСТ

ВСТУП

Освітній процес-це інтелектуальна і творча діяльність у сфері освіти, здійснювана за допомогою системи наукових, методичних та педагогічних заходів, спрямованих на засвоєння, передачу, накопичення і використання знань, умінь та інших здібностей між учнями, а також формування гармонійно розвиненої особистості [1]. Відсутність єдиного сховища і паперових технологій призвело до виникнення багатьох недоліків в роботі шкільного закладу на поточному етапі існування, з врахуванням багатьох факторів сьогодення. Сьогодні більшість шкіл ведуть електронні журнали та ведуть облік відвідуваності занять та успішності учнів для використання цієї інформації в майбутньому. Оскільки процес обліку відвідуваності занять та академічної успішності учнів є тривалим та трудомістким, завдання обліку відвідуваності занять у школі та академічної успішності учнів часто є складним.

Освітні установи можуть бути як державними, так і приватними. Основним завданням підприємства є задоволення потреб ринку в своїх товарах і послугах для отримання прибутку. У свою чергу, навчальний заклад є юридичною особою, яка надає освітні послуги та задовольняє потреби ринку. Отже, в цій роботі школа грає роль підприємства.

В рамках кваліфікаційної роботи були проведені характеристики об'єкта автоматизації-школи-ліцею, проведено аналіз існуючої системи для вирішення набору завдань, сформульовано вимоги до розроблюваного набору завдань, наведено опис постановки набору завдань, виконані завдання з обліку відвідуваності занять і успішності учнів, розроблені елементи інформатики, математики та програмного забезпечення.

Метою даної роботи є підвищення ефективності та результативності роботи викладачів, відділу навчально-методичної роботи (НВР) та заступника

директора з виховної роботи в частині скорочення витрат часу на формування основних документів з урахуванням відвідуваності занять та успішності учнів.

Метою розробки є врахування відвідуваності занять та успішності учнів.

Об'єктом розробки є програмне забезпечення для обліку виконаної роботи у відділі підприємства.

Метод дослідження-системний аналіз з використанням об'єктно-орієнтованого підходу (object).

Для досягнення поставленої мети необхідно було вирішити наступні завдання:

- провести аналіз існуючих аналогів систем обліку;
- узагальнити вимоги до задач обліку відвідувань учнями занять і їх успішності;
- узагальнити функціональну структуру обліку та критерії обліку;
- Розробити елементи інформаційної підтримки;
- - Розробити елементи математичної підтримки;
- - Розробити програмні елементи;

Звіт про кваліфікаційну роботу підготовлено відповідно до [2-3].

1 АНАЛІЗ СУЧАСНИХ МЕТОДІВ АВТОМАТИЗАЦІЇ ОБЛІКУ ТА КОНТРОЛЮ НАВЧАЛЬНОГО ПРОЦЕСУ

1.1 Організація обліку освітніх процесів з використанням інформаційних технологій

У будь-якому навчальному закладі великий документообіг. Вчителі щодня витрачають час на заповнення журналів.

Сучасні високо комп'ютеризовані інформаційні технології замінюють фізичні ресурси і збільшують кількість електронних документів для отримання офіційного статусу. Таким чином, комп'ютеризовані інформаційні технології оцифровують цей процес у навчальних закладах, що дозволяє впровадити систему, яка додатково контролює успішність та участь учнів у навчанні. Одним з основних документів, що ведеться в навчальному закладі, є класний журнал.

Класний журнал є основним державним документом для реєстрації відвідуваності та успішності учнів у всіх навчальних закладах. Важливо, щоб кожен вчитель / викладач вів окремий журнал для кожного класу чи групи. Журнал класів є аналогом віртуального простору і може бути легко матеріалізований фізично. Електронний журнал у класі-це комп'ютеризований інформаційний інструмент, який оцифровує процес ведення традиційного журналу в класі.

Електронні журнали для занять в ідеалі служать повноцінною комп'ютеризованою інформаційною системою, а комп'ютеризовані інформаційні системи являють собою набір рішень для обробки і зберігання інформації для задоволення інформаційних потреб користувачів, тому досить ввести всю інформацію, що відноситься до навчального процесу, один раз, і це буде можливо в майбутньому. Досить заповнити або виправити ці дані в міру необхідності і використовувати їх в міру необхідності.

1.2 Огляд новітніх програмних засобів для вирішення проблеми автоматизації ведення електронних журналів

На комерційному рівні існує безліч варіантів реалізації подібних систем. У країнах Заходу оцифровка освітнього процесу більш розвинена, ніж в українських навчальних закладах. Це пов'язано з тим, що це лише початок розвитку інформаційних технологій у нашій країні, включаючи нагальну потребу в дистанційному навчанні. Це сприяє оцифровці освітньої діяльності, впровадженню комп'ютерних інтерфейсів для доступу до інформаційних систем.

В Україні ступінь цифровізації освітньої діяльності тільки починає набирати обертів, тому кількість таких послуг не так важлива, як за кордоном. У той же час багато установ, що впроваджують послуги дистанційного навчання та автоматизації навчання, є філіями іноземних агентств і середніх навчальних закладів.

Беручи до уваги проблеми та нагальні потреби, що стоять перед системою освіти України в контексті боротьби з пандемією COVID-19 та воєнного часу, а також для покращення управління системою освіти, автоматизований документообіг освітніх закладів Міністерством освіти і науки України повинен відповідати завданням системи освіти України, тому було створено ДНУ "Інститут аналізу освіти" (Наказ Міністерства освіти №781 від 09.06.2020 р.) Вести розробку державного безкоштовного електронного журналу та щоденника Electronic Journal [1].

Аналогічні додатки в Україні успішно розвивається програмне забезпечення, що розроблене приватними компаніями. Наприклад, система навчання Moodle дуже популярна як за кордоном, так і в нашій країні [2].

Це спосіб інтеграції всіх учасників освітнього процесу в спеціальну систему, яка створює своєрідне освітнє середовище, в якому реалізуються власні електронні журнали занять.

Інші успішні комерційні рішення української КС "Класний журнал" включають:

1) "Електронний щоденник і журнал" platforme-schools.info [3] - Сервіс ТОВ "Київське енергетичне агентство" за підтримки МОН України надає освітнім установам України можливість почати користуватися їх розробками електронними послугами. щоденники і журнали (рис. 1.1);

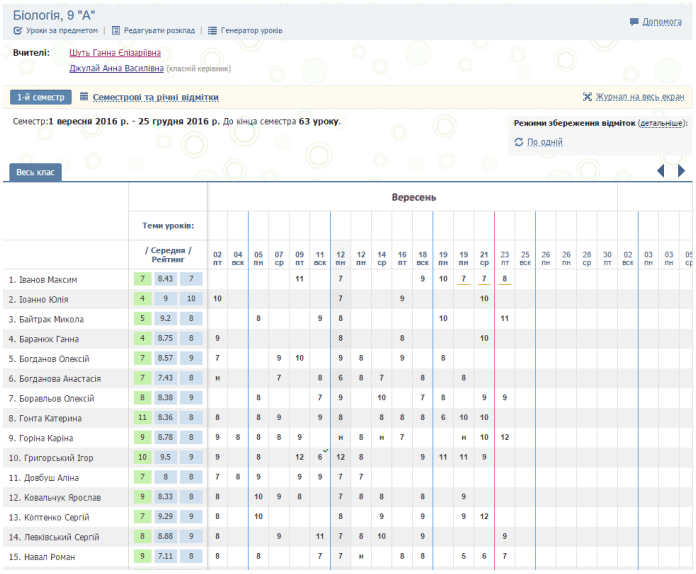

Рисунок 1.1 – Форма візуального інтерфейсу журналу для школярів системи E-schools.info

2) Автоматизована система "Щоденник" від ТОВ "Щоденник" [4] – Безкоштовна освітня мережа "Щоденник.ua" – унікальний електронний простір, система надає безліч функцій для учнів, вчителів та батьків (рис. 1.2);

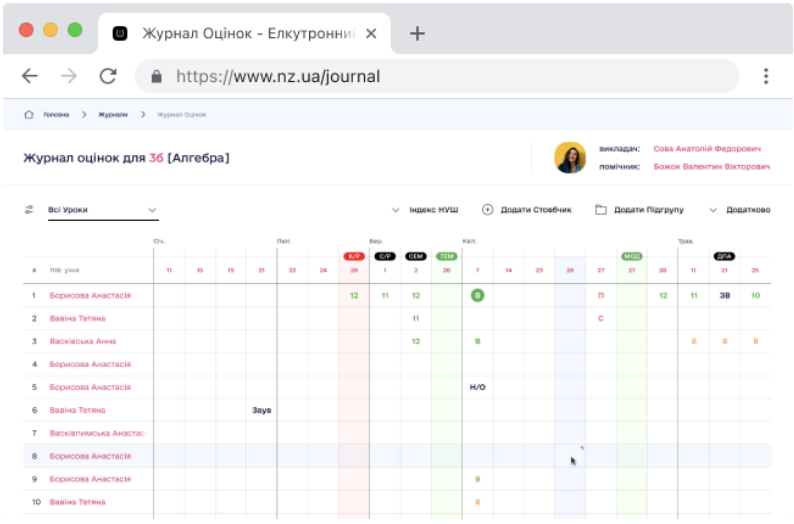

Рисунок 1.2 – Зразок сторінки електронного журналу системи «Щоденник.юа»

3) Використовуючи можливості web-порталу ukrschools.com.ua, у вас буде комерційний проект (річна підписка), створений для всіх учасників освітнього процесу.

Візуальний інтерфейс системи максимально схожий на традиційний щоденник або класний журнал. Всі розділи "Електронного журналу" зручні і інтуїтивно зрозумілі (рис. 1.3).

Рисунок 1.3 – Візуальні компоненти електронного журналу web-порталу ukrschools.com.ua

Українські вищі навчальні заклади зазвичай не впроваджують комерційні програмні продукти, а впроваджують свої власні або виготовлені на замовлення продукти інформація про його опис або структуру як правило не опубліковується в відкритому доступі.

Ось кілька університетів, які застосовують подібні рішення:

- Національний педагогічний університет імені М.П. Драгоманова [8].

- Київський державний економічний університет[5];

- Національна академія Державної прикордонної служби України [7];

- Буковинський державний медичний університет [6];

- Криворізький національний університет.

2 ПРОЕКТУВАННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ ОБЛІКУ ТА КОНТРОЛЮ НАВЧАЛЬНОГО ПРОЦЕСУ

2.1 Аналіз домену та призначення інформаційних об'єктів

Одним з основних інформаційних об'єктів є класний журнал (академічний журнал) в якості основного документа для обліку відвідуваності і успішності у школі. Таким чином, тематична область цієї сфери діяльності в рамках освітнього процесу повинна характеризуватися особливостями інформаційного потоку з його власними об'єктами, які складають основу такої інформаційної системи, як класний журнал.

Облікові журнали по роботі академічних груп і викладачів (далі - журнали) - це журнали, в яких описується успішність учня (школяра, студента), відвідуваність занять, виконання освітніх програм і т.д. та збереження відповідної інформації.

Наявність журналів у класі є обов'язковим. У позакласні години журнал зберігається в академічному відділі (очному відділенні).

Основою визначення предметної області є створення системи ведення обліку навчальних занять, зокрема, відповідно до витягів з правил ведення обліку навчальних занять державного освітнього закладу:

1) Журнал ведуть вчителів, викладачі університетів і керівники груп. Вчитель несе відповідальність за точність і достовірність записів;

2) Ведення журналу ведеться виключно державною мовою з предметів іноземної мови, з інформатики та обчислювальної техніки можливі часткові записи англійською;

3) Записи ретельно і акуратно зроблені синім чорнилом. Виправлення сторінок журналу не може бути виконано, і в разі виникнення помилки

поруч з ним створюється змінена версія, завірена підписом і скріплена печаткою відповідного представника адміністрації закладу;

4) куратори групи заповнюють розділ "Реєстрація відвідувань";

5) відсутність учня в класі позначається літерою "н", це робиться в кінці уроку, якщо учень запізнюється, ставиться буква "з", якщо учень отримав оцінку, вона поміщається в відповідну клітинку;

6) Основними різновидами оцінки освітніх досягнень учня в галузі освіти є поточна і підсумкова (тематична, модульна, семестрова, річна), державні підсумкові атестації, заліки, іспити. Тематичні оцінки проводяться не рідше 2 разів на семестр відповідно до навчальної програми;

7) незадовільні семестрові оцінки , кредити, модулі та іспити можуть бути піддані коригуванню. Виправлені оцінки розміщуються без дати в стовпцях з виправленими мітками поруч зі стовпцями I, II, заліками та іспитами. Задовільні оцінки можуть бути встановлені лише в особливих випадках з дозволу директора університету.

Основою навчального журналу є його зміст, іншими словами, робоче навантаження групи за певний період навчального року (див. рис. 2.1).

ЗМІСТ

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

Рисунок 2.1 – сторінка змісту журналу навчання

Відповідно до сторінки змісту формуються сторінки відповідних дисциплін які поділяються на «Ліву» частину (див. рис. 2.2) та «Праву» частину (див. рис. 2.3).

В лівій частині журналу заноситься інформація про успішність і відвідуваність учня в залежності від організованого освітнього процесу, основними характеристиками якого є дата, тип і предмет курсу (див. рис. 2.2).

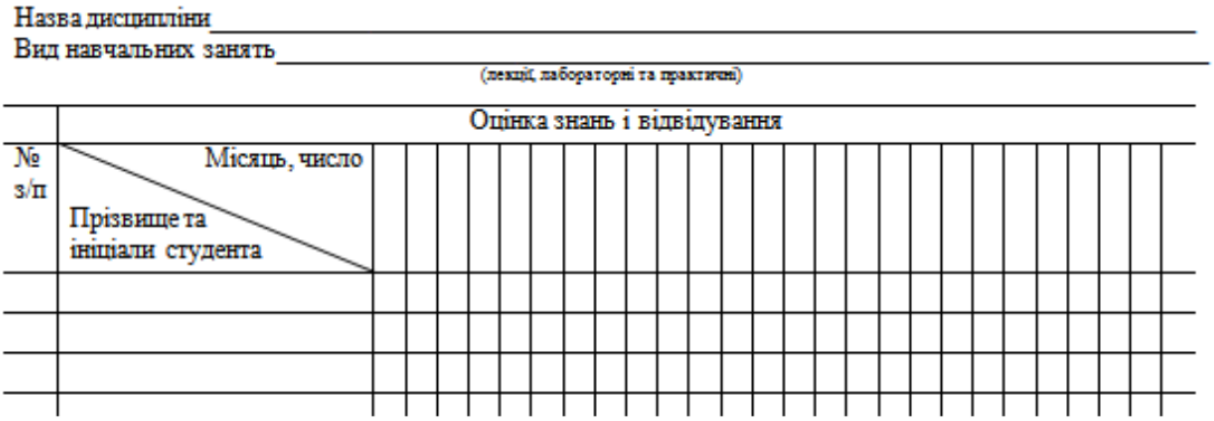

Рисунок 2.2 – Форма «Лівої» частини журналу

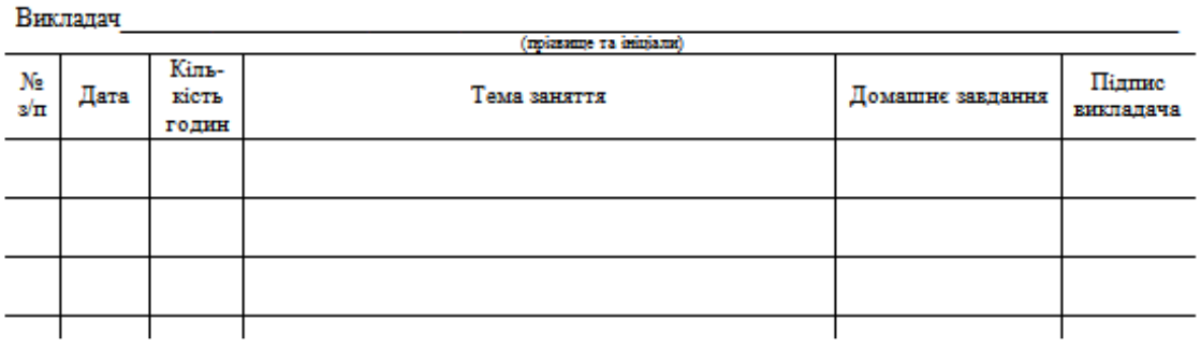

Рисунок 2.3 – Форма «Правої» частини журналу

На основі узагальнення цих компонентів навчального журналу можна скласти список необхідних інформаційних об'єктів предметної області і список функцій, безпосередньо пов'язаних з цими об'єктами, які необхідні для відтворення цього журналу в електронному вигляді (Таблиця 2.1).

Тому рекомендується розробити інтерфейс для управління навчальним навантаженням, вчителями, кураторами та студентами. Крім того, планується додати можливість роздруковувати заповнені сторінки журналу за формою так, що зшивши їх отримуємо готовий журнал встановленої форми.

Тому, найбільша увага приділяється формуванню відповідних форм журналу та їх виводу на друк.

Таблиця 2.1 - Інформаційні об'єкти і функціональні блоки в моделі класного журналу

Список інформаційних об’єктів	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)
Перелік функцій, де будуть використані інформаційні об’єкти	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.)

2.2. Побудова логічної моделі та проектування реляційної бази даних

На основі даних, отриманих та проаналізованих у попередньому підрозділі, обраних інформаційних об'єктів та функціональних можливостей

та блоків моделі спроектовано логічну модель автоматизованої системи (див. рис. 2.4).

Спираючись на інформацію про функціональні блоки розробленої , логічної моделі інформаційної системи, можна переходити наступного етапу –проектуванню бази даних.

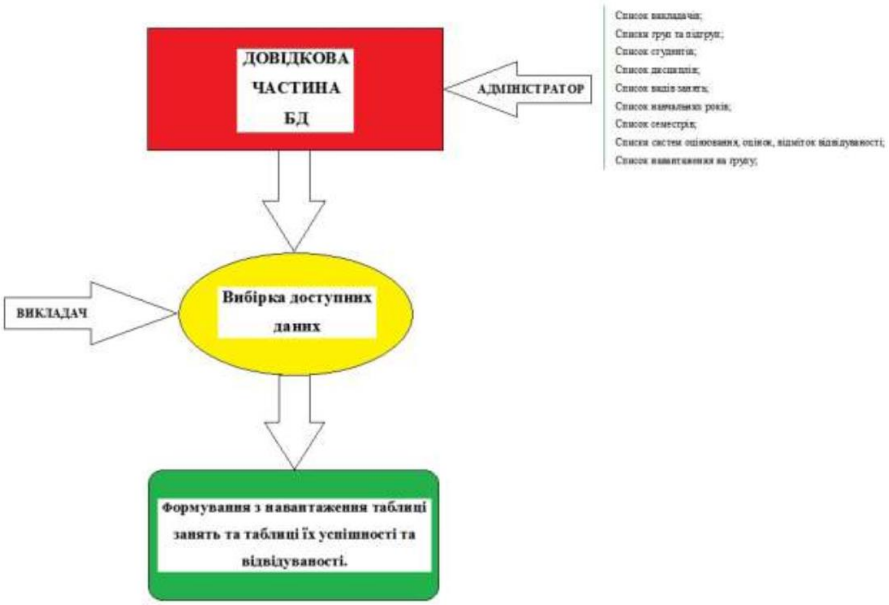

Рисунок 2.4 – Логічна схема інформаційних потоків автоматизованої системи

Глибокий аналіз структури інформаційних потоків, та типів даних, що використовуються в журналі дозволив розробити детальну структуру надійної бази даних інформаційної системи (див. рис. 2.5).

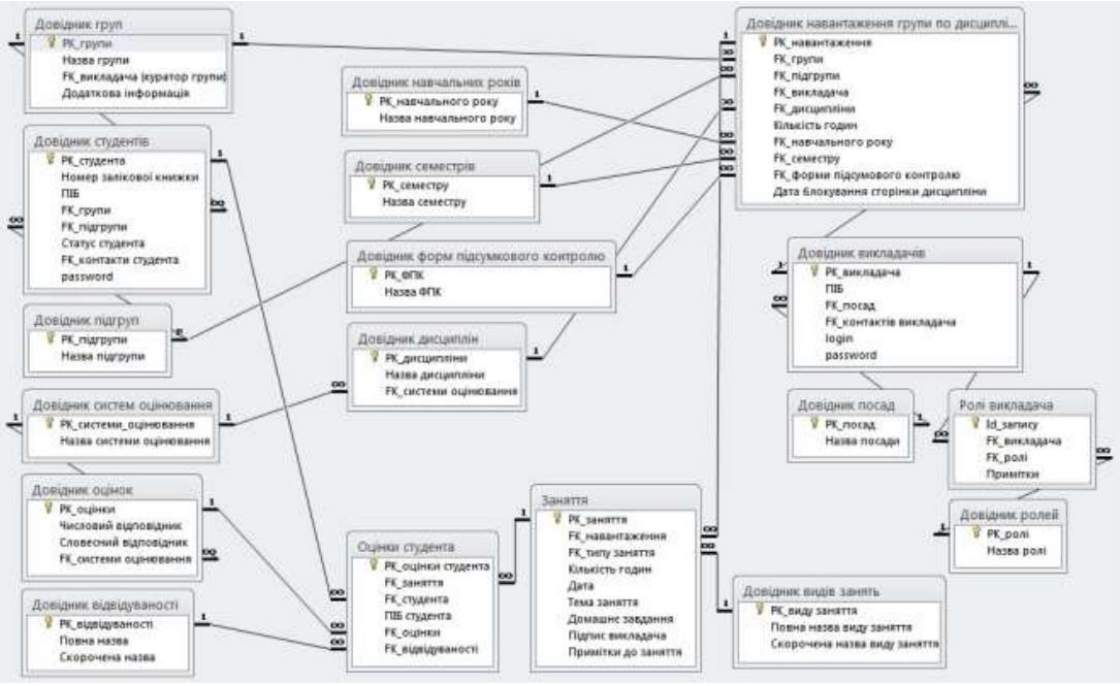

Рисунок 2.5 – Схема логічної структури бази даних

На побудову структури бази даних вплинули особливості організації та ведення навчального журналу:

1) учні - це учні тільки в одній групі. Це пов'язано з необхідністю чіткої ідентифікації учня за номером в реєстрі;

2) вчителі можуть одночасно виконувати кілька ролей у системі, і залежно від ролі буде доступний відповідний інтерфейс;

4) учні можуть бути згруповані в підгрупи, також різні вчителі можуть викладати одну і ту ж дисципліну в різних підгрупах. У той же час, якщо вчителя тимчасово замінить інший вчитель, він не матиме доступу до сторінки групи, але це може бути досягнуто з дозволу адміністратора. Примітки до виконання уроку іншим учителем вказуються в полі "підпис", де вказується прізвище вчителя, який вніс зміни;

5) дата заборони на внесення інформації, що відноситься до дисципліни, повинна бути вказана окремо для кожної сторінки журналу;

6) для кожної "сторінки" журналу слід вибирати вчителя відповідно до робочого навантаження;

7) робоче навантаження групи по підгрупах використовуються для перегляду навчальної програми по заданому робочому навантаженні і для перевірки кількості дисциплінарних годин в даній підгрупі групи при фактичному розрахунку. Якщо це число збігається, з'явиться повідомлення про віднімання часу;

8) якщо предмет викладається протягом 2х або більше семестрів, в базі вона враховується двома різними дисциплінами, а саме "математика" зі специфікацією "I семестр" і "Математика" зі специфікацією "II семестр" (для цього є довідкова таблиця. "Довідник по семестрах"). Подібним чином, окремими предметами є "Біологія" та "Біологія людини" тощо;

9) Заняття можуть бути "з обмеженнями" або "без обмежень" в тарифікації часу. відповідно, за замовчуванням кількість годин для кожного уроку дорівнює 0,75 , і при необхідності буде введено інше число (наприклад кількість годин екзамену впливає кількість учнів у класі);

10) Будь-яка рейтингова система включає в свій набір два жовтневих додаткових універсальних рейтингу "зарах".", "незарах.", оскільки конкретний тип виконуваної роботи не повинен бути "оцінений" у типовому сенсі;

11) База даних містить досить багато довідкових таблиць, тому довелося вирішити проблему видалення непотрібних (незв'язаних записів). Такі рішення, як для корпоративних баз даних, можуть допомогти вам вирішити цю проблему. Тобто додавання додаткового поля "сутність" до таблиці допомагає вирішити цю проблему. "Доступність", "релевантність") характеризується властивістю "1" або "0", Що означає, що об'єкт доступний (релевантний) або видалений (не має значення).

2.3 Вибір системи управління базами даних та середовища розробки програмного забезпечення

Оскільки MySQL одна з найпопулярніших на сьогодні систем управління базами даних, очевидно, що MySQL Server 1.0 був обраний як

сервер баз даних. Він використовується для створення веб-сайтів і додатків, комунікацію з ним підтримують різні мови програмування.

Тому при виборі СУБД, вона буде створена та налаштована за допомогою застосунку MySQL Workbench, але оскільки це продукт корпорації Oracle як і MySQL Server, тим досягається повна сумісність двох продуктів.

MySQL Workbench (рис. 2.6) - це уніфіковане програмне забезпечення для архітекторів, розробників та адміністраторів баз даних.

Рисунок 2.6 – Схема логічної структури бази даних

MySQL Workbench надає набір інструментів для моделювання даних, розробки SQL, управління та налаштування серверів та їх користувачів, а також резервного копіювання. MySQL Workbench можна використовувати у Windows, Mac OS X та Linux.

Серед основних функції та переваг MySQL Workbench:

- дозволяє адміністраторам баз даних, розробникам або архітекторам даних візуально проектувати, моделювати, генерувати та керувати

базами даних. Він містить усе необхідне для створення, прямого та зворотного проектування складних моделей діаграм EER та надає критичні можливості для виконання складних завдань управління змінами та документування, які зазвичай вимагають багато часу та зусиль;

- він має зручні та корисні візуальні інструменти для створення, запуску та оптимізації запитів SQL. Редактор SQL виконує підсвічування синтаксису, завершення виразів, повторне використання фрагментів SQL та історію виконання SQL. Ви можете використовувати панель підключень до бази даних для ефективного управління стандартними підключеннями до бази даних.

- браузер інформаційних об'єктів забезпечує миттєвий доступ до схем бази даних та інших об'єктів;

- включає візуальну консоль для легкого управління середовищем MySQL та покращення представлення бази даних. Розробники та адміністратори баз даних все частіше використовують візуальні інструменти для налаштувань та управління сервером та користувачами, а також для створення резервних копій та контролю за станом бази даних;

- надає набір інструментів для підвищення продуктивності додатків Mysql. Адміністраторами баз даних може використовуватися панель керування продуктивністю для швидкого перегляду ключових показників продуктивності. Звіти про ефективність полегшують визначення та доступ до точок доступу вводу-виводу, значень інструкцій SQL тощо. Крім того, одним натисканням миші розробники можуть скористатися вдосконаленим та простим у використанні інструментом планування візуального опису, щоб визначити, яким чином можна оптимізувати свої запити;

- Надає повне і просте у використанні рішення для перенесення Microsoft SQL Server, Microsoft Access, Sybase ASE, PostreSQL та інших таблиць,

об'єктів і даних в СУБД MySQL. Розробники баз даних і адміністратори можуть швидко і легко перетворити існуючі програми для роботи з MySQL як на Windows, так і на інших платформах. Міграція також підтримує оновлення застарілої версії MySQL до останньої версії.

В якості середовища розробки програмного забезпечення був обраний продукт IntelliJ IDEA компанії JetBrains (рис. 2.7), який в даний час вважається найкращим середовищем розробки на Java.

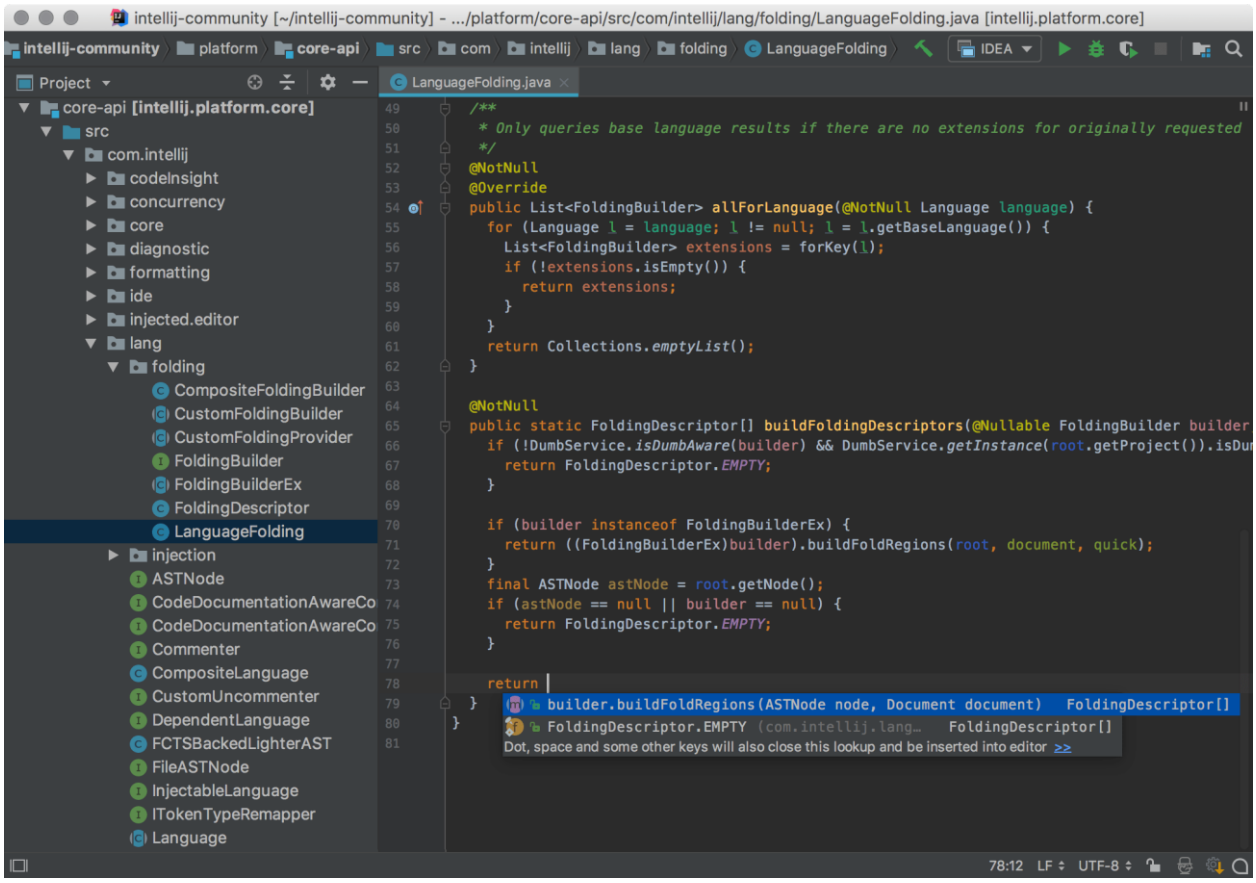

Рисунок 2.7 – Головне вікно IntelliJ IDEA з відкритим проектом

IntelliJ IDEA середовище розробки, що користуються повагою розробників з солідним стажем. платформа орієнтована на розробку на Java, але її універсальність дозволяє їй з успіхом працювати на безлічі інших мов.

Звичайно, немає такої безлічі плагінів, як у конкурентів, що мають відкритий код, але можна перевірити їх якість. Якщо порівняти це середовище

з операційною системою, IntelliJ буде такою ж, як Mac OS – закритою, трохи консервативною, але в той же час надійною.

Деякі функції та переваги IntelliJ IDEA:

- Детальний аналіз коду - аналізує код, вибудовуючи зв'язки між файлами проекту та символами на всіх мовах, що використовуються у ньому. Це надає допомогу у розробці коду, досить зручну навігацію по проекту, перевірку коду на помилки та, звичайно, рефакторинг;
- Ергономічне середовище-кожен компонент IntelliJ IDEA розроблений для зручності та ефективності використання. Основна ідея цього середовища розробки полягає в тому, що ніщо не відволікає розробників від творчого процесу;
- Великий набір вбудованих інструментівм "з коробки " для розробників, що дозволяють максимально спростити їх робочий процес, включаючи декомпілятори, засоби перегляду байт-коду і FTP;

3 РЕАЛІЗАЦІЯ ПРОЕКТУ

3.1 Створення бази даних

Під час створення бази даних необхідно враховувати три основні компоненти моделі даних реляційної бази даних:

- структура даних;
- операції, які будуть виконуватися над даними;
- обмеження, від яких залежить цілісність даних.

Основними структурами даних реляційної моделі є таблиці, які в реляційній теорії називаються "відношеннями". Власне, термін "реляційний" і є походженням назви моделі relational.

Логічна модель бази даних відображає інформаційно-логічну модель цільової області.

Інформаційні об'єкти моделі даних представлені відповідними реляційними таблицями. Структура таблиці визначається реквізитами відповідного інформаційного об'єкта, при цьому кожне поле відповідає одному з реквізитів об'єкта.

Основні реквізити об'єкта утворюють унікальний ключ реляційної таблиці. Для всіх полів таблиці вказується тип даних, розмір та інші властивості. Записи таблиці відповідають компонентам об'єкта і створюються під час ініціалізації таблиці.

Таблиці бази даних були створені на основі логічної моделі, попередньо створеної в середовищі MySQL Workbench (рис. 3.1).

Усі таблиці та відносини були створені відповідно до логічної моделі бази даних. Загалом схема бази даних, включно з усіма відношеннями, полями, розмірами полів і типами даних, має такий вигляд (рис. 3.2).

Рисунок 3.1 – Список таблиць створеної бази даних

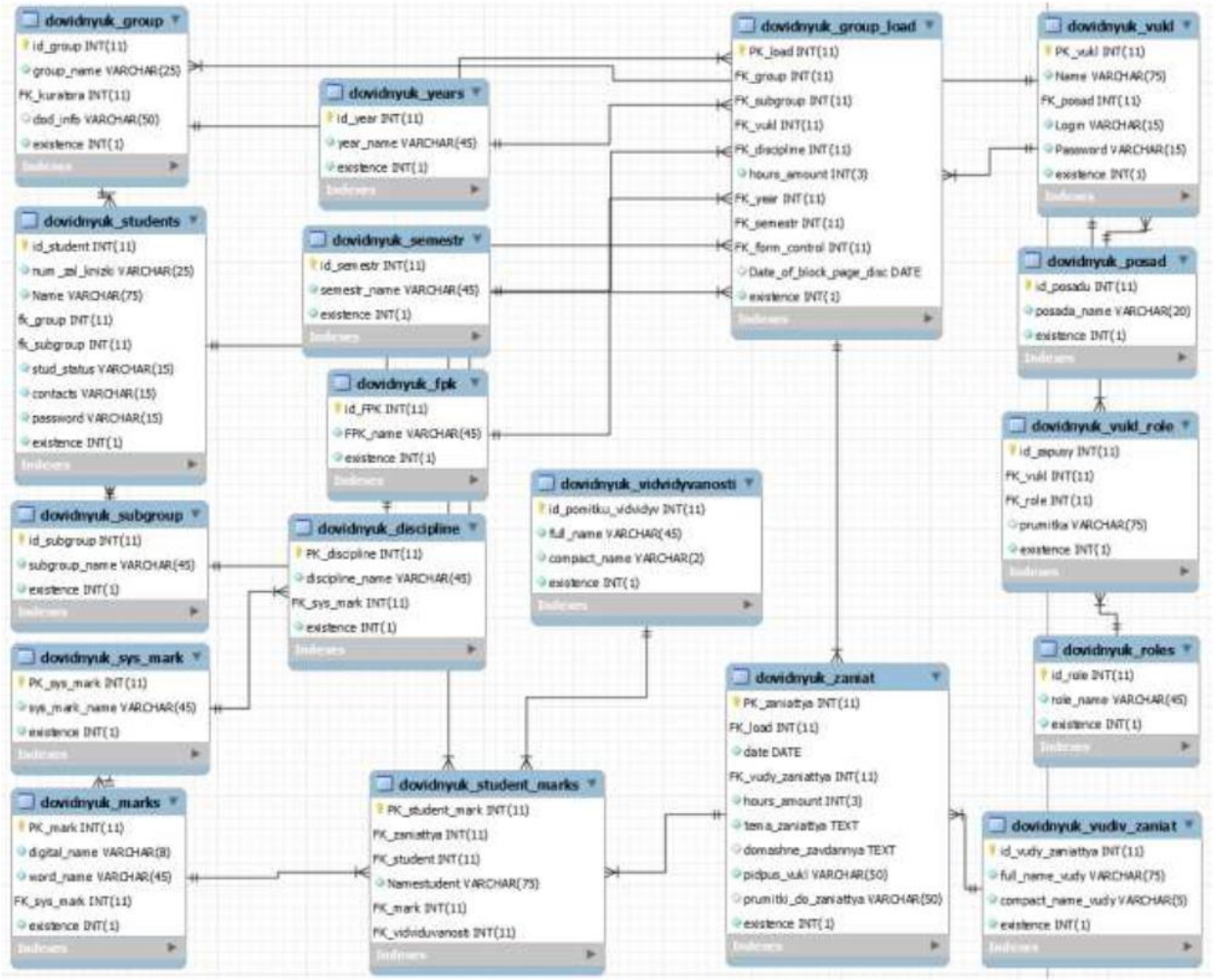

Рисунок 3.2 – EER-діаграмма створеної бази даних

Цей процес складається з шести етапів:

1) Клієнт запитує дані;

2) Запит перетворюється на SQL-запит;

3) SQL-запит надходить через мережу до серверу.

4) Програмне забезпечення серверу обробляє запит та ініціює процес пошуку;

5) необхідні записи повертаються клієнту;

6) дані представляються користувачеві;

Використання технології клієнт-сервер має такі переваги:

- відносно низька вартість системи;
- достатня обчислювальна потужність
- легке налаштування під конкретні завдання
- низькі вимоги до обсягу оперативної пам'яті на клієнті;
- низькі вимоги до дискового простору на стороні клієнта;
- сервери можуть зберігати великі обсяги даних; низька вартість; низькі вимоги до дискового простору на стороні клієнта; низька вартість резервного копіювання даних; низька вартість резервного копіювання даних;
- спрощуються процеси резервного копіювання даних;
- більш просте загальне адміністрування системи та управління безпекою.

Основним інструментом, використовуваним для розроблення програми, є мова програмування Java.

Java - мова програмування та обчислювальна платформа, що розроблена компанією Sun Microsystems у 1995. Нові інноваційні продукти та цифрові послуги, розроблені для майбутнього, також продовжують спиратися на Java. Хоча більшість сучасних Java-додатків базуються на середовищі виконання Java і додатках разом, усе ще існує безліч додатків і веб-сайтів, які не працюватимуть, якщо Java не встановлено на настільному комп'ютері.

Для початку розробки необхідно встановити середовище виконання Java (JRE), яке містить у собі віртуальну Java машину (JVM), класи платформи Java

та бібліотеки Java;JRE - це складова програмного забезпечення Java, яка запускається під час виконання програми на Java. також необхідно Java Development Kit (JDK), який є інструментом розробки.

Основою для створення Java-додатків є платформа JavaFX. JavaFX - це платформа на базі Java, призначена для створення додатків із розвиненим графічним інтерфейсом. Її можна використовувати для створення настільних додатків (що запускаються з операційної системи), розроблення веб-додатків, таких як RIA, що запускаються в браузері, і мобільних додатків. JavaFX є наступником застарілої графічної бібліотеки Swing. Конкурентами цієї платформи є Adobe Flash, Microsoft Silverlight та інші подібні системи.
Безпосередньо під час розроблення додатка використовувалися такі бібліотеки (драйвери):

1) MySQL Connector Java (драйвер або зовнішня бібліотека) - бібліотека, що забезпечує під'єднання додатків до баз даних MySQL і надає спеціальні інструменти для налаштування створених з'єднань. З'єднання - забезпечує підключення (сеанс) до певної бази даних. SQL-запити виконуються і результати повертаються в контексті з'єднання; база даних в об'єкті Connection може надавати інформацію, що описує її таблиці, підтримуваний синтаксис SQL, збережені процедури, особливості даного з'єднання і т.д.
2) DriverManager - основний клас, що керує набором драйверів JDBC. У процесі ініціалізації клас DriverManager завантажує класи драйверів, на які посилається система jdbc.drivers. Це дозволяє користувачеві налаштувати драйвер JDBC, який буде використовуватися додатком.
3) ResultSet – таблиця, що представлена набором результатів бази даних, яка генерується оператором бази даних, що побудував запит ResultSet має покажчик поточного кортежу.
4) Statement - об'єкт, який використовується для виконання статичного SQL-оператора і повернення результату. За замовчуванням в об'єкті Statement одночасно може бути відкритий тільки один об'єкт

ResultSet. Тому, якщо ви хочете чергувати читання одного об'єкта ResultSet із читанням іншого, кожен із них має бути створений на окремому об'єкті Statement. Усі методи виконання інтерфейсу Statement неявно закривають поточний об'єкт Statement ResultSet, якщо він існує.

5) SQLException - виняток, що надає інформацію про помилки доступу до бази даних та інші помилки.
6) ObservableList - список, що дозволяє відстежувати зміни, коли вони відбуваються.
7) FXCollections - сервісний клас, що складається зі статичних методів, які є копіями java.util.FXCollections. Методи-обгортки (такі як synchronisedObservableList і emptyObservableList) мають таку саму функціональність, як і методи в наборі, за винятком того, що вони повертають ObservableList, тож вхідні дані Вони підходять для методів, які потребують ObservableList як вхідні дані. Сервісні методи тут в основному з міркувань продуктивності.
8) ArrayList - найпоширеніший тип списку; ArrayList - це автоматично розширюваний масив. Він може працювати з масивами, але квадратні дужки не використовуються.
9) JFoenix - Java-бібліотека з відкритим вихідним кодом, яка реалізує Material Design від Google за допомогою Java-компонентів.
10) iTextPDF - одна з найбільш зручних бібліотек для роботи з PDF-файлами, яка надає можливість генерувати PDF-функціональність програмні продукти. Ключовою перевагою є її гарна документація.

3.2 Розробка візуальних інтерфейсів користувачів автоматизованої системи

Першим кроком було створення структури ("скелету") проекту (рис. 3.4).

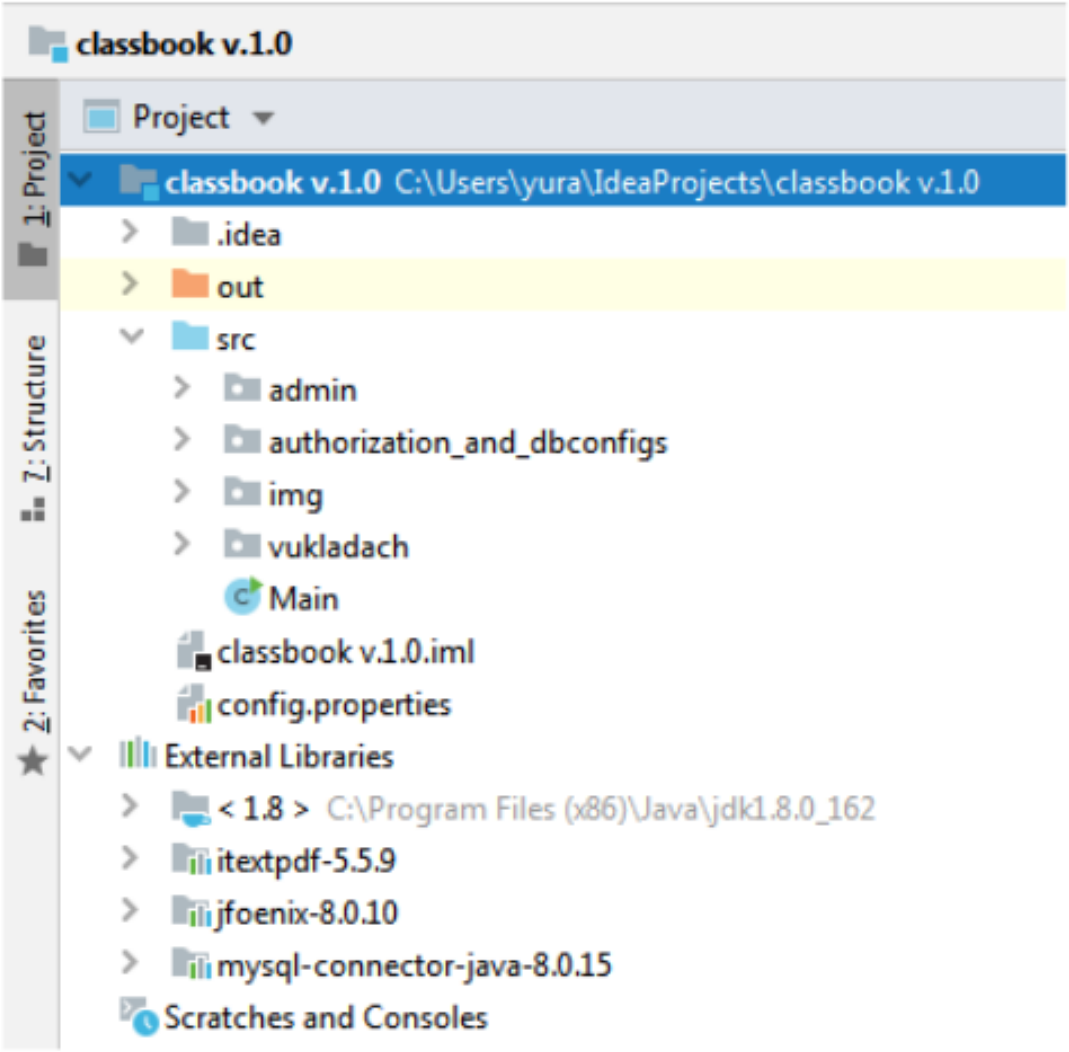

Рисунок 3.4 – Структура проекту

Далі було створено клас DBConnection (рис. 3.5), який відповідає за взаємодію з базою даних.

Важливим аспектом цієї реалізації є те, що дані для доступу до бази даних зберігаються не безпосередньо в коді, а у файлі властивостей. Файл властивостей призначений для зберігання статичних даних, необхідних проєкту, наприклад, імені та пароля для входу в базу даних.

Після створення класу DBConnection було створено форму і дизайн адміністративної панелі додатка (рис. 3.6).

```
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=tru
e&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(); }}
```

Рисунок 3.5 – Код класу DBConnection

Рисунок 3.6 – Інтерфейс адміністрування застосунку

Загалом для створення функціональності адміністративної панелі застосунку використовували два типи форм: повнорозмірні (рис. 3.7) і підформи (діалогові) (рис. 3.8).

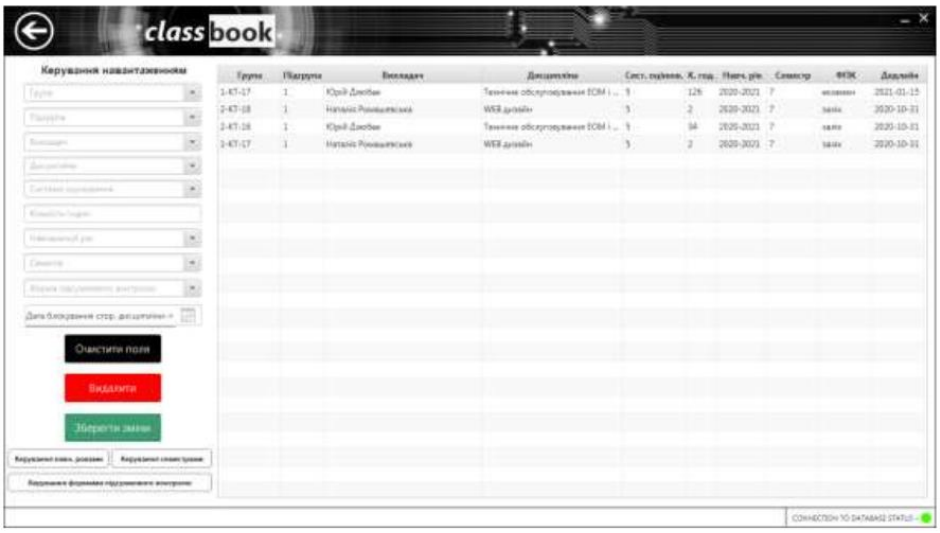

Рисунок 3.6 – Повнорозмірна панель "Управління навантаженням"

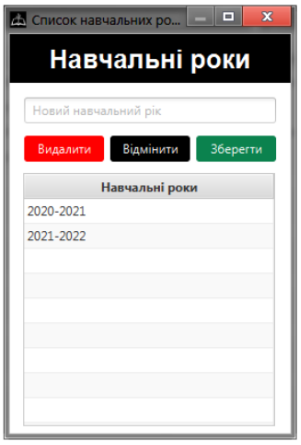

Рисунок 3.6 – Підформа "Управління оцінками"

Усі інші панелі створюються за аналогічною моделлю форми. Кожна панель пов'язана з власною таблицею або декількома пов'язаними таблицями в базі даних.

Для кожної форми в програмі створюються три окремі класи: клас FXML (у ньому описується зовнішній вигляд форми і вказується ID компонента), клас контролера форми (відповідає за методи подій, що виникають під час маніпулювання формою), клас відправника/одержувача (відповідає за змінні, через які дані зчитуються з бази даних і передаються в неї).

Для розробки загального стилю застосунку та створення фонових зображень ми використовували фоторедактор і власну фантазію. Передня частина форми була створена за допомогою редактора JavaFX - Gluon's Scene Builder. Цей редактор дає змогу легко створювати елементи керування, діаграми, фігури та контейнери інтерфейсу JavaFX, даючи змогу швидко й ефективно створювати прототипи користувацьких інтерфейсів.

Після створення адміністративної панелі було створено дизайн і функціональність головного вікна входу в систему (рис. 3.9).

Рисунок 3.9 - Головне вікно входу в систему

Крім того, було розроблено і додано функцію перевірки введених у поля ідентифікаторів, а також створено візуальні підказки. Наприклад, якщо один з ідентифікаторів (або обидва) було введено невірно, то символ введеного поля підсвічувався червоним кольором, а правильно введеного - зеленим (рис 3.10).

Під час розробки було враховано можливість перевірки статусу з'єднання із сервером.

Після завершення цього етапу створюється стартова панель особистого кабінету користувача системи. На цій панелі відображаються імена авторизованих користувачів і ролі, доступні для поточної сесії (рис. 3.11). Основна роль - "Вчитель", яка за своєю суттю схожа з роллю вчителя при заповненні традиційного паперового щоденника.

Рисунок 3.10 - Головне вікно входу в систему з невірно введеним паролем

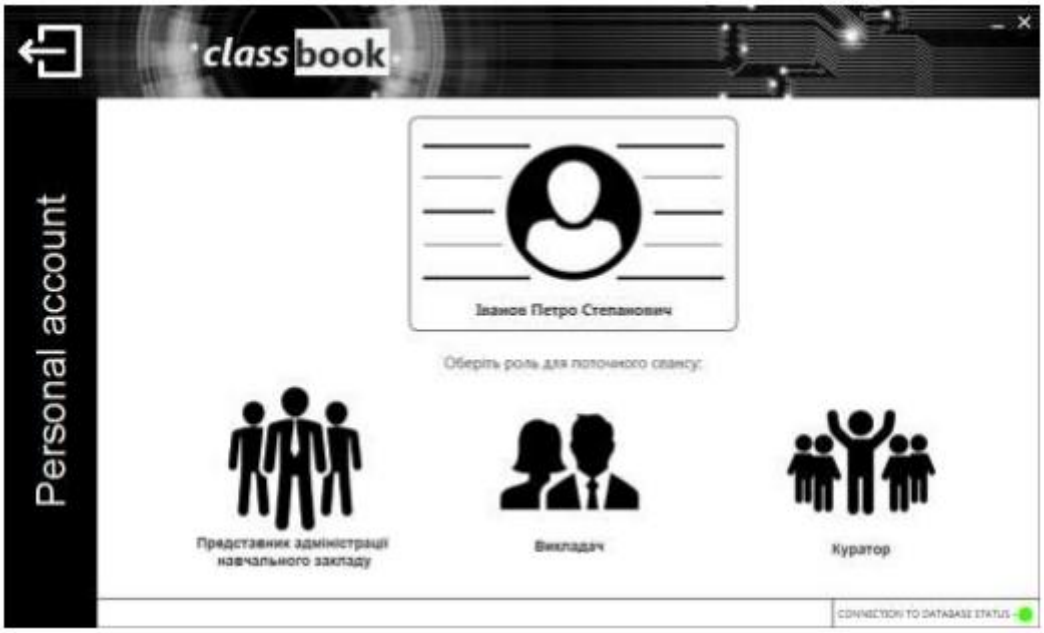

Рисунок 3.11 – Головне інтерфейсне вікно користувача

Різниця між додатковими ролями полягає в тому, що "Представник адміністрації навчального закладу" може вибрати будь-яку групу, рік, семестр або поле і переглянути їхні дані, тоді як "Куратор" може переглянути тільки всі дані за своєю групою, але обидві ці ролі можуть вводити нові дані в рамках

інформації, доступної для (за винятком службових записок), і не мають змоги змінювати наявні дані.

В подальшому було створено форму керування данними з колекції навчального навантаження (рис. 3.12).

Рисунок 3.12 – Інтерфейс вибору вчителем доступних предметів

Наступним створено форму для керування заняттями (рис. 3.13), що по суті заповнюється на правій сторінці паперового журналу

Рисунок 3.13 – Інтерфейс керування даними заняття

Потім створено форму оцінювання занять (рис. 3.14). Щоб перейти до форми оцінювання, потрібно вибрати потрібне заняття у таблиці та натиснути на синю кнопку "До оцінки/статусу відвідуваності".

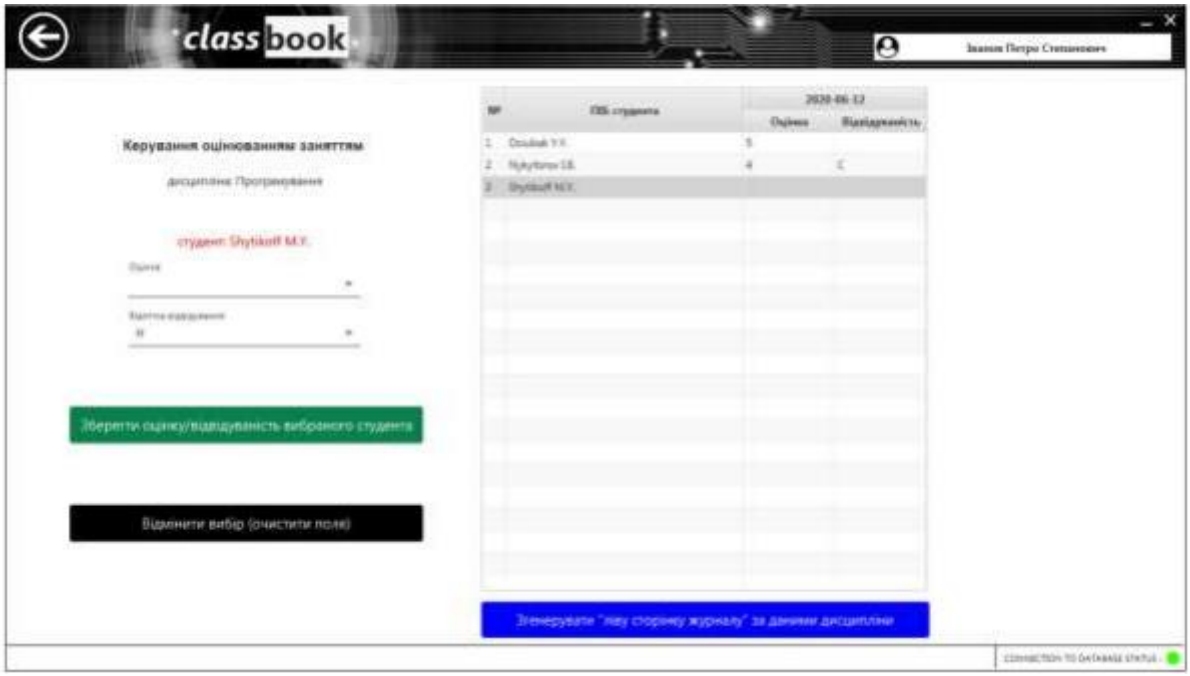

Рисунок 3.14 – Інтерфейс оцінювання заняття

Щоб зробити інформацію з усіх дисциплін наочнішою, було створено окрему форму для відображення "лівого боку" журналу (рис. 3.15).

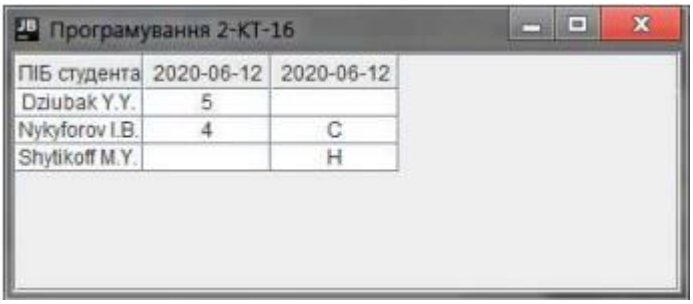

Рисунок 3.15 – Таблична (традиційна) форма представлення успішності класу

3.3Реалізація функції роздрукування сторінок журналу на основі даних, введених у базу даних

Для того щоб реалізувати функцію друку журнальних сторінок, спочатку необхідно було розробити алгоритм генерації документа, придатного для друку, на основі даних, введених у базу даних. Також необхідно було розробити алгоритм генерації шаблону, придатного для цього документа.

Для реалізації цих алгоритмів було обрано бібліотеку iTextPDF, призначену для розроблення Java-додатків, що дають змогу створювати, конвертувати й обробляти PDF-документи.

Оскільки необхідно було роздрукувати ліву і праву сторінки журналу (рис. 2.2 - 2.3), для створюваних документів потрібно було розробити два шаблони.

Функції, що відповідають за генерацію, заповнення і збереження шаблонів, були поміщені в класи, що відповідають за заповнення даних для "лівої" і "правої" сторінок журналу. Це означає, що алгоритми правильного формування цих сторінок не потрібно повторювати для створення друкованого документа, оскільки ці класи формують таблицю для відображення попередньо заповнених даних і фактично є цифровими версіями "правої" і "лівої" сторінок класного журналу.

Узагальнений алгоритм роботи застосунку, що забезпечує реалізацію зазначеного функціоналу програмного продукту наведено в додатку Б.

Цей алгоритм простіший, ніж алгоритм генерації "лівої" сторінки. Це пов'язано з тим, що, на відміну від "лівої" сторінки, "права" сторінка має фіксовану кількість колонок. Це означає, що "ліва" сторінка заповнюється по горизонталі, а "права" - по вертикалі.

Усі особливості було враховано, і було розроблено абсолютно унікальний алгоритм заповнення згенерованого шаблону "лівої" сторінки (на прикладі шаблону "правої"), який показано в методі AddToPDF() класу LeftSide в додатку Б.

У результаті створюється PDF-документ для "лівої" і "правої" сторінок журналу (рис. 3.17).

Відповідний функціонал було додано на візуальні форми взаємодії з даними (рис. 3.16)

Рисунок 3.16 – На форми додано кнопки для створення PDF-документа.

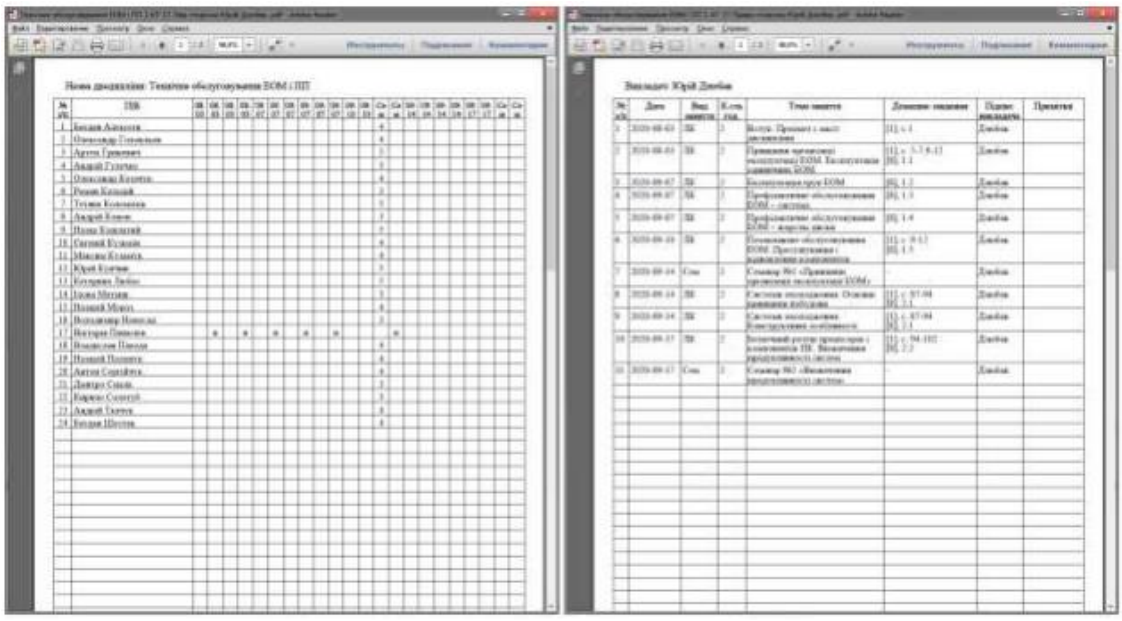

Рисунок 3.17 – PDF-документи сторінок журналу

3.4 Збірка і компіляція проекту

Додаток було зібрано та скомпільовано у виконуваний файл за допомогою інструменту інтеграції із середовищем розробки IntelliJ IDEA. Для цього в пункті меню Project Structure виберано Artifacts - Add - JavaFx Application - From module 'classbook v.1.0' (рис. 3.18).

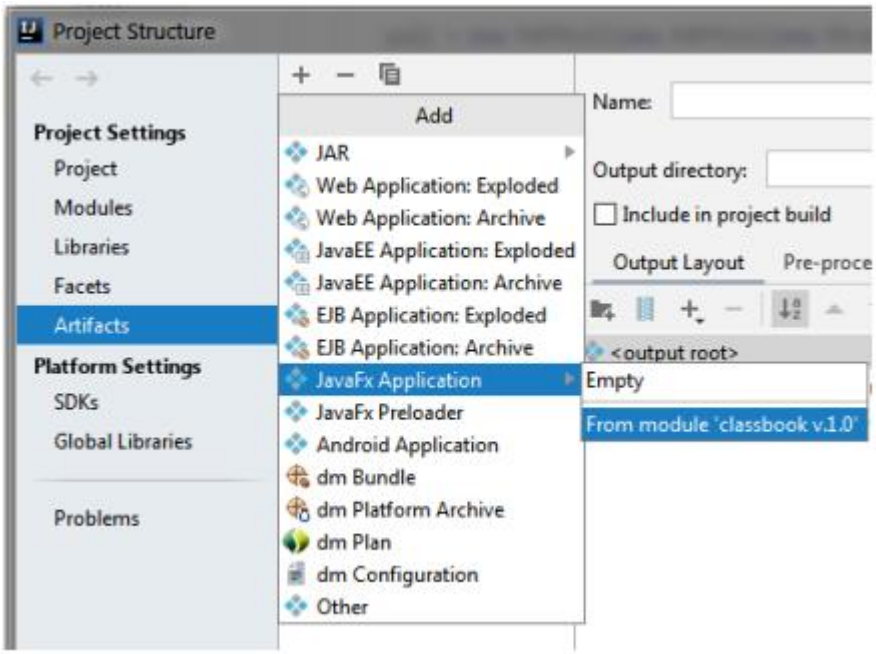

Рисунок 3.18 – Перші кроки збірки застосунку

Далі необхідно вказати вихідні файли проєкту, необхідні для збірки (рис. 3.19).

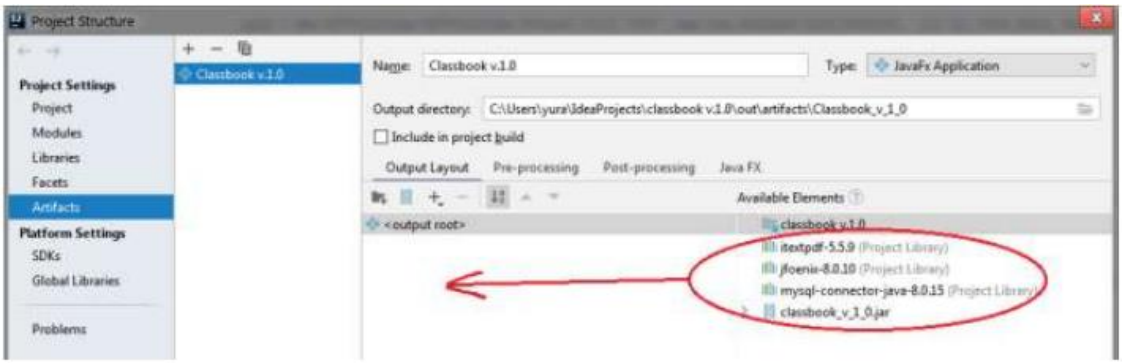

Рисунок 3.18 – Передача вихідних файлів проекту для формування зібраного додатка

Далі необхідно задати необхідні параметри і вказати файл іконки додатка в меню параметрів JavaFx (рис. 3.19).

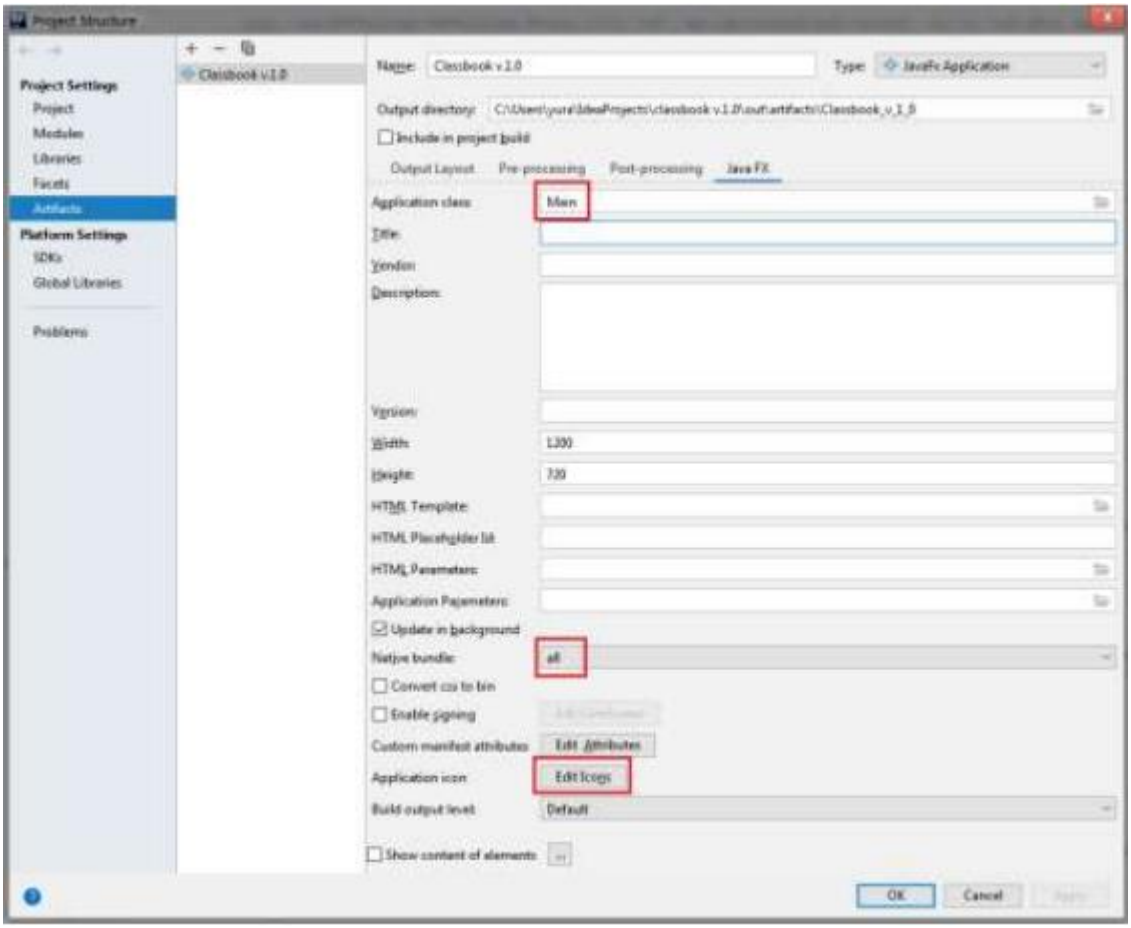

Рисунок 3.19 - Налаштування в меню параметрів JavaFx

Далі відкрито Build - Build Artifacts і виконано Build.

Якщо артефакт збірки буде виконано без помилок, то зібрана програма, включно з виконуваним файлом, з'явиться в каталозі out-artifacts папки проєкту (рисунок 3.20).

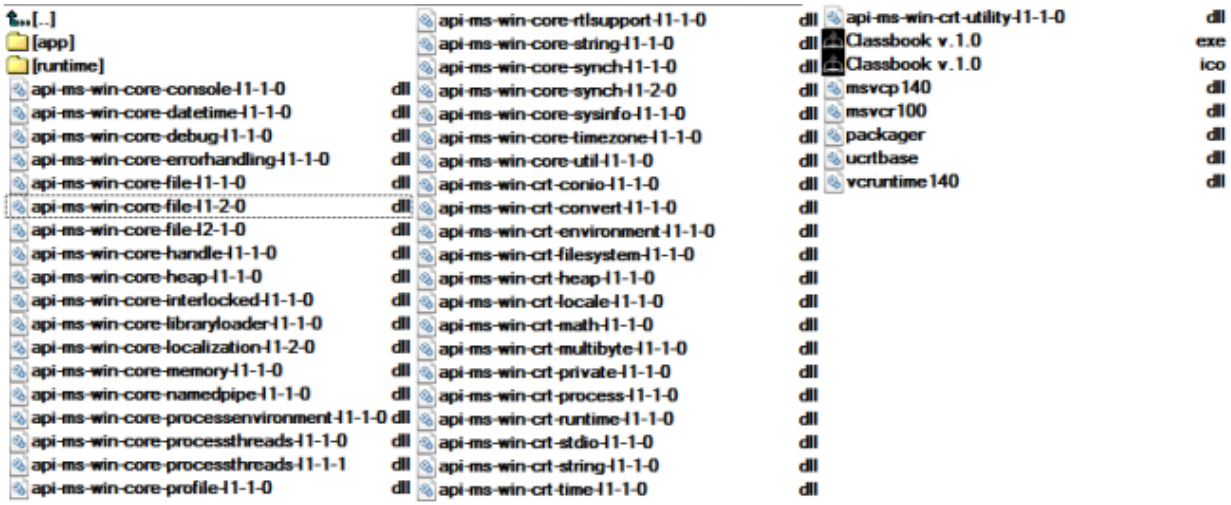

Рисунок 3.20 – Каталог зібраних файлів проекту

Після цих операцій застосунок готовий до використання на комп'ютерах з операційною системою Windows 7 та більш пізніх.

3.5 Установка, налаштування і тестування проекту

Для використання програмного продукту, необхідно, щоб було встановлено компоненти MySQL:

- сервер MySQL;
- СУБД WorkBench;
- Notifier;
- Connectors.

Наступним етапом імпортується damp, що містить базу даних автоматизованої системи (рис. 3.21).

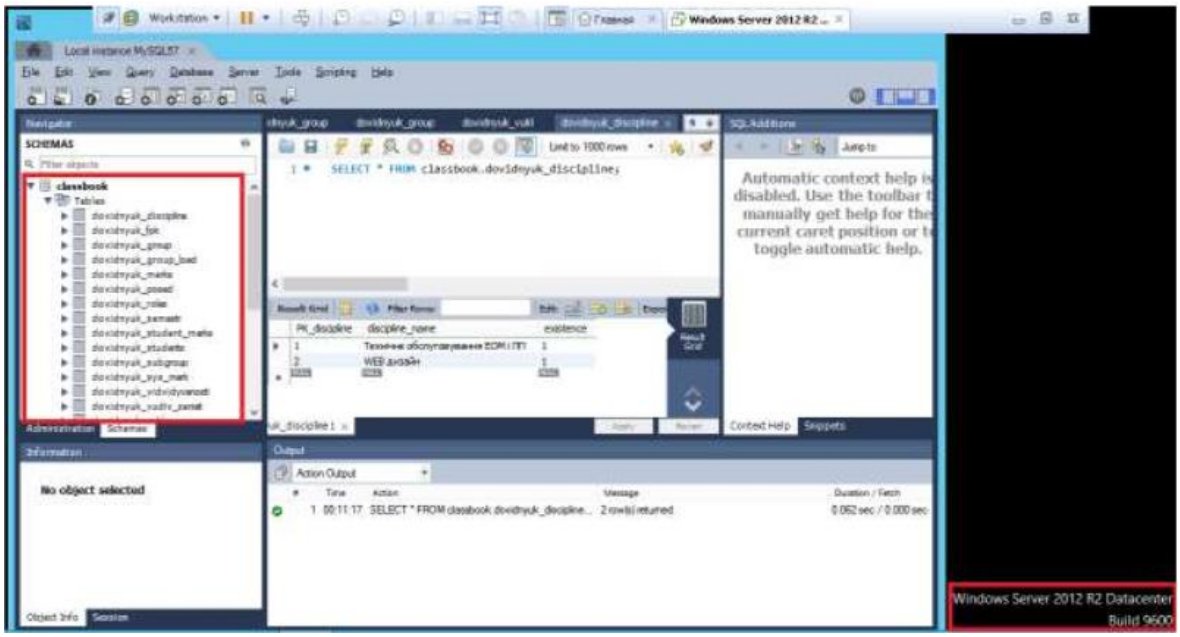

Рисунок 3.21 – Імпортовані таблиці бази даних

Останній крок - створення адміністраторів бази даних у таблиці користувачів і надання йому відповідних прав на використання функціоналу в програмному продукті (рис. 3.22).

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

Рисунок 3. 21 – SQL-запити на створення користувача.

Після встановлення та виконання необхідних налаштувань проведено перевірку з'єднання клієнтського додатку з сервером.

Проведено тестування роботи програмного продукту його роботи. Розглянуто безліч варіантів авторизації та спроб несанкціонованого використання функціоналу застосунку.

Особливу увагу приділено коректності відображення форм привнесені інформації великої довжини.

В процесі тестування було виявлено ряд невідповідностей які було усунуто.

ВИСНОВКИ

В процесі виконання кваліфікаційної роботи, та на основі триманих теоретичних знань в області організації та ведення обліку та контролю успішності школярів розроблено автоматизований програмний комплекс обліку та контролю успішності та відвідування занять.

Метою даної роботи було підвищення ефективності роботи вчителів та оперативності отримання актуальної інформації щодо обліку відвідувань занять і успішності учнів адміністрацією школи та скорочення часу на оформлення звітної документації встановленого зразку в паперовому вигляді. Детальне вивчення цього питання довело актуальність даної роботи. результати якого показали, що розробка даного завдання є актуальною.

За результатами критичного аналізу існуючих подібних систем розроблено елементи інформаційного забезпечення, на основі яких розроблена фізична модель зберігання даних. На підставі функціоналу інформаційного забезпечення розроблено алгоритм роботи та елементи інтерфейсу взаємодії користувачів з автоматизованою системою.

Розроблений програмний продукт для вирішення завдання обліку контролю успішності школярів дозволяє скоротити трудомісткі, рутинні, ручні обчислення. Програмний продукт реалізовано в середовищі розробки IntelliJ IDEA, мова розробки Java. Для зберігання даних обрано клієнт-серверну систему управління MySQL. Даний вибір елементів дозволяє успішно використовувати розроблений програмний продукт на операційних системах Windows 7 та більш новітніх.

Результати кваліфікаційної роботи можуть використовуватись працівниками закладів середньої освіти та іншими подібними освітніми закладами для організації та контролю успішності.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ДСТУ 3008-2015. Документація. Звіти у сфері науки та техніки. структура та правила оформлення. – Введ. 2015-06-22. – К. Держстандарт України, 2017 – 29 с.
2. Веб-сайт ресурсу Moodle. Головна [Електронний ресурс]. – Режим доступу: moodle.org/?lang=uk.
3. Веб-сайт платформи E-schools. Електронні щоденники та журнали [Електронний ресурс]:. – Режим доступу: e-schools.info/e-service.
4. Веб-сайт товариства «Щоденник». [Електронний ресурс]. – Режим доступу: company.shodennik.ua service.
5. Прохоренок М. А. Основи Java / М. А. Прохоренко. – СПб.: БХВПетербург, 2017. –704 с.
6. Типи даних MySQL [Електронний ресурс] – Режим доступу до ресурсу: http://www.kievoit.ippo.kubg.edu.ua/kievoit/2016/58_SQL/types.html.

ДОДАТОК А

Вихідний код основних функцій

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_dbcon
figs/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();

event.getScreenY();
                }


yOffset = primaryStage.getY() -


{
```

```
xOffset); yOffset);

});
scene.setOnMouseDragged(new EventHandler<MouseEvent>()

    @Override
  public void handle(MouseEvent event) {
     primaryStage.setX(event.getScreenX() +

               primaryStage.setY(event.getScreenY() +

                  }

              });

              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=tru
  e&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();

hile(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("ПІБ 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 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 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_CENT
                ER); 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(paragra
                ph);

                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, (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_CENT
                ER); tbl1.addCell(cell);

                        cell = 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);
                tbl1.addCell(ce
```

```
                    ll);

                    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++) {



   table1.getRowCount()) {

   {


   l).toString();


if (l < table1.getColumnCount() && i <

    if (lim > b && l < 2 || l >= used + 2) N =

        table1.getValueAt(i,

        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(c
                                   ell); 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(c
                                    ell); g++;
                                } else if (24 - lim
                                    == g) { b = 0;
                                    g = 0;
                                    l = -1;
                                }

                            } else if (table1.getColumnCount() >
l && 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));
            }

}
```

ДОДАТОК Б

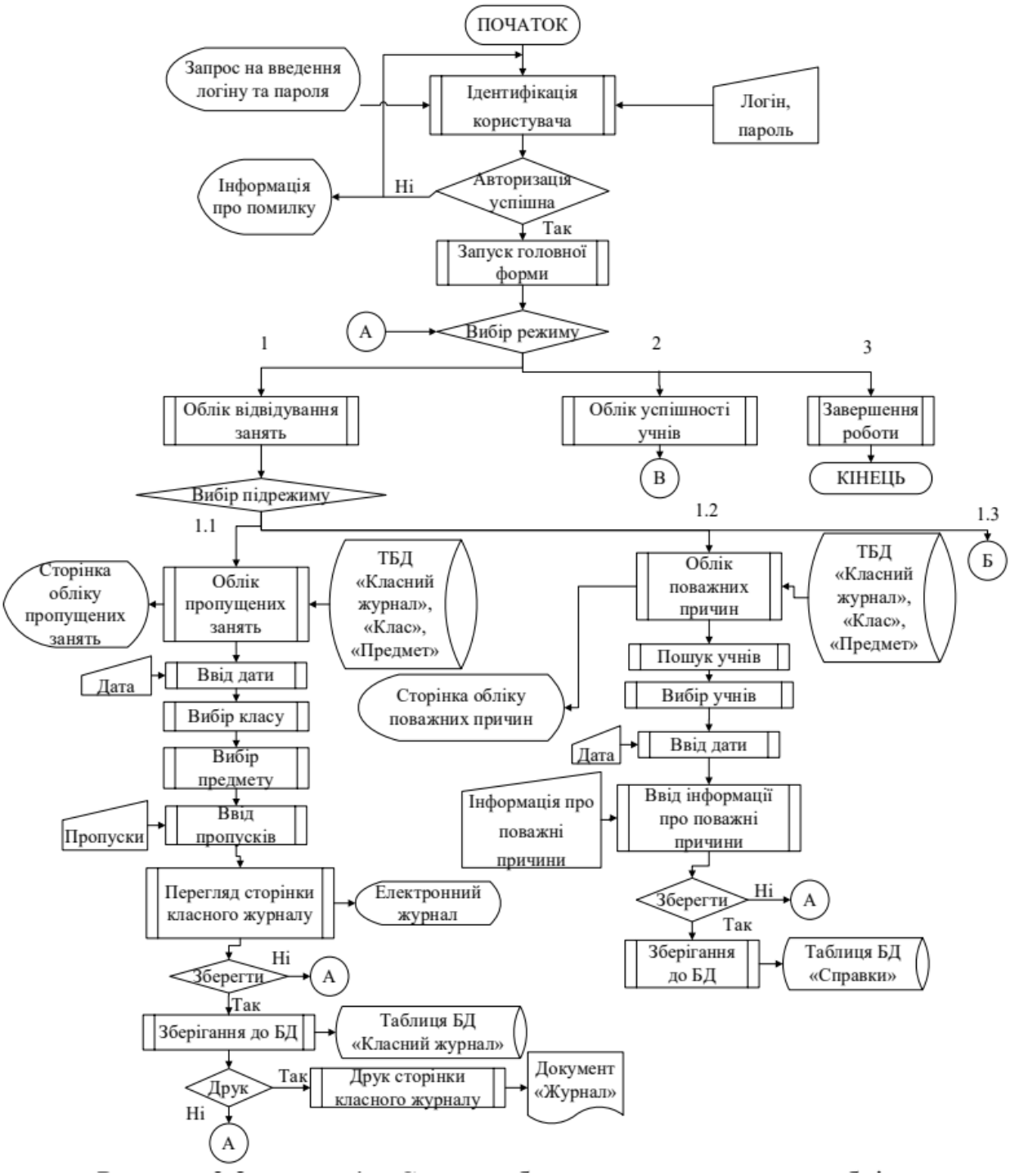

Рисунок А.1 – Узагальнений алгоритм роботи автоматизованої системи (аркуш №1)

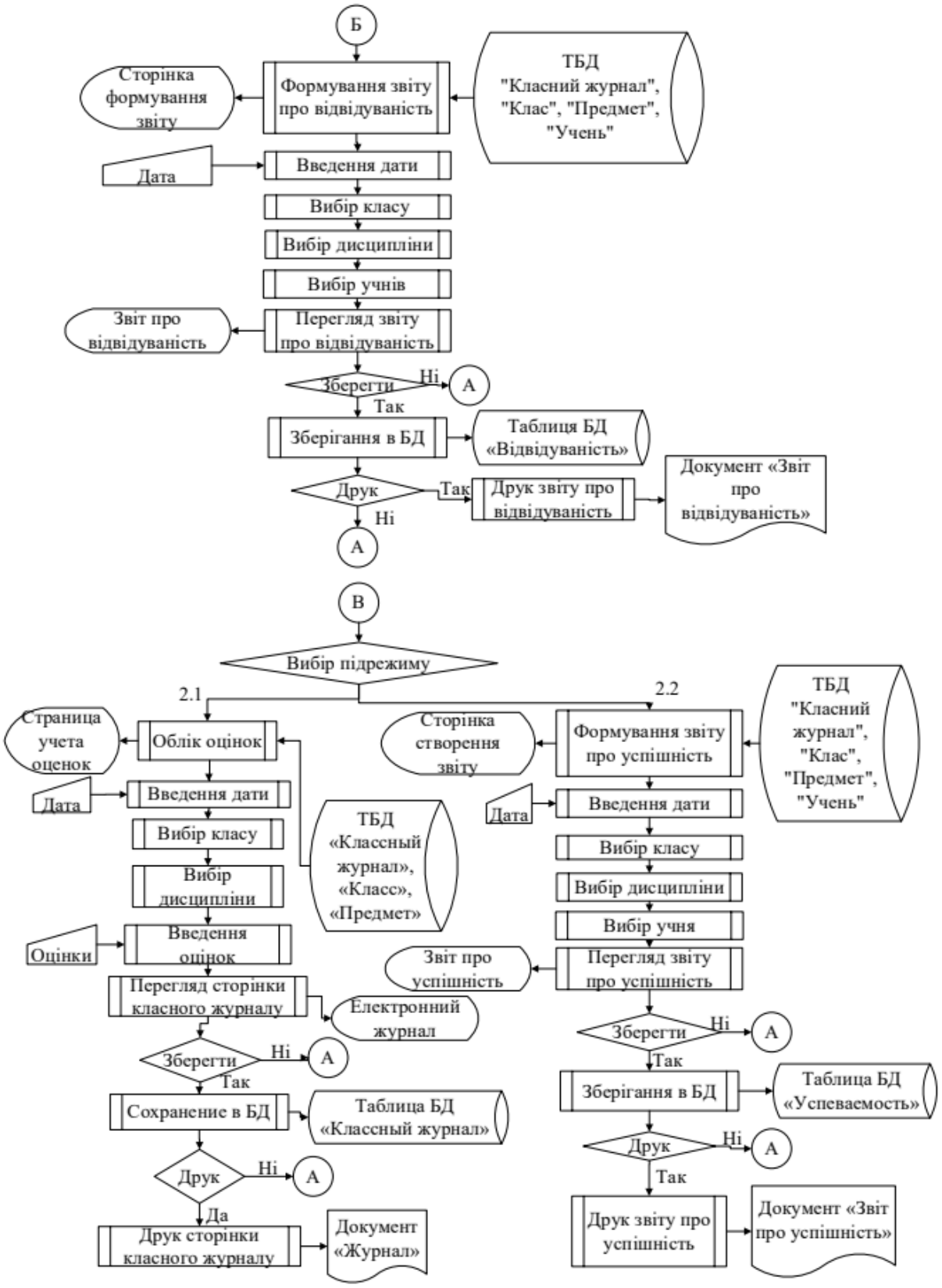

Рисунок А.1 – Узагальнений алгоритм роботи автоматизованої системи (аркуш №2)