

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ**

**КАФЕДРА КОМП'ЮТЕРНИХ НАУК**

# **КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА**

**на тему:**

**«Інформаційна система обліку студентів в умовах  
дистанційного навчання»**

**Завідувач  
випускаючої кафедри**

**Довбиш А.С.**

**Керівник роботи**

**Берест О.Б.**

**Студента групи ІІІмз – 91с**

**Бойко О.Ю.**

**Консультант  
з охорони праці**

**Гурець Л.Л.**

**Нормоконтроль**

**Проценко О.Б.**

**СУМИ 2020**

Факультет ІЗДВФН Кафедра Комп'ютерних наук  
Спеціальність «Інформатика»

Затверджую:  
зав.кафедрою \_\_\_\_\_  
“ \_\_\_\_\_ ” \_\_\_\_\_ 20\_\_ р.

## ЗАВДАННЯ НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТОВІ

Бойко Олександр Юрійович  
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Інформаційна система обліку студентів в умовах дистанційного навчання

затверджую наказом по інституту від “ \_\_\_\_\_ ” \_\_\_\_\_ 20\_\_ р. № \_\_\_\_\_

2. Термін здачі студентом закінченого проекту (роботи) \_\_\_\_\_

3. Вхідні данні до проекту (роботи) \_\_\_\_\_  
\_\_\_\_\_

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)  
1) Аналітичний огляд літератури по темі випускної роботи та аналіз проблеми. Постановка задачі дослідження. 2) Розроблення інформаційної моделі обліку успішності студентів. 3) Розробка інформаційного та програмного забезпечення системи. 4) Висновки та пропозиції щодо вдосконалення системи оцінювання.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання \_\_\_\_\_

Керівник \_\_\_\_\_  
(підпис)

Завдання прийняв до виконання \_\_\_\_\_  
(підпис)

## КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів дипломного проекту (роботи)	Термін виконання проекту (роботи)	Примітка
1.	<i>Аналітичний огляд літератури по темі випускної роботи та аналіз проблеми. Постановка задачі дослідження</i>	20.09.2020	
2.	<i>Розроблення інформаційної моделі обліку успішності студентів</i>	29.10.2020	
3.	<i>Розробка інформаційного та програмного забезпечення системи</i>	20.11.2020	
4.	<i>Висновки та пропозиції щодо вдосконалення системи оцінювання</i>	26.11.2020	

Студент – дипломник \_\_\_\_\_  
(підпис)

Керівник проекту \_\_\_\_\_  
(підпис)

## РЕФЕРАТ

**Записка:** 59 стор., 13 рис., 2 табл., 2 додаток, 31 джерел.

**Об'єкт дослідження** — системи обліку успішності студентів та методи створення веб-додатків.

**Мета роботи** — розробка онлайн системи оцінювання успішності студентів в умовах дистанційного навчання.

**Методи дослідження** — метод обліку успішності, метод створення веб-додатків.

**Результати** — розроблено онлайн журнал викладача для оцінювання успішності студентів.

СИСТЕМА ОБЛІКУ УСПІШНОСТІ, ВЕБ-ДОДАТОК,  
НАВЧАЛЬНА КАРТКА, БАЗОВІ МОЖЛИВОСТІ ОНЛАЙН-  
ЖУРНАЛУ, МЕТОД ОЦІНЮВАННЯ УСПІШНОСТІ,  
ТЕСТУВАННЯ ПРОГРАМИ.

# ЗМІСТ

ВСТУП .....	6
1 ДОСЛІДЖЕННЯ ТА АНАЛІЗ ІСНУЮЧИХ ЗАСОБІВ ОЦІНЮВАННЯ УСПІШНОСТІ СТУДЕНТІВ .....	8
1.1 Огляд аналогів програмних продуктів .....	8
1.1.1 «Журнал викладача» (SJournal) .....	8
1.1.2 TeacherEase .....	9
1.1.3 Opengrade .....	11
1.1.4 Additio .....	11
1.1.5 Gnota .....	12
1.1.6 Teacher Gradebook .....	13
1.1.7 Think Wave Administrator .....	14
1.1.8 Підсумки огляду .....	15
2 АНАЛІЗ МОЖЛИВИХ ПІШЕНЬ .....	17
2.1 Мови програмування .....	17
2.2 Система керування базами даних .....	22
2.3 Фреймворки .....	25
3 ДИЗАЙН ДОДАТКУ .....	27
3.1 Розробка ER моделі .....	27
3.2 Розробка UML діаграми класів .....	29
4 РОЗРОБКА ДОДАТКУ .....	32
4.1 Опис файлів та їх мета .....	33
4.2 Опис функцій та процесу користувача .....	35
4.3 Опис стандартних процедур та функцій, використаних у програмі .....	38
5 ТЕСТУВАННЯ ПРОГРАМИ ТА РЕЗУЛЬТАТИ .....	40
ВИСНОВКИ .....	42
СПИСОК ЛІТЕРАТУРИ .....	43
ДОДАТОК А. ІНІЦІАЛІЗАЦІЯ БАЗИ ДАНИХ .....	46
ДОДАТОК Б. ЛІСТИНГ .....	53

## ВСТУП

У наш час інформаційні технології стають дедалі популярнішими та кориснішими. За їх допомогою було автоматизовано велику частину завдань, котрі раніше виконувалися вручну. Також, інформаційні технології стають невід'ємною частиною студентського життя і одним із способів поєднання навчального процесу з сучасними інформаційними технологіями є використання електронного журналу. Їх багато в Інтернеті, і що викладач має зробити - це просто обрати та розпочати користуватися.

Засоби оцінювання роботи студентів дозволяють викладачеві організувати зворотний зв'язок зі студентом і управляти якістю навчання на курсі.

- викладач може оцінювати виконання завдань, коротких есе і тестів;
- викладач може бачити оцінки в онлайн журналі по окремих завданнях, за всі завданням, по студентам (індивідуально), по всім студентам;
- викладач може експортувати оцінки з журналу в окремі «заликовки» (spreadsheet);
- викладач може редагувати всі оцінки;
- викладач може створити шкалу для оцінки курсів, яка може працювати в наступному діапазоні: відсотки, літерне позначення, відмітка «здав» - «не здав».

Введення електронного журналу скорочує кількість часу, який викладачі витрачають на адміністративні завдання. Дуже просто налаштувати свій журнал оцінок, щоб він відображав розклад занять та спосіб оцінювання. Тоді викладач може легко вести журнал оцінок, вводячи бали, коли вони є. Вам не доведеться вручну обчислювати оцінки, оскільки вони будуть автоматично розраховані за вас.

Кожен електронний журнал має свої особливості, але є кілька загальних речей:

- online-доступ - ви можете використовувати звідник звідусіль;

- зручна навігація;
- студенти можуть отримати доступ до оцінок;
- просте оцінювання та встановлення оцінок для кожної групи;
- може мати велику кількість додаткових додаткових налаштувань.

Крім того, дуже важливим для чіткого та чесного оцінювання є те, що студент розуміє, чому він або вона отримує таку остаточну оцінку. Але не дивлячись на це, не всі викладачі навчальних закладів використовують інформаційні системи для автоматизації облік відвідування та успішності студентів. Однією з причин є невеликий вибір вже готових засобів автоматизації учбового процесу з безкоштовним доступом та зручним налаштуванням. Ось чому існує проблема розробити базові функціональні можливості користувальницького електронного журналу, який буде максимально зручним у використанні.

# **1 ДОСЛІДЖЕННЯ ТА АНАЛІЗ ІСНУЮЧИХ ЗАСОБІВ ОЦІНЮВАННЯ УСПІШНОСТІ СТУДЕНТІВ**

Відповідно зі завданням роботи необхідно розробити програмний продукт: електронний журнал викладача вищого навчального закладу. Для того, щоб виконати поставлене завдання, був виконаний аналіз аналогів, при цьому були відмічені їх сильні та слабкі сторони.

## **1.1 Огляд аналогів програмних продуктів**

Єдиних рекомендацій та вимог до електронних журналів у сфері вищої освіти не існує, тому навчальні заклади використовують різноманітні підходи и програмні продукти. В деяких університетах система обліку успішності студентів є встроєною в університетську структуру і доступ до таких систем має закритий характер.

Також, існує ряд програмних продуктів, які є у відкритому доступі і викладач може використовувати їх у своїй роботі:

- «Журнал викладача» (SJournal);
- TeacherEase;
- Opengrade;
- Addito;
- Gnota;
- Teacher Gradebook;
- Think Wave.

Розглянемо докладніше кожний із наведених програмних продуктів.

### **1.1.1 «Журнал викладача» (SJournal)**

«Журнал викладача» (SJournal) – додаток, який призначений для викладачів вищих навчальних закладі. Надає можливість вести журнали успішності та відвідування студентів [1].



Додаткові можливості:

- налаштування зведеної відомості;
- студенти можуть бути зараховані в кілька груп / підгрупи одночасно;
- налаштування графічного відображення журналу (таблиці);
- імпорт записів про студентів / групи з файлів;
- збереження / відновлення бази даних (SD-card).
- графа "Середнє значення";
- синхронізація з календарем (подія календаря повинно містити назву групи, а також аббревіатури дисципліни і виду заняття).

Позитивними сторонами продукта є безкоштовний доступ, простота завантаження із Play Market, мобільність, зручність інтерфейсу.

Недоліками SJournal є відсутність багатоплатформенності (ПО під ОС Android), відсутність імпорту даних. Додаток є пріоритетним програмним забезпеченням. Головне вікно додатку SJournal зображено на рисунку 1.1 [2].

AT-11-1   ТОЭ Зч - Теоретические основы электротехники   Лб									
№	Ф.И.О.	Лб 1 24.09.2013	Лб 2 01.10.2013	Захист Лб 1 08.10.2013	Захист Лб 2 15.10.2013	Лб 3 22.10.2013	Лб 4 29.10.2013	Лб 5.1 05.11.2013	Лб 5.2 12.11.2013
1	Белый Л.Г.			4	4				
2	Богданов С.С.			4	4	н	н	н	
3	Буц В.В.	н	н	4	4	н	н	н	н
4	Волошин А.Ю.		н	4	4		н		н
5	Иванов И.И.		н	4	4				н
6	Крамарев В.С.	н	н	4	4	н	н	н	н
7	Кулинич А.Ю.			4	4				
8	Петров Д.П.			4	4	6	н		н
9	Потапов Ю.В.			4	4	н	н	н	
10	Приморский В.А.								н
11	Романов С.М.	н	н	4	4	н	н	н	н
12	Степанов В.В.			4	4	н	н		н
	Отсутствуют	3	5	0	0	6	8	5	8

Семестр 1

- Показать/скрыть панель
- Дисциплины
- Виды занятий
- Группы
- Студенты
- Стиль таблицы
- Экспорт
- База данных
- Настройки
- Помощь
- О программе

Рисунок 1.1 - Головне вікно додатку SJournal

### 1.1.2 TeacherEase

TeacherEase – система оцінювання студентів [3]. Даний засіб оцінювання був розроблений у 2001 році, але продукт активно підтримується

й в наш час. Система має лише веб-інтерфейс, в системі існує три групи користувачів: викладач, студент та батьки.

До переваг системи можна віднести вбудований модуль LMS, в якому викладач має можливість налаштувати курс, а також тривалість курсу від 1 до 4 семестрів. Також наявний вбудований модуль комунікації студента та викладача та модуль аналізу процесу навчання та оцінювання. Продукт надає можливість до безкоштовного доступу для особистого використання викладачем, для організацій система надає платний доступ, які використовують систему: чим більше викладачів використовує систему, тим дешевше вартість одного акаунту викладача. Система відноситься до власницького програмного забезпечення.

До недоліків системи можна віднести відсутність можливості імпорту даних про студентів та відсутність експорту оцінок в .csv чи .xlsx. TeacherEase в основному розроблений та орієнтований на учбові заклади США та Канади, тому що саме ці країни є пріоритетними в налаштуваннях особистого кабінету користувача. Наведена система не підтримує сторонні LMS, в переважності LMS Moodle. Головне вікно особистого кабінету викладача системи TeacherEase наведено на рисунку 1.2 [4].

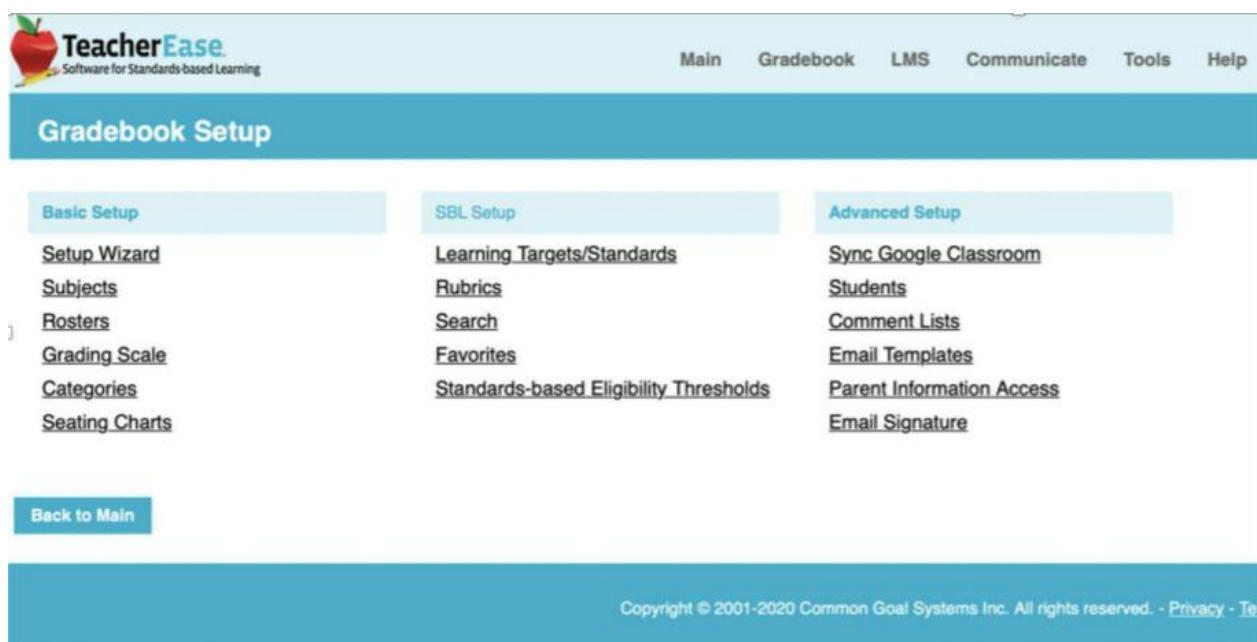
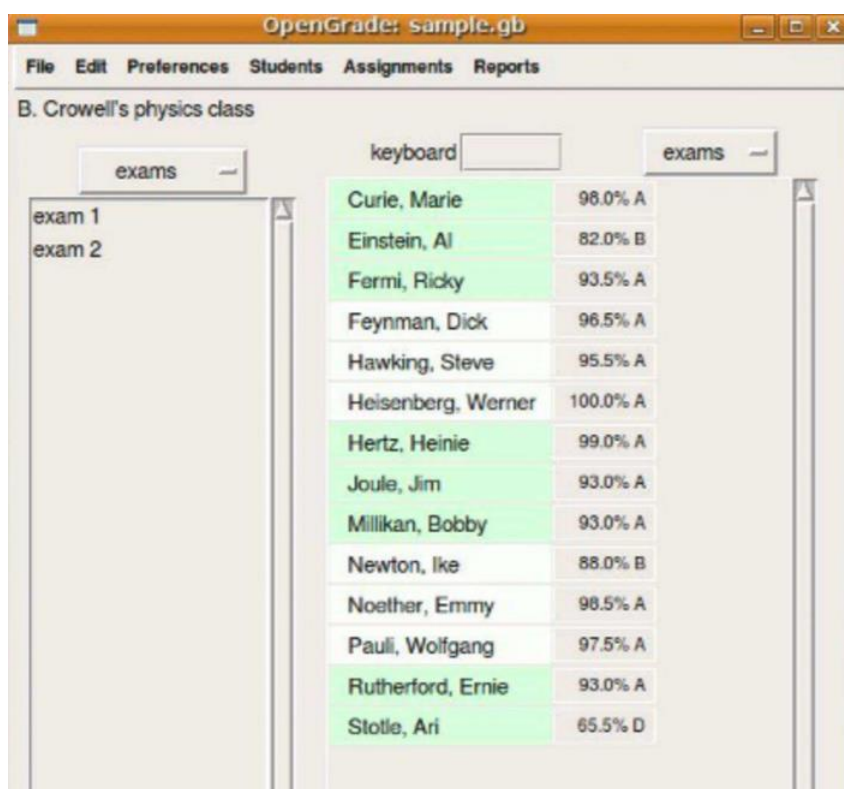


Рисунок 1.2 - Головне вікно особистого кабінету викладача системи TeacherEase

### 1.1.3 Opengrade

Opengrade – програмне забезпечення для обліку оцінок студентів, яке розповсюджене під ліцензією GPL v2. Opengrade дозволяю створювати звіти, обчислювати оцінку студентів за зазначеними формулами, захистити оцінки студентів від взлому, враховувати відвідування, експортувати в електронну таблицю та інші формати, виконувати автоматичне збереження и автоматичне резервне копіювання. Наведений продукт має версії для Linux, MacOS, Windows та FreeBSD. Інтерфейс продукту зображений на рисунку 1.3 [5].



The screenshot shows the OpenGrade application window titled 'OpenGrade: sample.gb'. The menu bar includes 'File', 'Edit', 'Preferences', 'Students', 'Assignments', and 'Reports'. The main content area is titled 'B. Crowell's physics class'. There are three dropdown menus: 'exams' on the left, 'keyboard' in the center, and 'exams' on the right. The central area displays a table of student grades.

Student Name	Percentage	Grade
Curie, Marie	98.0%	A
Einstein, Al	82.0%	B
Fermi, Ricky	93.5%	A
Feynman, Dick	96.5%	A
Hawking, Steve	95.5%	A
Heisenberg, Werner	100.0%	A
Hertz, Heinie	99.0%	A
Joule, Jim	93.0%	A
Millikan, Bobby	93.0%	A
Newton, Ike	88.0%	B
Noether, Emmy	98.5%	A
Pauli, Wolfgang	97.5%	A
Rutherford, Ernie	93.0%	A
Slote, Ari	65.5%	D

Рисунок 1.3 - Облік результатів екзамену в системі OpenGrade

### 1.1.4 Additio

Additio – додаток для IOS, Android г Web version [6]. Продукт оснащено наступними функціями: планування занять на семестр, можливість експертної оцінки та самооцінювання студентів в деяких розділах, налаштування шаблону оцінки, індивідуальні звіти, облік відвідування, розклад, календар, має здатність працювати в автономному режимі на мобільних пристроях, інтегрований з Google Classroom. Google Classroom – це сервіс Google Apps

для освіти, який дозволяє викладачам швидко створювати та систематизувати завдання, оцінювати завдання, додавати коментарі та спілкуватися з студентами [7]. Additio – підтримує функцію спілкування зі студентами та їх батьками, є можливість експортувати дані в Excel та PDF, можливість функціонування з Google Drive та Microsoft OneDrive [8]. Додаток є пріоритетним програмним забезпеченням.

Інтерфейс додатку не підтримує можливість імпорту оцінок з LMS Moodle. Інтерфейс системи зображено на рисунку 1.4 [9].

The screenshot shows the 'Digital gradebook' interface for 'Additio App'. It displays a table for '3A Language and Literature Classroom 3A' for 'Term 1'. The table includes columns for student names, exercise scores, average exercise scores, exam scores, attendance, writing and reading scores, final averages, and pass/fail status. The interface uses color coding: green for passing students and red for those who failed.

3A Language and Literature Classroom 3A		Term 1								
		+ - 10%	Exercise 1	Exercise 2	Average exercise 30%	Exam 40%	Attendance 10%	Writing and reading 10%	Final Average	Fail-Pass
1.	Collins, Adam	6	7.4	8	7.7	8.4	9.5	8.2	8.04	Pass
2.	Stond, Simon	9	8	7	7.5	7	10	6.6	7.61	Pass
3.	Gordon, Martha	8	7	9	8	7.2	10	8.8	7.96	Pass
4.	Pierce, Justin	6	9	8.5	8.75	6.8	10	6.2	7.57	Pass
5.	Booth, Peter	7	7.4	7	7.2	6.7	10	6.8	7.22	Pass
6.	Klein, Roger	4	4	4.3	4.15	6	10	7.8	5.83	Pass
7.	Farr, Amy	8	8.5	9	8.75	6.6	10	7.8	7.85	Pass

Рисунок 1.4 - Цифровий журнал обліку Additio

### 1.1.5 Gnota

Gnota – додаток для оцінювання успішності студентів, яке було розроблене під керівництвом Леандро Ламейро із Google. Дана система дозволяє здійснювати облік оцінок, відвідування занять, система надає можливість імпорту інформації про студентів та можливість збереження особистої інформації про студентів, можливість додавання декількох предметів та налаштування різноманітних фор контролю за ними [10]. Додаток розповсюджується під ліцензією GNU GPL v3.

Продукт Gnota має деякі недоліки: підтримка лише ОС Linux, відсутність імпорту даних з LMS Moodle та коректування звітів. Інтерфейс системи зображено на рисунку 1.5 [11].

Name	Homework 1	Homework 2	Homework 3	Overall score	Missed classes	Approved
Bar, Mary				F	0	<input type="checkbox"/>
Baz, Richard				F	0	<input type="checkbox"/>
Foo, John	Pass	Pass	Pass	B-	0	<input checked="" type="checkbox"/>
Quux, Foo 0				F	0	<input type="checkbox"/>
Quux, Foo 1				F	0	<input type="checkbox"/>
Quux, Foo 2				F	0	<input type="checkbox"/>
Quux, Foo 3				F	0	<input type="checkbox"/>
Quux, Foo 4				F	0	<input type="checkbox"/>
Quux, Foo 5				F	0	<input type="checkbox"/>
Quux, Foo 6				F	0	<input type="checkbox"/>
Quux, Foo 7				F	0	<input type="checkbox"/>
Quux, Foo 8				F	0	<input type="checkbox"/>
Quux, Foo 9				F	0	<input type="checkbox"/>
Quux, Foo 10				F	0	<input type="checkbox"/>
Quux, Foo 11				F	0	<input type="checkbox"/>
Quux, Foo 12				F	0	<input type="checkbox"/>

Рисунок 1.5 - Інтерфейс для оцінювання успішності Gnote

### 1.1.6 Teacher Gradebook

Teacher Gradebook – мобільний додаток для Android [12]. Особливості системи: особливий дизайн для планшетів, тжневий розклад, експорт даних в PDF та CSV, зашифрована база даних для забезпечення безпеки даних, управління студентами: додати/видалити студента, додати коментарі, вибір випадкового студента для опитування, підтримка різноманітних кольорових схем для журналу оцінювання, є можливість групування студентів чи предметів, облік відвідування, імпорт даних про студентів із формату CSV [13]. Підтримує наступні програмні продукти у сфері освіти: ClassDojo [14], Kahoot! [15], Moodle [16], Edmodo [17], Socrative [18], Google Classroom [7]. Додаток є пріоритетним програмним забезпеченням.

Недоліками системи є відсутність підтримки інших операційних систем, в тому числі ОС для стаціонарних ПК. Інтерфейс системи наведених на рисунку 1.6 [13].

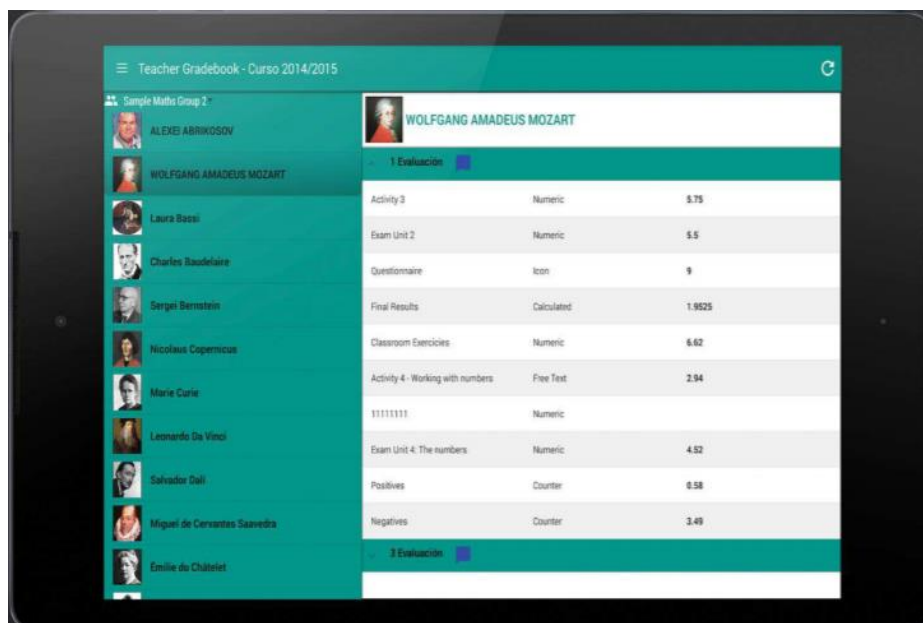


Рисунок 1.6 - Інтерфейс системи оцінювання Teacher Gradebook

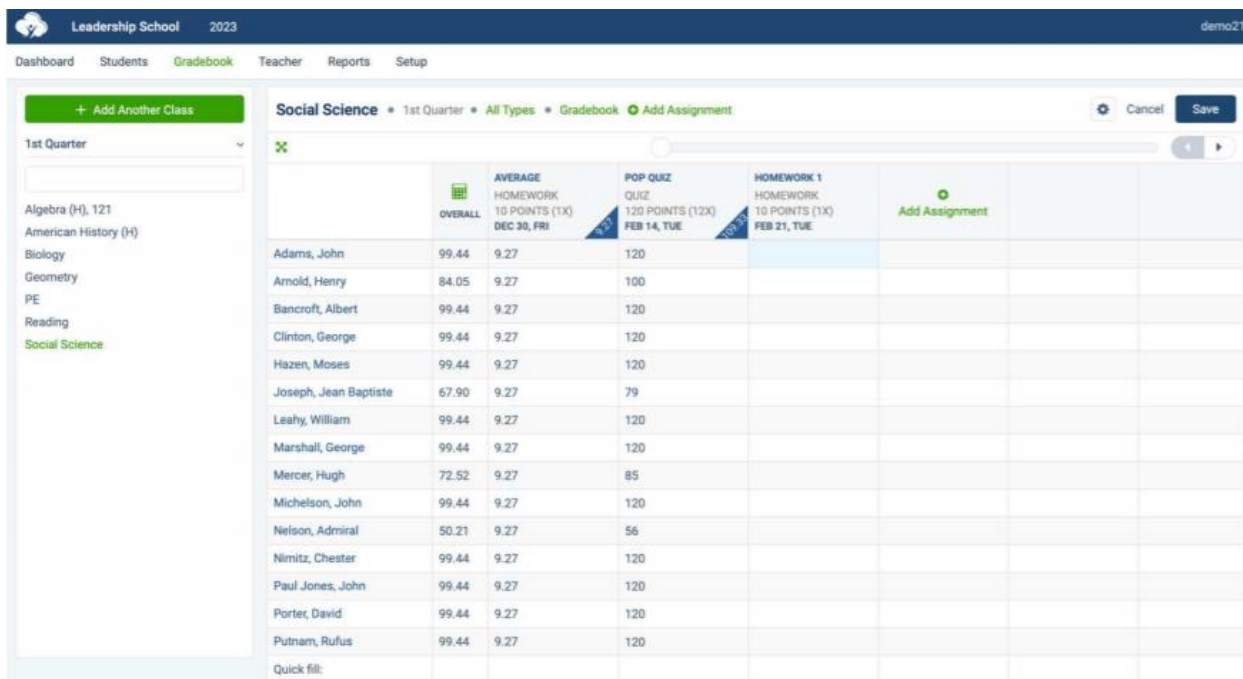
### 1.1.7 Think Wave Administrator

Think Wave Administrator – це мережева система навчального адміністрування, яка автоматично збирає оцінки, коментарі та відвідування занять, що спрощує створення зведених таблиць успішності студентів [19]. Дані з декількох джерел автоматично об'єднуються для створення звіту, котрі можна роздрукувати чи відправити по електронній пошті студентам чи батькам. Архів студентів містить результати навчання студентів на протязі всіх років навчання. «Звіт історії» в системі дозволяє вузам записувати дані минулих років для створення звітів по випускникам.

Система дозволяє завантажувати файли до 25 МБ кожний, розмір базового онлайн-сховища складає 100 ГБ, який при необхідності можна збільшити. Think Wave Administrator підтримує велику кількість форматів, серед них є .pdf, .doc, файли зображень. Робота з системою відбувається через веб-версію. Викладачі мають можливість ставити оцінки, відстежувати наявність, робити звіти, додавати коментарі, прикріплювати файли, видавати домашнє завдання та робити розсилку. Продукт є платним, ціна залежить від кількості студентів, які користуються системою. Ціна починається від 299\$ в



місяць. Відсутня підтримка офлайн доступу. Інтерфейс додатку зображено на рисунку 1.7[19].



	OVERALL	AVERAGE HOMEWORK 10 POINTS (1X) DEC 30, FRI	POP QUIZ QUIZ 120 POINTS (12X) FEB 14, TUE	HOMEWORK 1 HOMEWORK 10 POINTS (1X) FEB 21, TUE	Add Assignment
Adams, John	99.44	9.27	120		
Arnold, Henry	84.05	9.27	100		
Bancroft, Albert	99.44	9.27	120		
Clinton, George	99.44	9.27	120		
Hazen, Moses	99.44	9.27	120		
Joseph, Jean Baptiste	67.90	9.27	79		
Leahy, William	99.44	9.27	120		
Marshall, George	99.44	9.27	120		
Mercer, Hugh	72.52	9.27	85		
Michelson, John	99.44	9.27	120		
Nelson, Admiral	50.21	9.27	56		
Nimitz, Chester	99.44	9.27	120		
Paul Jones, John	99.44	9.27	120		
Porter, David	99.44	9.27	120		
Putnam, Rufus	99.44	9.27	120		

Рисунок 1.7 - Головна сторінка особистого кабінету викладача в ThinkWave

### 1.1.8 Підсумки огляду

З огляду програмних продуктів, які представлені в пунктах 1.1.1 – 1.1.7, можна зробити висновок, що всі вони призначені для обліку успішності студентів.

Кожен додаток дозволяє додавати студента до журналу та виставляти йому оцінки. Також, деякі з додатків мають особливості, які відсутні в інших програмах. Наприклад, Teacher Gradebook підтримує різноманітні кольорові схеми для журналу оцінок, Gnota та Teacher Gradebook дозволяють здійснювати імпорт даних з LMS Moodle. ThinkWave Administrator, Additio та TeacherEase є системами, які підтримують різноманітні особисті кабінети викладача та студента. Система особистого кабінету дозволяє забезпечити взаємодію викладача та студента в дистанційному режимі. Деякі програмні продукти, такі як SJournal, TeacherEase, Opengrade, Teacher Gradebook, підтримують імпорт та експорт в машинозчитуваному файловому форматі CSV та XML.

В результаті аналізу програмних продуктів, було зроблено висновок: в даний час не існує програмного продукту, який би володів повним набором необхідних функцій, зокрема в більшості розроблених продуктів відсутня можливість імпорту даних із LMS Moodle та більшість продуктів не є багатоплатформовими. Отримані дані з огляду розроблених програмних продуктів були зведені в таблицю 1.1.

Таблиця 1.1 Огляду розроблених програмних продуктів

Властивість	<b>SJournal</b>	<b>Teacher Ease</b>	<b>Opengrade</b>	<b>Addito</b>	<b>Gnota</b>	<b>Teacher Gradebook</b>	<b>ThinkWave</b>
Ліцензія	власницьке	GPL v2	власницьке	власницьке	GNU GPL v3	власницьке	власницьке
Налаштування повідомлень	-	-	-	+	-	+	+
Обмін даними з LMS Moodle	-	-	-	-	+	+	-
Можливість роботи на ПК	-	+	+	+	+	-	+
Графічний інтерфейс	+	+	+	+	+	+	+
Введення/виведення в CSV та XML форматах	+	+	+	-	-	+	-
Розрахунок оцінок за вказаними формулами	-	+	-	+	-	+	+
Створення звітів	-	-	-	+	-	+	+



## 2 АНАЛІЗ МОЖЛИВИХ ПШЕНЬ

З огляду на популярність та досягнення веб-технологій, кожному бізнесу обов'язково потрібно мати веб-сайт, який має високу функціональність та є візуально привабливий. Процес створення якісного веб-сайту, мобільних додатків чи інших подібних платформ відомий як веб-розробка.

Оскільки існує безліч різних технологій, які можна використовувати для розробки онлайн-журналу, необхідно зробити деякий огляд, щоб зрозуміти, що краще використовувати. Основними аспектами, які можна вибрати, є:

- мова програмування та відповідна IDE;
- постачальник бази даних та мова програмування;
- фреймворки, які можна використовувати для прискорення розвитку.

### 2.1 Мови програмування

Одним із найбільш невід'ємних аспектів веб-розробки є веб-програмування, яке досягається за допомогою мов програмування. Мови веб-розробки - це платформи, за допомогою яких інструкції передаються машині та виконуються дії.

Мова програмування, як правило, розділена на два компоненти - семантику та синтаксис. Якщо, з одного боку, синтаксисом є форма або тип, семантика є значенням цього типу або форми. Кожна мова програмування різна; де, з одного боку, деякі можуть бути позначені специфікаційними документами, інші можуть мати домінуючу реалізацію або посилання. Таким чином, мова програмування - це позначення, яке допомагає писати програми, які ідентифікуються як алгоритм [20].

Давайте розглянемо основні фактори при виборі мови програмування. Оскільки існує так багато різних типів мов програмування, веб-розробникові може бути важко обрати, яку використовувати, а яку залишити. Існують певні фактори, на підставі яких може бути прийнято рішення, і вони наведені нижче:

1. Цільова платформа - перше, що вам потрібно вирішити, це те, де буде запущена програма. Не всі мови здатні працювати на всіх видах платформ. Наприклад, програма, написана мовою C, вимагає, щоб компілятори працювали в системах на базі Windows та Linux.
2. Збіг мовного домену - мову потрібно вибирати на основі проблемного домену, який у вас є. Один з кращих способів це зробити, шукаючи мову, якою користуються інші в тому самому домені чи галузі, або намагаючись знайти код, який вирішує ваші проблеми.
3. Ефективність - компілятори, які добре поєднуються з вибраною Вами мовою, повинні бути ефективними, щоб пришвидшити роботу мови.
4. Еластичність та продуктивність - обрана вами мова повинна бути достатньо гнучкою, щоб була можливість додавати в неї більше програм або функцій. Крім того, загальна продуктивність повинна відповідати вашим потребам і смакам.
5. Наявність бібліотек - повинна бути бібліотека, здатна вирішити всі ваші проблеми з мовою, яку ви вибрали для веб-розробки.
6. Розмір проєкту - існує два типи програмування: велике та маленьке. Ви повинні вибрати мову, яка може підтвердити вашу справу та відповідатиме розміру проєкту.
7. Виразність та час на виготовлення - переконайтеся, що ви вибрали ту мову, яка є дуже виразною, і час, необхідний для створення програм або кодів, вас не дуже турбує.
8. Підтримка інструментів - придбайте мову, орієнтовану на інструменти, яка пропонує вам безліч елементів та способів редагування, контролю та роботи.

Вибір мов програмування широкий, і знання всіх або вивчення кожної з них не є ні практичним, ні можливим. Якщо ви розробник, який зацікавлений у вивченні найбільш корисних та популярних, то спочатку ви повинні знати, яку з тисяч мов вивчати. На щастя, існують певні веб-сайти та платформи, які створюють списки найкращих мов відповідно до їх популярності. Список 10

найкращих мов програмування цього року наведено нижче, і зрозуміло, що JavaScript є найпопулярнішим, за яким слідують Java, Python та інші.

Рейтинг мов програмування в 2020 році [21]:

1. JavaScript
2. Java
3. Python
4. PHP
5. C++
6. TypeScript
7. Swift
8. Kotlin
9. Ruby
10. Go

Давайте більш детально розглянемо деякі з цих мов. Розглянемо лише топові, які є найбільш придатними для розробки онлайн-журналу (PHP, C #, JAVA).

Термін «PHP» використовується для визначення мови PHP Hypertext Processor, яка є безкоштовною мовою сценаріїв на стороні сервера, розробленою не лише для веб-розробки, але й як платформи загального призначення для програмування [22]. Це широко вживана мова, яка була створена в 2004 році, і в даний час працює над 200 мільйонами веб-сайтів у всьому світі. Деякі популярні приклади веб-сайтів, що працюють на цій платформі, включають Facebook, WordPress та Digg.com.

PHP – це інтерпретована мова сценаріїв, що означає, що її зазвичай обробляє перекладач. З цієї причини мова є найбільш підходящою для програмування на стороні сервера, при якому серверні завдання неодноразово виконуються при увімкненому процесі розробки веб-сайту. Нижче наведено ще кілька пунктів, які допоможуть вам краще зрозуміти мову:

- PHP – це мова з відкритим кодом та мова швидкого прототипування.
- Ця мова сумісна з ОС на базі UNIX, а також з ОС Windows.

- Деякі галузі, де здебільшого використовується PHP, включають стартапи, рекламні програми та невеликі програмні організації, а також медіа-агенції.
- Мова може бути вбудована в HTML безпосередньо.

C # був призначений мовою загального призначення для продуктів Microsoft [25]. C # використовується для розробки програмного забезпечення Windows та веб-додатків на стороні сервера. Мова є частиною широко використовуваного Microsoft. Сімейство .NET, яке містить багато інших мов, поєднує їх із C # у великих проєктах. Якщо ви зацікавлені у розробці програм або веб-сайтів, специфічних для Windows, тоді C # - гарний вибір. Оскільки C # була розроблена корпорацією Microsoft для роботи на їхніх продуктах, можна з упевненістю припустити, що всі веб-програми, розроблені корпорацією Microsoft, використовують її. C # також використовується для створення програм для мобільних телефонів для операційної системи Windows Phone.

Java – ще одна дуже популярна та широко використовувана мова програмування, яку ви можете розглянути для веб-розробки [23]. Ця мова є об'єктно-орієнтованою, заснованою на класах і одночасно мовою, яка була розроблена компанією Sun Microsystems у 1990-х. З тих пір мова продовжує залишатися найбільш затребованою мовою, яка також діє як стандартна платформа для підприємств та декількох розробників мобільних пристроїв та ігор у всьому світі. Додаток був розроблений таким чином, що він працює на декількох типах платформ. Це означає, що якщо програма написана в операційній системі Mac, вона також може працювати в операційних системах на базі Windows.

Java, коли вона була розроблена спочатку, була розроблена для інтерактивного телебачення, але розробники зрозуміли, що ця мова та технологія були надто передовими для цієї галузі. Лише пізніше він був включений до того вжитку, яким він служить сьогодні.

Кожна мова створюється з урахуванням певної місії, мети чи завдання. Нижче наведено 5 основних принципів або цілей, які мали на увазі під час створення цієї мови:

- Синтаксис мови повинен бути «простим, об'єктно-орієнтовним та звичним».
- Реалізація має бути «безвідмовною та безпечною».
- Повинна зберегтися «незалежність від архітектури та переносність».
- Висока продуктивність виконання
- Мова має бути «інтерпретованою, багатонитевою, із динамічним зв'язуванням модулів».

Хоча це особистий вибір, програмісти також повинні стежити за новими подіями, що відбуваються в цьому просторі. Кожен розробник повинен постійно отримувати інформацію про нові мови та функції. Згідно з вищезазначеними твердженнями та перевагами мов програмування, Java була обрана базовою технологією для розробки онлайн журналу.

## 2.2 Система керування базами даних

У сучасних інформаційних системах для забезпечення роботи з базами даних використовують системи керування базами даних (СКБД). Система керування базами даних – це система, заснована на програмних та технічних засобах, яка забезпечує визначення, створення, маніпулювання, контроль, керування та використання баз даних. Застосунки для роботи з базою даних можуть бути частиною СКБД або автономними. Найпопулярнішими СКБД є MySQL, PostgreSQL, Microsoft SQL Server, Oracle, Sybase, Interbase, Firebird та IBM DB2. СКБД дозволяють ефективно працювати з базами даних, обсяг яких робить неможливим їх ручне опрацювання.

Через тісний зв'язок баз даних з СКБД під терміном «база даних» інколи необґрунтовано та неточно мають на увазі систему керування базами даних. Але варто розрізняти базу даних – сховище даних, та СКБД – засоби для роботи з базою даних. СКБД з інформаційної системи може бути видалена, але база даних продовжить існувати. І навпаки: СКБД може функціонувати без жодної бази даних [24]. У наведеному нижче списку ми коротко поговоримо про кожен СУБД, наведемо деякі плюси та недоліки.

**MySQL Community Server.** MySQL-сервер баз даних був розроблений компанією ТсХ для внутрішнього споживання і швидкої обробки великих баз даних. Сьогодні це одне з найкращих рішень для малих та середніх програмних додатків, а вихідні сервера баз даних MySQL компілюють на багатьох платформах. Найкраще використовувати MySQL-сервер на Unix-серверах - вони підтримують багатопоточність, що забезпечує значний приріст виробництва. В результаті можливості сервера відкриваються на повну потужність.

На сьогоднішній день MySQL являє собою безкоштовний сервер, якщо мова йде про некомерційні використання. У зворотному випадку вам знадобиться придбати ліцензію.

MySQL-сервер підтримує мову SQL-запитів і стандарт ANSI 92. Крім того, підтримується ряд розширень до цього стандарту, що вигідно відрізняє MySQL-сервер від конкурентів. Основні функціональні можливості:

- підтримується необмежена кількість користувачів, що одночасно працюють в базах даних;
- число табличних рядків може досягати 50 мільйонів, чого більше, ніж достатньо;
- сервер MySQL вважається одним з кращих за швидкістю;
- система безпеки в MySQL-сервері проста і ефективна.
- З недоліків MySQL можна назвати відсутність:
  - підтримки різних транзакцій;
  - підтримки тригерів та збережених процесів;
  - підтримки уявлення.

При цьому перераховані вище недоліки не є критичними при розробці додатків.

**Microsoft Access.** На відміну від MySQL, доступ Microsoft не є безкоштовним. Однак, оскільки він є частиною Microsoft Office, у багатьох розробників він уже є. Навіть якщо ви не використовуєте Microsoft Office, ви все одно можете завантажити пробну версію Access, яка дасть вам 60 днів, щоб пограти з нею та перевірити, чи вона для вас.

Доступ має деякі ключові подібності до MySQL. Це дозволяє вам візуально розробити таблицю або завдання, а потім побачити створений синтаксис. Є також позитивні та негативні відмінності. Позитивним є те, що Access не має двох окремих інсталяцій (одна для СУБД та інша для інструменту проєктування), а все, швидше за все, є єдиною програмою. З іншого боку Access має деякі суттєві обмеження. Не на всіх операційних системах його можна встановити. Більш важливим є те, що він має серйозне обмеження щодо кількості одночасних з'єднань, які він може обробити до погіршення продуктивності. Це означає, що Access ідеально підходить для

навчання та для веб-сайтів для любителів із низьким рівнем відвідуваності, але стає млявим та непридатним для веб-сайтів, які мають помірний або більший трафік.

**Microsoft SQL Server Express.** За допомогою цієї СУБД та наступних двох ми підходимо до найбільш популярних - Microsoft SQL Server, Oracle та DB2. Кожна з них має окремі засоби проектування та серверні програми, які необхідно встановити.

Справді, Microsoft SQL Server, Oracle Express та DB2 Express дуже схожі за інструментами та функціональністю. Найбільші їхні відмінності полягають у синтаксисі SQL. Варіант Microsoft SQL Server має один недолік, оскільки операційні системи, на яких він буде працювати, обмежені.

**Oracle Express Edition.** Oracle Express також має інструменти та окремий серверний додаток. Він також має більше опцій операційної системи, ніж Microsoft SQL Server Express. Варіанти завантаження також набагато ширші, ніж у інших СУБД в цьому списку.

**DB2 Express-C.** Параметри встановлення та підтримувані операційні системи DB2 Express-C дуже схожі на такі, як у Oracle Express. Однак, інтернет-підручників, статей та прикладів, не так багато, як для інших варіантів у цьому списку. Ви також можете завантажити DB2 Express-C для роботи.

Ми збираємось використовувати SQL із технологією веб-розробки. Тому важливо вибрати СУБД, яка створена або створена для роботи з вибраною веб-технологією. Наприклад, розробники ASP.NET зазвичай вибирають Microsoft SQL Server або Microsoft Access, розробники PHP зазвичай вибирають MySQL або Oracle, а розробники Java/WebSphere зазвичай вибирають DB2 або Oracle. Тож ми зупинилися на Oracle SQL [25].



## 2.3 Фреймворки

Програмний фреймворк – це готовий до використання комплекс програмних рішень, включаючи дизайн, логіку та базову функціональність системи або підсистеми. Відповідно — програмний фреймворк може містити в собі також допоміжні програми, деякі бібліотеки коду, скрипти та загалом все, що полегшує створення та поєднання різних компонентів великого програмного забезпечення чи швидке створення готового і не обов'язково об'ємного програмного продукту. Побудова кінцевого продукту відбувається, зазвичай, на базі єдиного API.

Структури програмного забезпечення, як правило, містять значний код ведення організаційного обслуговування та утиліти, щоб допомогти завантажити користувацькі програми, але, як правило, зосереджуються на конкретних проблемних доменах, таких як:

- Художній малюнок, музична композиція та механічні CAD;
- Програми фінансового моделювання;
- Застосування моделювання систем Землі;
- Системи підтримки прийняття рішень;
- Відтворення та створення медіафайлів;
- Фреймворк Ajax / фреймворк JavaScript;
- Веб-фреймворк;
- Проміжне програмне забезпечення;
- Cactus Framework - високопродуктивні наукові обчислення
- Фреймворк додатків - загальний графічний інтерфейс;
- Структура архітектури підприємства;
- Oracle Framework для розробки додатків.

У нашому випадку ми повинні розглянути кращі веб-фреймворки. Веб-фреймворк або фреймворк веб-програм - це фреймворк, призначений для підтримки розробки веб-додатків, включаючи веб-сервіси, веб-ресурси та веб-API. Веб-фреймворки забезпечують стандартний спосіб створення та розгортання веб-додатків. Веб-фреймворк спрямований на автоматизацію накладних витрат, пов'язаних із загальними діями, що виконуються у веб-розробці. Наприклад, багато веб-фреймворки надають бібліотеки для доступу до баз даних, шаблонів та управління сесіями, і вони часто сприяють повторному використанню коду. Хоча вони часто націлені на розробку динамічних веб-сайтів, вони також застосовні до статичних веб-сайтів.

Більшість веб-фреймворків базуються на шаблоні модель-вигляд-контролер (MVC) [26]. Багато фреймворків дотримуються архітектурного шаблону MVC, щоб відокремити модель даних за допомогою бізнес-правил від інтерфейсу користувача. Зазвичай це вважається гарною практикою, оскільки воно модулює коди, сприяє повторному використанню коду та дозволяє застосовувати кілька інтерфейсів. У веб-додатках це дозволяє представляти різні подання, наприклад веб-сторінки для споживача та інтерфейси веб-служб для віддалених програм.

## 3 ДИЗАЙН ДОДАТКУ

Першим кроком у розробці будь-якої програми є створення різних допоміжних діаграм, які можуть спростити весь процес розробки. Для розробки електронного журналу необхідно розробити дві основні схеми:

- ER - діаграма;
- Діаграма класів UML.

### 3.1 Розробка ER моделі

Модель «сутність-зв'язок» (ER-модель) це модель даних, яка дозволяє описувати концептуальні схеми за допомогою узагальнених конструкцій блоків. ER-модель - це мета-модель даних, тобто засіб опису моделей даних. Існує ряд моделей для представлення знань, але одним з найзручніших інструментів уніфікованого представлення даних, незалежного від програмного забезпечення, що його реалізує, є модель «сутність-зв'язок». Важливим є той факт, що з моделі «сутність-зв'язок» можуть бути породжені всі існуючі моделі даних (ієрархічна, мережева, реляційна, об'єктна), тому вона є найзагальнішою.

У програмній інженерії модель ER зазвичай формується для представлення речей, які бізнес повинен пам'ятати для здійснення бізнес-процесів. Отже, модель ER стає абстрактною моделлю даних, яка визначає структуру даних або інформації, яка може бути реалізована в базі даних, як правило, в реляційній базі даних. Модель сутність-зв'язок є результатом систематичного процесу, який описує та визначає деяку предметну область. Вона не визначає сам процес, а лише візуалізує його. Дані представлені у вигляді компонентів (сутностей), які пов'язані між собою певними зв'язками, які виражають залежності і вимоги між ними.

Коли ми говоримо про сутність, ми зазвичай говоримо про деякий аспект реального світу, який можна виділити поміж інших аспектів. Сутність - це збірне поняття, деяка абстракція реального об'єкта, процесу, явища чи деякого

уявлення про об'єкт. Для розробки онлайн журналу книжки було створено п'ять сутностей (STUDENT, LOCATION, COURSE, GRADE). На рисунку 3.1 зображено ER-модель з необхідними зв'язками між сутностями та всіма атрибутами.

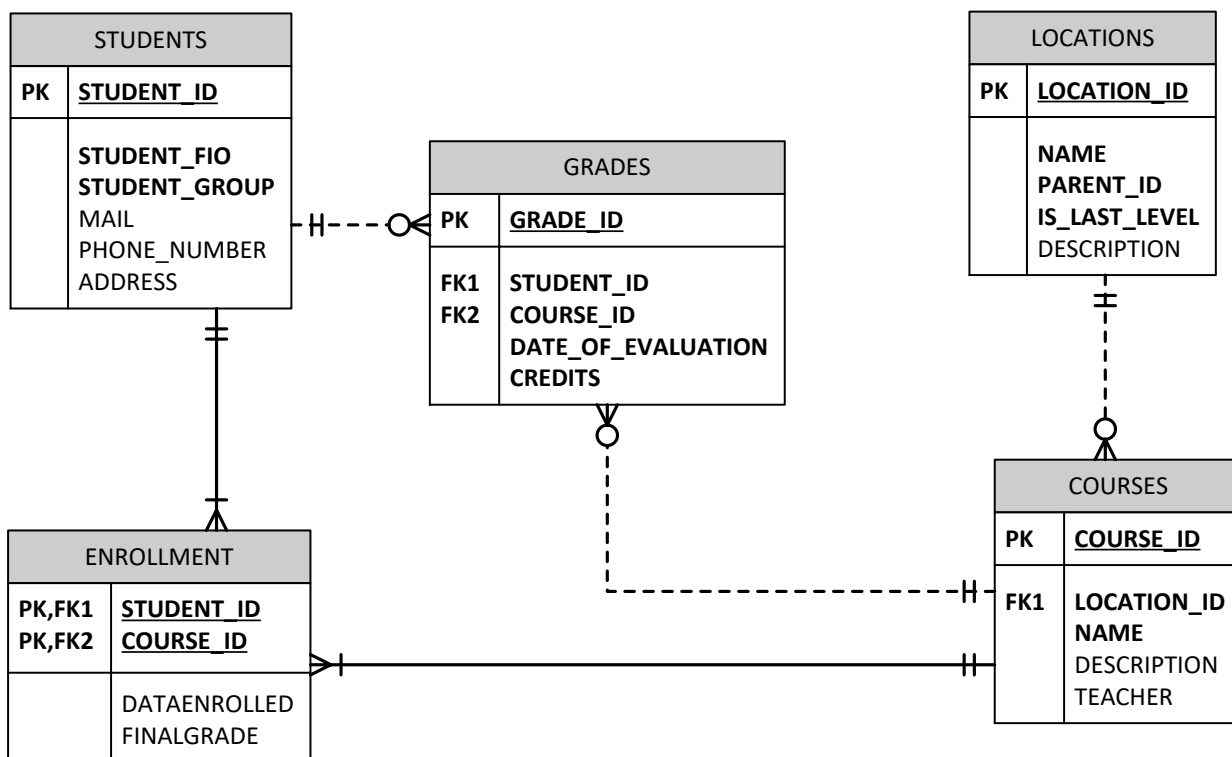


Рисунок 3.1 - ER-модель для розробки онлайн-журналу

## 3.2 Розробка UML діаграми класів

У програмній інженерії діаграма класів в Уніфікованій мові моделювання (UML) – це тип статичної структурної діаграми, що описує структуру системи, показуючи класи системи, їх атрибути, операції (або методи) та взаємозв'язки між об'єктами.

Діаграма класів є основним будівельним елементом об'єктно-орієнтованого моделювання. Він використовується як для загального концептуального моделювання систематики програми, так і для детального моделювання переведення моделей в код програмування. Діаграми класів також можна використовувати для моделювання даних. Класи в діаграмі класів представляють як основні елементи, взаємодії в програмі, так і класи, що програмуються.

На схемі класи представлені вікнами, які містять три відділення:

- 1 Верхній відсік містить назву класу.
- 2 Середній відсік містить атрибути класу.
- 3 У нижньому відділенні містяться операції, які клас може виконувати.

При проєктуванні системи ряд класів ідентифікується і групується в схему класів, яка допомагає визначити статичні зв'язки між ними. При детальному моделюванні класи концептуального проєкту часто поділяються на ряд підкласів.

Щоб розробити концептуальну схему класів, необхідно ітеративно змоделювати:

- Заняття
- Обов'язки
- Асоціації
- Спадкові відносини
- Композиційні асоціації
- Словники

Щоб розробити дизайн схему класу, необхідно ітеративно змоделювати:

- Заняття
- Обов'язки
- Асоціації
- Спадкові відносини
- Композиційні асоціації
- Інтерфейси

Для створення додатку онлайн-журнал була створена діаграма класів UML відповідно до шаблону проектування MVC. На рисунку 3.2 зображена діаграма класів UML з необхідними зв'язками між класами та всіма атрибутами, такими як поля та методи. Схема класів була створена за допомогою інтегрованого плагіна в IDE IntelliJ Idea Ultimate.

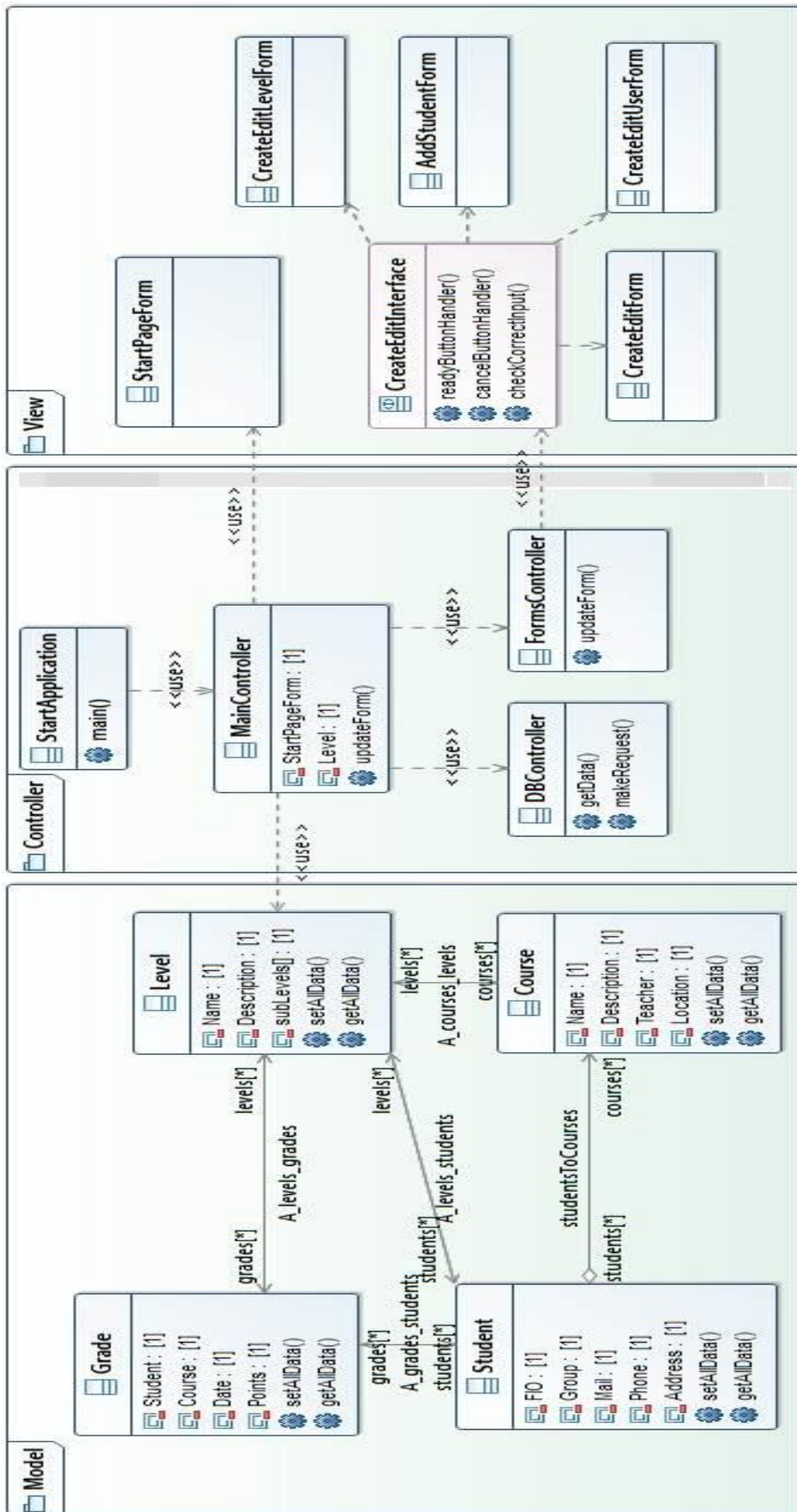


Рисунок 3.2 - Діаграма класів UML

## 4 РОЗРОБКА ДОДАТКУ

На початку створення онлайн-додатку необхідно вибрати зручні засоби розвитку. Найпопулярніший програмний додаток для Java EE, який надає комплексні можливості для програмістів для розробки програмного забезпечення - інтегроване середовище розробки (IDE):

- Eclipse Neon EE (безкоштовний доступ);
- Професійні інструменти для розробників від JetBrains, IntelliJ IDEA Ultimate (необхідний дозвіл).

Оскільки шанс отримати доступ до IntelliJ IDEA Ultimate великий, і все, що вам потрібно, це подати заявку, бути студентом та мати доступ до вашої адреси електронної пошти або дійсної картки ISIC. Тож було обрано цю IDE. Він надає повний набір інструментів для розробки Інтернету, мобільних пристроїв та підприємств на основі JVM.

IntelliJ IDEA являє собою високотехнологічний комплекс тісно інтегрованих інструментів програмування, що включає інтелектуальний редактор вихідних текстів з розвиненими засобами автоматизації, потужні інструменти рефакторинга коду, вбудовану підтримку технологій J2EE, механізми інтеграції з середовищем тестування Ant / JUnit і системами управління версіями, унікальний інструмент оптимізації та перевірки коду Code Inspection, а також інноваційний візуальний конструктор графічних інтерфейсів.

Отже, завдання полягає в розробці програми Java Enterprise Edition, яка буде розміщена на веб-сервері для тестування та використання. Як веб-сервер використовується Oracle WebLogic Server. Oracle WebLogic Server 12c R2 - це галузевий прикладний сервер для побудови та розгортання корпоративних програм Java EE з підтримкою нових функцій для низьких витрат операцій, підвищення продуктивності, підвищення масштабованості та підтримки портфеля програм Oracle.



Додатки WebLogic Server Java EE базуються на стандартизованих модульних компонентах; WebLogic Server надає повний набір послуг для цих модулів і автоматично обробляє багато деталей поведінки додатків, не вимагаючи програмування.

## 4.1 Опис файлів та їх мета

Дуже важливим кроком розробки є створення правильної, гнучкої та масштабованої архітектури додатків [27-31]. На рисунку 4.1 зображена структура проекту.

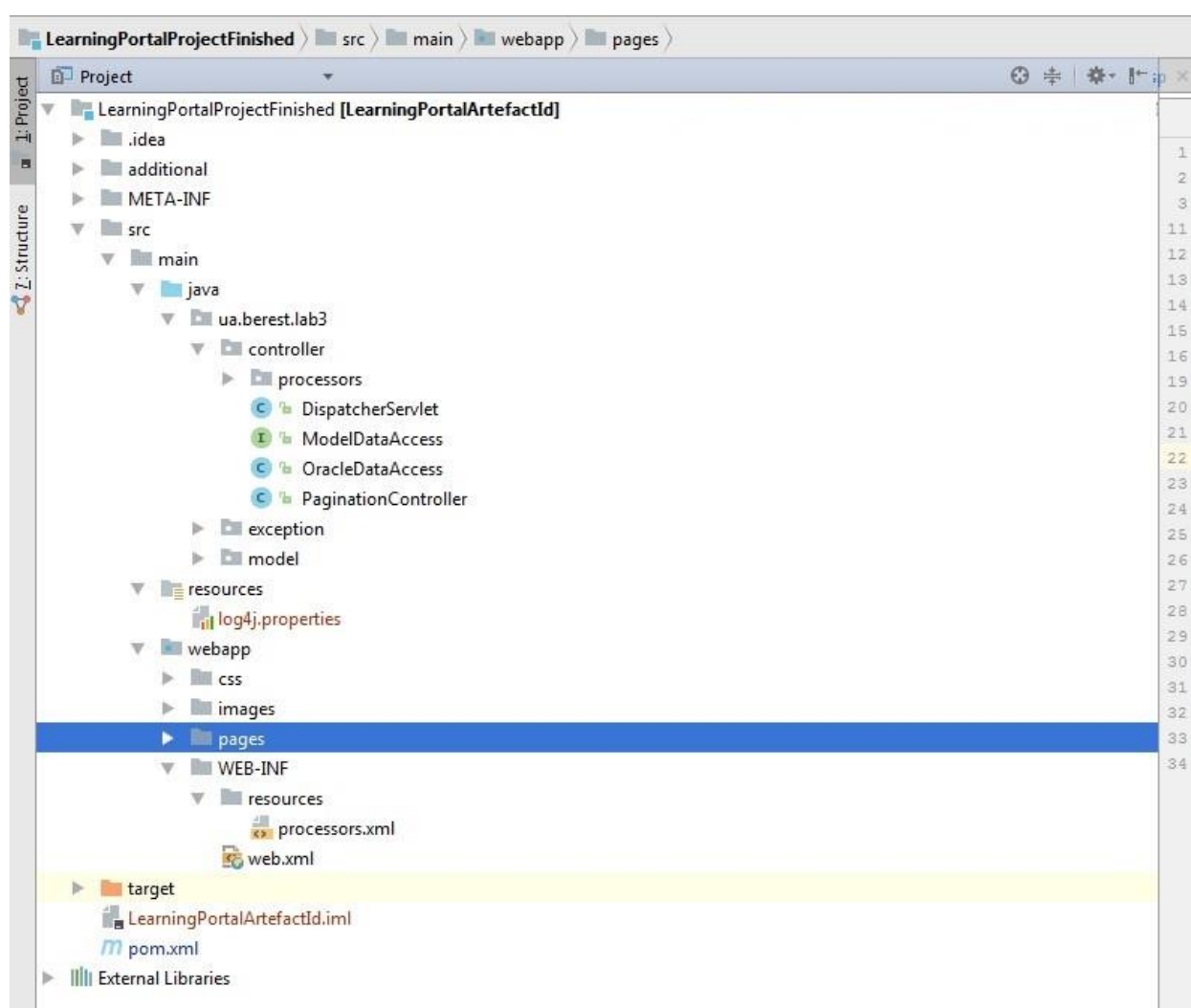


Рисунок 4.1 - Структура проекту

В таблиці 4.1 наведено опис та призначення головних файлів в структурі проєкту.

Таблиця 4.1 Опис та призначення головних файлів в структурі проєкту

<b>Файл</b>	<b>Опис</b>
.idea : folder	files, that are required for IDEA project
additional : folder	optional files, that do not take part in project lifecycle
META-INF : folder	Optional folder for web project, is not used
src/main : folder	<ul style="list-style-type: none"> <li>• java (folder with source files due to MVC pattern) <ul style="list-style-type: none"> <li>○ controller</li> <li>○ exception</li> <li>○ model</li> </ul> </li> <li>• resources (files for logging)</li> <li>• webapp (web part of application) <ul style="list-style-type: none"> <li>○ css (files that are responsible for html style)</li> <li>○ images (images in the project)</li> <li>○ pages (contains jsp-pages)</li> <li>○ WEB-INF (web-descriptor of project) <ul style="list-style-type: none"> <li>▪ resources (xml – required file)</li> <li>▪ web.xml</li> </ul> </li> </ul> </li> </ul>
target : folder	Output folder of project where compiled classes and required jar files are located
LearningPortalArtefact Id.iml	Identifier for Idea project
pom.xml	File which is required in case of Maven usage
External Libraries : folder	External libraries to resolve class import statements

Як можна побачити з таблиці 4.1, структуру проєкту можна поділити на дві основні частини - вихідні файли, які групуються завдяки реалізації

шаблону MVC, та веб-частина, яка дозволяє нам мати візуальний інтерфейс програми в будь-якому браузері.

Всі ці файли необхідні для розгортання програми на сервері Weblogic та її запуску.

Спрощена логіка застосування буде наступною:

1. Після розгортання програми на сервері відображається сторінка привітання (welcome.jsp). Це вимагає входу для підтримки користувача під час сеансу;
2. Далі всі подальші дії обробляються ядром програми - класом DispatcherServlet, який має функцію обробки всіх запитів на сервер weblogic та адресу вихідної програми;
3. Клас DispatcherServlet завантажує всі доступні процесори. Кожен процесор може працювати зі спеціальною дією (змінною Sting), яка визначає його назву та поведінку;
4. Майже всі інші дії мають подібний алгоритм. Згідно з процесором дій за допомогою класу OracleDataAccess витягує необхідні дані з бази даних Oracle і відображає їх на відповідній jsp-сторінці.
5. Якщо є проблема з підключенням або логікою програми, ми отримуємо об'єкт DataAccessExcerpton, який містить детальну інформацію про проблему.

## **4.2 Опис функцій та процесу користувача**

Існує більше 50 Java-класів, які створюють структуру проєкту. Майже кожен клас має свій набір полів і методів. Пакети являють собою файлову та логічну структуру зв'язків класів в світі java. Дуже схоже з файловою системою комп'ютера. На рівні файлової системи пакети це і є папки, в яких лежать інші папки (підпаки) і класи. Але пакети не завжди описують безпосередньо всю структуру проєкту. На практиці проєкт включає в себе різні ресурси, а структура папок, яку ми призначаємо як імена пакетів для наших класів - може

бути лише невеликою частиною цілого проєкту. Адже, крім основного коду в пакетах, у нас повинні бути ще й тести, бібліотеки або навіть інші мови програмування в проєкті в цілому.

Пакети необхідні не тільки для зрозумілого розміщення файлів по папках, але і для поділу доступу класів один до одного. Саме завдяки пакетам ми можемо через крапку «.» отримувати доступ до потрібних нам класів різних бібліотек.

Для класу його пакет - це його місце розташування в проєкті, щодо інших класів. Завдяки розподілу класів на кілька пакетів - ми організуємо структуру програми.

Давайте розглянемо деякі приклади з більшості важливих пакетів.

- package model, class StudentImpl

```
public class StudentImpl implements Student {
    private int studentId;
    private String fio;
    private String group;
    private String mail;
    private String phone;
    private String address;

    public StudentImpl(int studentId, String fio, String group, String mail, String phone, String
address) {
        this.studentId = studentId;
        ...
    }
    public int getStudentId() {
        return studentId;
    }
    public void setStudentId(int studentId) {
        this.studentId = studentId;
    }
    @Override
    public int hashCode() {
return toString().hashCode();
    }
}
```

```

@Override
public boolean equals(Object obj) {
    if (this == obj)
        return true;
    if (obj == null)
        return false;
    if (getClass() != obj.getClass())
        return false;
    StudentImpl other = (StudentImpl) obj;
    return !(studentId != other.studentId || !fio.equals(other.fio) || !group.equals(other.group) ||
    !mail.equals(other.mail) || !phone.equals(other.phone) || !address.equals(other.address));
}
@Override
protected Object clone() {
    try{
        StudentImpl copy = (StudentImpl) super.clone();
        return copy;
    }
    catch(CloneNotSupportedException e){
        throw new AssertionError("Impossible");
    }
}
@Override
public String toString() {
    return "Student id = " + studentId + ", full name = " + fio + ", e-mail = " + mail + ",
    phone number = " + phone + ", home address = " + address + ", group = " + group + ".";
}
}

```

Класи цього пакету - це прості класи Java Bean, що включає необхідні поля, геттери та сеттери для цього поля, конструктор для ініціалізації та створення екземплярів об'єктів та деякі перевизначені методи використання об'єктів цих класів у колекціях.

- package controller.processors, class Processor

```

public abstract class Processor {
    protected String actionToPerform = null;
    public boolean canProcess(String action) {

```

```

        return action.equals(actionToPerform);
    }
    public abstract ProcessorResult process(HttpServletRequest request) throws
        DataAccessException;
}

```

Кожен представник цього пакету має лише одне поле “action”, яке визначає його поведінку, метод `canProcess ()`, що дозволяє об’єкту цього класу взяти під контроль і метод `process ()`, який визначає, що саме робить процесор.

### **4.3 Опис стандартних процедур та функцій, використаних у програмі**

Усі функції Java розташовані в класах. Тож краще перерахувати класи, які використовуються в додатку:

- Класи для підключення до бази даних
  - `import java.sql.*;`
  - `import java.sql.Date;`
  - `import java.util.*;`
  - `import javax.naming.*;`
  - `import javax.sql.DataSource;`
- Класи для логіну в систему
  - `import org.apache.log4j.Logger;`
- Класи для роботи з файлами xml
  - `import org.w3c.dom.*;`
  - `import org.xml.sax.SAXException;`
  - `import javax.xml.parsers.DocumentBuilder;`
  - `import javax.xml.parsers.DocumentBuilderFactory;`
  - `import javax.xml.parsers.ParserConfigurationException;`
- Класи для роботи з http протоколами
  - `import javax.servlet.RequestDispatcher;`
  - `import javax.servlet.ServletException;`

- `import javax.servlet.http.HttpServlet;`
- `import javax.servlet.http.HttpServletRequest;`
- `import javax.servlet.http.HttpServletResponse;`
- Класи для роботи з колекціями
  - `import java.util.ArrayList;`
  - `import java.util.List;`
- Класи для роботи з датами та файлами
  - `import java.io.File;`
  - `import java.io.IOException;`
  - `import java.util.Date;`

## 5 ТЕСТУВАННЯ ПРОГРАМИ ТА РЕЗУЛЬТАТИ

Після успішного завантаження програми можна розпочинати роботу с онлайн-журналом. На рисунку 5.1 зображена сторінка для авторизації в систему.

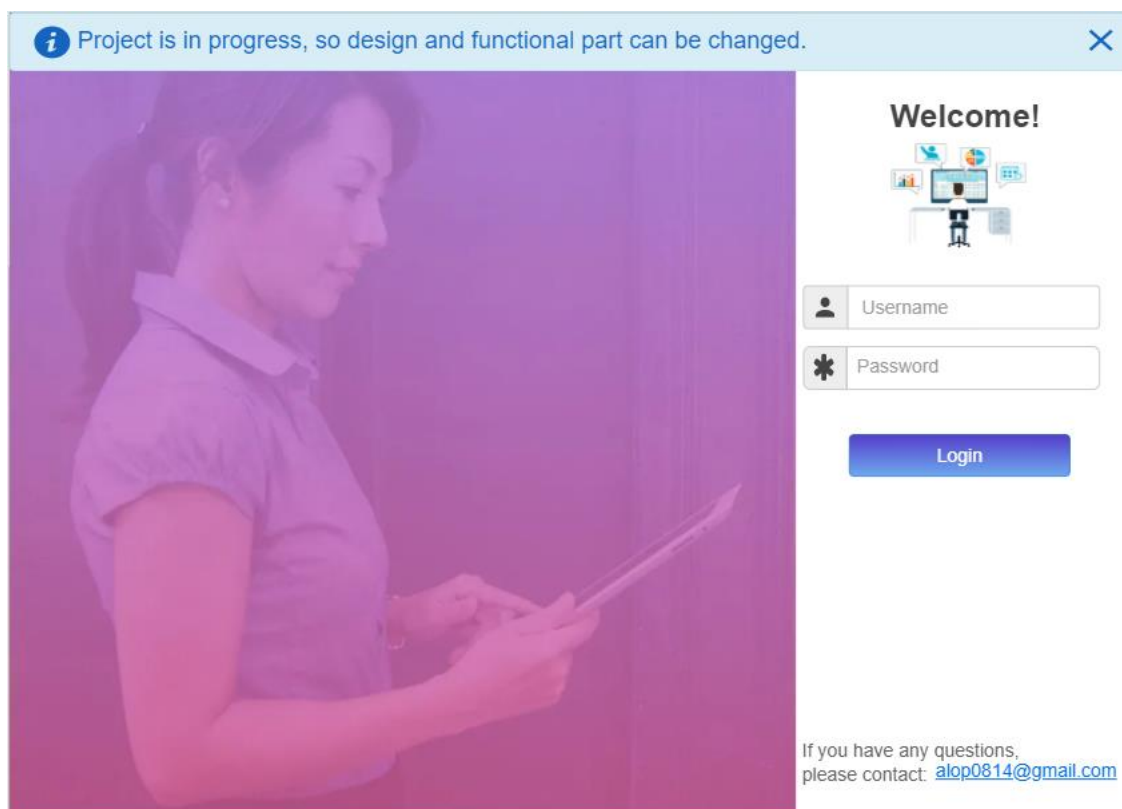
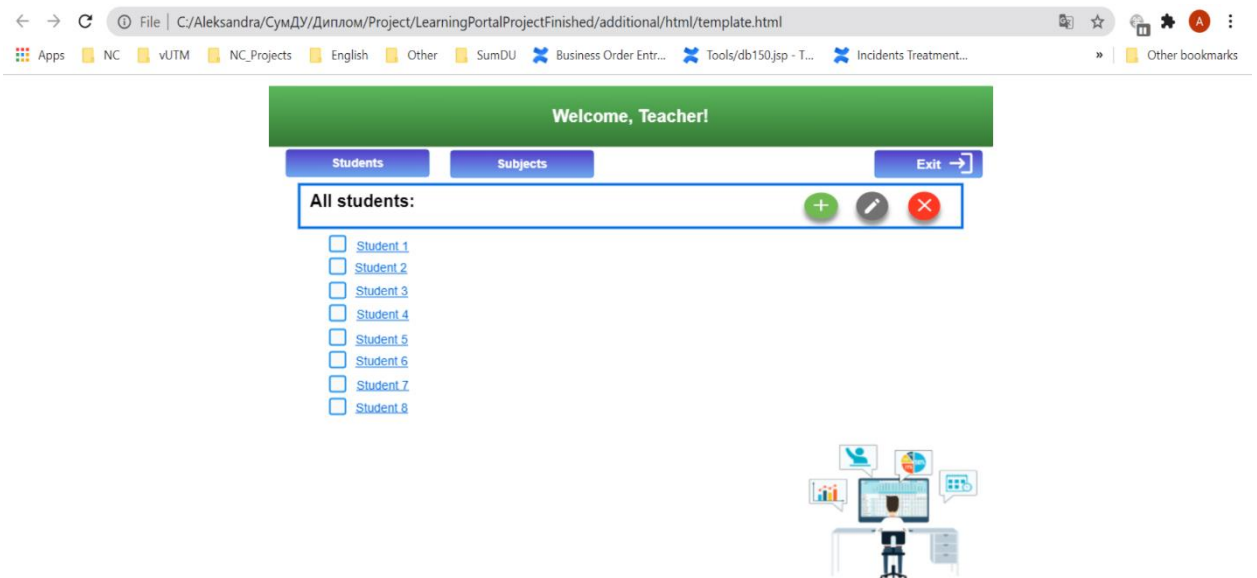


Рисунок 5.1 - Сторінка авторизації

Після того, як користувач вдало авторизувався, він може розпочати користування системою. На рисунку 5.2 зображено основний функціонал програми.





Рисунку 5.2 - Основний функціонал програми

Система надає можливість вносити корективи до інформації про студентів, які вже існують в онлайн-журналі, а також, викладач може додавати нових студентів до навчального курсу. Функціонал додавання нових студентів до курсу наведено на рисунку 5.3.

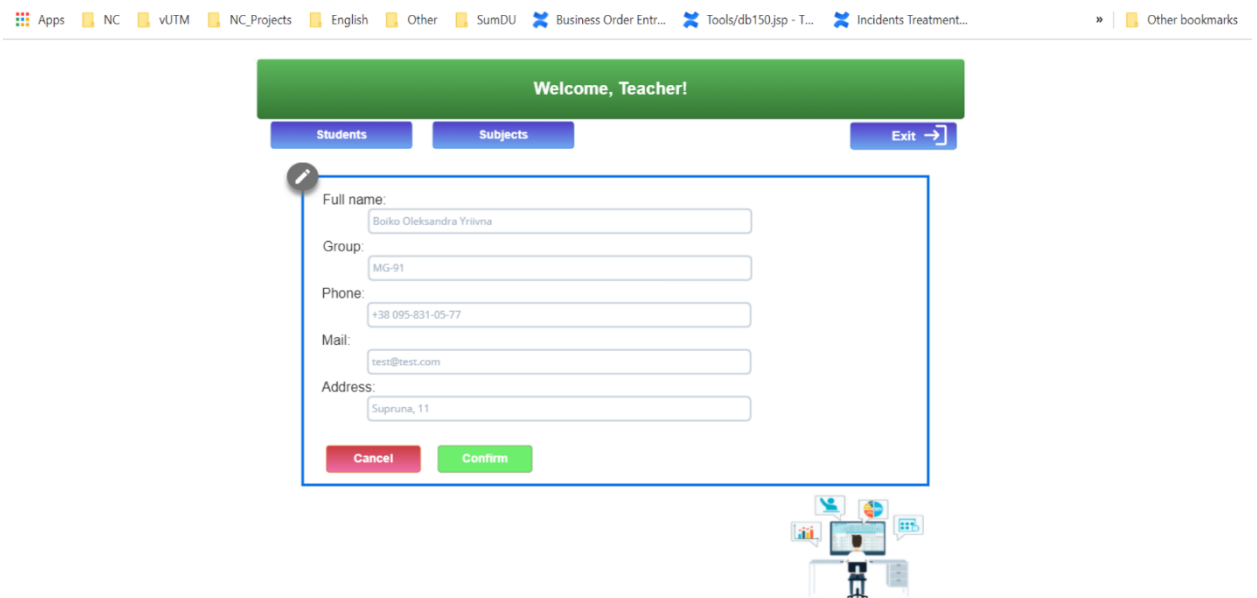


Рисунок 5.3 - Форма редагування/додавання студентів

## ВИСНОВКИ

Всі завдання, які необхідно виконати для початку використання онлайн-журналу для оцінювання успішності студентів, виконані:

- огляд існуючих систем обліку;
- огляд існуючих мов програмування;
- огляд веб-баз даних;
- огляд веб-орієнтованих фреймворків;
- розробка ER-діаграми зі сценарієм ініціалізації бази даних SQL;
- розробка схеми класів UML;
- вибір веб-сервера для розгортання програми;
- дизайн інтерфейсу користувача;
- написання та тестування коду.

Слід зазначити, що для повноцінного веб-сайту, де студенти та викладачі можуть контролювати успішність, є деякі вдосконалення, які потрібно зробити в майбутньому. Наприклад:

- використання системи контролю версій, таких як git або svn;
- особливості безпеки реалізації: Spring security;
- експорт даних;
- створення звітів;
- реалізація різних корисних фреймворків: Spring MVC, Hibernate, бібліотеки Google;

Створений додаток можна використовувати в навчальному процесі для будь-якої навчальної дисципліни та може спонукати студентів підвищити середній бал.

## СПИСОК ЛІТЕРАТУРИ

1. Додаток в Google Play – Журнал викладача [Електронний ресурс]. – Режим доступу:  
<https://play.google.com/store/apps/details?id=com.drprog.sjournal&hl=ru>
2. Журнал викладача на Android (SJournal) [Електронний ресурс]. – Режим доступу: <https://www.youtube.com/watch?v=-w-cBqfNT-I>
3. Teacherease: Gredebook Setup [Електронний ресурс]. – Режим доступу - <https://www.teacherease.com/instructors/SetupGradebook.aspx>
4. Teacherease: Gredebook Setup [Електронний ресурс]. – Режим доступу - <https://www.teacherease.com/instructors/SetupGradebook.aspx>
5. OpenGrade Setup [Електронний ресурс]. – Режим доступу - <http://www.lightandmatter.com/ogr/ogr.html>
6. Additio App | Cuaderno de notas para el profesora [Електронний ресурс]. – Режим доступу - <https://www.additioapp.com/en/>
7. Google Клас [Електронний ресурс]. – Режим доступу - <https://chrome.google.com/webstore/detail/google-classroom/>
8. Додаток в Google Play – Teacher Gradebook – Additio [Електронний ресурс]. – Режим доступу:  
<https://play.google.com/store/apps/details?id=com.additioapp.additio&hl=ru>
9. Features | Additio App [Електронний ресурс]. – Режим доступу:  
<https://www.additioapp.com/en/features/>
10. GNota in Launchpad App [Електронний ресурс]. – Режим доступу:  
<https://launchpad.net/gnota>
11. Tuttle SVC: December 2007 [Електронний ресурс]. – Режим доступу:  
<http://www.tuttlesvc.org/2007/12/>
12. Teacher Gradebook; Educational innovation project: digital gradebook. [Електронний ресурс]. – Режим доступу: <http://apps4edu.org/>
13. Додаток в Google Play – Журнал викладача [Електронний ресурс]. – Режим доступу:  
<https://play.google.com/store/apps/details?id=com.apolosoft.cuadernoprofesor&hl=ru>

14. ClassDojo [Електронний ресурс]. – Режим доступу:  
<https://www.classdojo.com/>
15. Kahoot! | Learning Games | Make Learning Awesome! ClassDojo  
[Електронний ресурс]. – Режим доступу: <https://kahoot.com/>
16. Moodle - Open-source learning platform | Moodle.org - [Електронний  
ресурс]. – Режим доступу: <https://moodle.org/?lang=ru>
17. Edmodo org - [Електронний ресурс]. – Режим доступу:  
<https://socrative.com/>
18. Home – Socrative - [Електронний ресурс]. – Режим доступу:  
<https://socrative.com/>
19. ThinkWave | Cloud Based School Management Software | Free Gradebook  
Socrative - [Електронний ресурс]. – Режим доступу:  
<https://www.thinkwave.com/>
20. Programming languages for web-development [Електронний ресурс] –2011.–  
Режим доступу: <https://www.cleverism.com/programming-languages-web-development/>– Screen title.
21. Рейтинг мов програмування 2020 [Електронний ресурс] - Режим доступу:  
<https://dou.ua/lenta/articles/language-rating-jan-2020/>
22. IBM Corporation. "IBM Information Management System (IMS) 13  
Transaction and Database Servers delivers high performance and low total cost  
of ownership". Retrieved Feb 20, 2014.
23. Wagner, Michael (2010), SQL/XML:2006 – Evaluierung der  
Standardkonformität ausgewählter Datenbanksysteme, Diplomica Verlag,  
ISBN 978-3836696098
24. ISO/IEC 9075-2:2016: Information technology -- Database languages -- SQL --  
Part 2: Foundation (SQL/Foundation)". December 2016.
25. База даних [Електронний ресурс] - <https://uk.wikipedia.org/>
26. Head First Design Pattern Eric Freeman, Elisabeth Robson, Bert Bates, Kathy  
Sierra – O'Reilly Media, 2004. – 694 с.
27. Cay S. Horstmann. Java. Core Java Volume I--Fundamentals (9th Edition)  
(Core Series), 2014. - 864 p.

28. Joshua Bloch. Java. Effective programming. Second edition, 2014. - 439 p.
29. Хабибуллин И. Ш. Java 7. - СПб.: БХВ-Петербург, 2012. - 768 с.
30. Eckel В. Philosophy of Java. Library of the professional. 4 - 2009. - 640 p.
31. Schildt Herbert. Java 8. The Complete Guide, 9th Edition, 2015. - 1376 p.

## ДОДАТОК А. ІНІЦІАЛІЗАЦІЯ БАЗИ ДАНИХ

```
BEGIN
  EXECUTE IMMEDIATE 'DROP TABLE ENROLLMENT';
EXCEPTION
  WHEN OTHERS THEN
    IF SQLCODE != -942 THEN
      RAISE;
    END IF;
END;
```

```
BEGIN
  EXECUTE IMMEDIATE 'DROP TABLE GRADES';
EXCEPTION
  WHEN OTHERS THEN
    IF SQLCODE != -942 THEN
      RAISE;
    END IF;
END;
```

```
BEGIN
  EXECUTE IMMEDIATE 'DROP TABLE COURSES';
EXCEPTION
  WHEN OTHERS THEN
    IF SQLCODE != -942 THEN
      RAISE;
    END IF;
END;
```

```
BEGIN
  EXECUTE IMMEDIATE 'DROP TABLE LOCATIONs';
EXCEPTION
  WHEN OTHERS THEN
    IF SQLCODE != -942 THEN
      RAISE;
    END IF;
END;
```

```
BEGIN
  EXECUTE IMMEDIATE 'DROP TABLE STUDENTS';
EXCEPTION
  WHEN OTHERS THEN
    IF SQLCODE != -942 THEN
      RAISE;
    END IF;
END;
```

```
BEGIN
```

```

EXECUTE IMMEDIATE 'DROP SEQUENCE STUDENTS_SEQ';
EXCEPTION
  WHEN OTHERS THEN
    IF SQLCODE != -942 THEN
      RAISE;
    END IF;
END;

BEGIN
  EXECUTE IMMEDIATE 'DROP SEQUENCE LOCATIONS_SEQ';
EXCEPTION
  WHEN OTHERS THEN
    IF SQLCODE != -942 THEN
      RAISE;
    END IF;
END;

BEGIN
  EXECUTE IMMEDIATE 'DROP SEQUENCE GRADES_SEQ';
EXCEPTION
  WHEN OTHERS THEN
    IF SQLCODE != -942 THEN
      RAISE;
    END IF;
END;

BEGIN
  EXECUTE IMMEDIATE 'DROP SEQUENCE COURSES_SEQ';
EXCEPTION
  WHEN OTHERS THEN
    IF SQLCODE != -942 THEN
      RAISE;
    END IF;
END;

CREATE table "STUDENTS" (
  "STUDENT_ID" NUMBER,
  "STUDENT_FIO" VARCHAR2(50) NOT NULL,
  "STUDENT_GROUP" VARCHAR2(10),
  "MAIL" VARCHAR2(50),
  "PHONE_NUMBER" VARCHAR2(20),
  "ADDRESS" VARCHAR2(50),
  constraint "STUDENTS_PK" primary key ("STUDENT_ID")
);

CREATE sequence "STUDENTS_SEQ";

```

```

CREATE trigger "BI_STUDENTS"
  before insert on "STUDENTS"
  for each row
begin
  if :NEW."STUDENT_ID" is null then
    select "STUDENTS_SEQ".nextval into :NEW."STUDENT_ID" from dual;
  end if;
end;

```

```

alter table "STUDENTS" add
constraint "STUDENTS_UK1"
unique ("MAIL");

```

```

alter table "STUDENTS" add
constraint "STUDENTS_UK2"
unique ("PHONE_NUMBER");

```

```

INSERT INTO STUDENTS (STUDENT_ID, STUDENT_FIO, STUDENT_GROUP,
MAIL, PHONE_NUMBER, ADDRESS) VALUES (1, 'Student 1', 'SU-51',
'student1@gmail.com','099-293-43-94',' M.Lushpy 22');
INSERT INTO STUDENTS (STUDENT_ID, STUDENT_FIO, STUDENT_GROUP,
MAIL, PHONE_NUMBER, ADDRESS) VALUES (2, 'Student 2', 'SU-51',
'student2@gmail.com','066-545-43-94',' M.Lushpy 22');
INSERT INTO STUDENTS (STUDENT_ID, STUDENT_FIO, STUDENT_GROUP,
MAIL, PHONE_NUMBER, ADDRESS) VALUES (3, 'Student 3', 'ES-51',
'student3@gmail.com','099-293-46-94',' M.Lushpy 22');
INSERT INTO STUDENTS (STUDENT_ID, STUDENT_FIO, STUDENT_GROUP,
MAIL, PHONE_NUMBER, ADDRESS) VALUES (4, 'Student 4', 'ES-51',
'student4@gmail.com','099-293-43-34',' M.Lushpy 22');
INSERT INTO STUDENTS (STUDENT_ID, STUDENT_FIO, STUDENT_GROUP,
MAIL, PHONE_NUMBER, ADDRESS) VALUES (5, 'Student 5', 'PM-51',
'student5@gmail.com','066-935-43-94',' M.Lushpy 22');
INSERT INTO STUDENTS (STUDENT_ID, STUDENT_FIO, STUDENT_GROUP,
MAIL, PHONE_NUMBER, ADDRESS) VALUES (6, 'Student 6', 'PM-51',
'student6@gmail.com','099-293-53-94',' M.Lushpy 22');
INSERT INTO STUDENTS (STUDENT_ID, STUDENT_FIO, STUDENT_GROUP,
MAIL, PHONE_NUMBER, ADDRESS) VALUES (7, 'Student 7', 'XM-51',
'student7@gmail.com','099-293-47-34',' M.Lushpy 22');
INSERT INTO STUDENTS (STUDENT_ID, STUDENT_FIO, STUDENT_GROUP,
MAIL, PHONE_NUMBER, ADDRESS) VALUES (8, 'Student 8', 'XM-51',
'student8@gmail.com','066-145-43-94',' M.Lushpy 22');
INSERT INTO STUDENTS (STUDENT_ID, STUDENT_FIO, STUDENT_GROUP,
MAIL, PHONE_NUMBER, ADDRESS) VALUES (9, 'Student 9', 'RG-51',
'student9@gmail.com','099-293-53-91',' M.Lushpy 22');

```



```

INSERT INTO STUDENTS (STUDENT_ID, STUDENT_FIO, STUDENT_GROUP,
MAIL, PHONE_NUMBER, ADDRESS) VALUES (10, 'Student 10', 'RG-51',
'student10@gmail.com','066-345-43-94',' M.Lushpy 22');
INSERT INTO STUDENTS (STUDENT_ID, STUDENT_FIO, STUDENT_GROUP,
MAIL, PHONE_NUMBER, ADDRESS) VALUES (11, 'Student 11', 'MU-51',
'student11@gmail.com','066-941-43-94',' M.Lushpy 22');
INSERT INTO STUDENTS (STUDENT_ID, STUDENT_FIO, STUDENT_GROUP,
MAIL, PHONE_NUMBER, ADDRESS) VALUES (12, 'Student 12', 'MU-51',
'student12@gmail.com','099-293-65-94',' M.Lushpy 22');
INSERT INTO STUDENTS (STUDENT_ID, STUDENT_FIO, STUDENT_GROUP,
MAIL, PHONE_NUMBER, ADDRESS) VALUES (13, 'Student 13', 'DA-51',
'student13@gmail.com','099-293-76-94',' M.Lushpy 22');
INSERT INTO STUDENTS (STUDENT_ID, STUDENT_FIO, STUDENT_GROUP,
MAIL, PHONE_NUMBER, ADDRESS) VALUES (14, 'Student 14', 'DA-51',
'student14@gmail.com','099-345-76-94',' M.Lushpy 22');
INSERT INTO STUDENTS (STUDENT_ID, STUDENT_FIO, STUDENT_GROUP,
MAIL, PHONE_NUMBER, ADDRESS) VALUES (15, 'Student 15', 'KI-51',
'student15@gmail.com','099-293-43-45',' M.Lushpy 22');
INSERT INTO STUDENTS (STUDENT_ID, STUDENT_FIO, STUDENT_GROUP,
MAIL, PHONE_NUMBER, ADDRESS) VALUES (16, 'Student 16', 'KI-51',
'student16@gmail.com','066-942-43-34',' M.Lushpy 22');
INSERT INTO STUDENTS (STUDENT_ID, STUDENT_FIO, STUDENT_GROUP,
MAIL, PHONE_NUMBER, ADDRESS) VALUES (17, 'Student 17', 'OM-51',
'student17@gmail.com','099-293-43-76',' M.Lushpy 22');
INSERT INTO STUDENTS (STUDENT_ID, STUDENT_FIO, STUDENT_GROUP,
MAIL, PHONE_NUMBER, ADDRESS) VALUES (18, 'Student 18', 'OM-51',
'student18@gmail.com','099-293-43-43',' M.Lushpy 22');
INSERT INTO STUDENTS (STUDENT_ID, STUDENT_FIO, STUDENT_GROUP,
MAIL, PHONE_NUMBER, ADDRESS) VALUES (19, 'Student 19', 'NO-51',
'student19@gmail.com','099-293-44-94',' M.Lushpy 22');
INSERT INTO STUDENTS (STUDENT_ID, STUDENT_FIO, STUDENT_GROUP,
MAIL, PHONE_NUMBER, ADDRESS) VALUES (20, 'Student 20', 'NO-51',
'student20@gmail.com','099-293-43-56',' M.Lushpy 22');
INSERT INTO STUDENTS (STUDENT_ID, STUDENT_FIO, STUDENT_GROUP,
MAIL, PHONE_NUMBER, ADDRESS) VALUES (21, 'Student 21', 'LJ-51',
'student21@gmail.com','099-293-47-76',' M.Lushpy 22');
INSERT INTO STUDENTS (STUDENT_ID, STUDENT_FIO, STUDENT_GROUP,
MAIL, PHONE_NUMBER, ADDRESS) VALUES (22, 'Student 22', 'LJ-51',
'student22@gmail.com','099-293-43-57',' M.Lushpy 22');

```

```

CREATE table "LOCATIONS" (
  "LOCATION_ID" NUMBER,
  "NAME" VARCHAR2(50) NOT NULL,
  "PARENT_ID" NUMBER,
  "IS_LAST_LEVEL" VARCHAR2(5) NOT NULL,
  "DESCRIPTION" VARCHAR2(1000),

```

```

    constraint "LOCATIONS_PK" primary key ("LOCATION_ID")
);

CREATE sequence "LOCATIONS_SEQ";

CREATE trigger "BI_LOCATIONS"
  before insert on "LOCATIONS"
  for each row
begin
  if :NEW."LOCATION_ID" is null then
    select "LOCATIONS_SEQ".nextval into :NEW."LOCATION_ID" from dual;
  end if;
end;

INSERT INTO LOCATIONS (LOCATION_ID, NAME, PARENT_ID, IS_LAST_LEVEL,
DESCRIPTION) VALUES (1, 'UKRAINE', NULL, 'false', 'Best country in the world');
INSERT INTO LOCATIONS (LOCATION_ID, NAME, PARENT_ID, IS_LAST_LEVEL,
DESCRIPTION) VALUES (2, 'Sumy', 1, 'false', 'Best city in the world');
INSERT INTO LOCATIONS (LOCATION_ID, NAME, PARENT_ID, IS_LAST_LEVEL,
DESCRIPTION) VALUES (3, 'Sumy State University', 2, 'false', 'Best university in the
world');
INSERT INTO LOCATIONS (LOCATION_ID, NAME, PARENT_ID, IS_LAST_LEVEL,
DESCRIPTION) VALUES (4, 'Computer Science Department', 3, 'true', 'Best department in
the world');
INSERT INTO LOCATIONS (LOCATION_ID, NAME, PARENT_ID, IS_LAST_LEVEL,
DESCRIPTION) VALUES (5, 'Foreign Language Department', 3, 'true', 'Best foreign
language department in the world');

CREATE table "COURSES" (
  "COURSE_ID" NUMBER,
  "LOCATION_ID" NUMBER,
  "NAME" VARCHAR2(50) NOT NULL,
  "DESCRIPTION" VARCHAR2(1000),
  "TEACHER" VARCHAR2(50),
  constraint "COURSES_PK" primary key ("COURSE_ID")
);

CREATE sequence "COURSES_SEQ";

CREATE trigger "BI_COURSES"
  before insert on "COURSES"
  for each row
begin
  if :NEW."COURSE_ID" is null then
    select "COURSES_SEQ".nextval into :NEW."COURSE_ID" from dual;
  end if;
end;

```

```

end;

ALTER TABLE "COURSES" ADD CONSTRAINT "COURSES_FK"
FOREIGN KEY ("LOCATION_ID")
REFERENCES "LOCATIONS" ("LOCATION_ID")
ON DELETE CASCADE;

INSERT INTO COURSES (COURSE_ID, LOCATION_ID, NAME, DESCRIPTION,
TEACHER) VALUES (1, 4, 'Java', 'Best course in the world', 'Berest O.B.');
```

```

CREATE table "GRADES" (
  "GRADE_ID"      NUMBER,
  "STUDENT_ID"    NUMBER NOT NULL,
  "COURSE_ID"     NUMBER NOT NULL,
  "DATE_OF_EVALUATION" DATE NOT NULL,
  "CREDITS"       FLOAT NOT NULL,
  constraint "GRADES_PK" primary key ("GRADE_ID")
);
```

```

CREATE sequence "GRADES_SEQ";
```

```

CREATE trigger "BI_GRADES"
before insert on "GRADES"
for each row
begin
  if :NEW."GRADE_ID" is null then
    select "GRADES_SEQ".nextval into :NEW."GRADE_ID" from dual;
  end if;
end;
```

```

ALTER TABLE "GRADES" ADD CONSTRAINT "GRADES_FK2"
FOREIGN KEY ("STUDENT_ID")
REFERENCES "STUDENTS" ("STUDENT_ID")
ON DELETE CASCADE;
```

```

ALTER TABLE "GRADES" ADD CONSTRAINT "GRADES_FK3"
FOREIGN KEY ("COURSE_ID")
REFERENCES "COURSES" ("COURSE_ID")
ON DELETE CASCADE;
```

```

INSERT INTO GRADES ( GRADE_ID, STUDENT_ID, COURSE_ID,
DATE_OF_EVALUATION, CREDITS) VALUES (1, 1, 1, '12-17-2015', 5.0);
```

```

CREATE table "ENROLLMENT" (
  "STUDENT_ID" NUMBER,
  "COURSE_ID"  NUMBER,
```

```
"DATEENROLLED" DATE,  
"FINALGRADE" FLOAT  
);  
  
alter table "ENROLLMENT" add constraint "ENROLLMENT_PK" primary key  
("STUDENT_ID","COURSE_ID");  
  
ALTER TABLE "ENROLLMENT" ADD CONSTRAINT "ENROLLMENT_FK"  
FOREIGN KEY ("STUDENT_ID")  
REFERENCES "STUDENTS" ("STUDENT_ID");  
  
ALTER TABLE "ENROLLMENT" ADD CONSTRAINT "ENROLLMENT_FK2"  
FOREIGN KEY ("COURSE_ID")  
REFERENCES "COURSES" ("COURSE_ID");  
  
INSERT INTO ENROLLMENT (STUDENT_ID, COURSE_ID, DATEENROLLED,  
FINALGRADE) VALUES (1, 1, '01-09-2015', null);
```

## ДОДАТОК Б. ЛІСТИНГ

Class DispatcherServlet – core of application.

```
public class DispatcherServlet extends HttpServlet {
    static final Logger logger = Logger.getLogger(DispatcherServlet.class);
    private List<Processor> listOfAllProcessors = new ArrayList<Processor>();
    private List<String> namesOfAllProcessors = new ArrayList<String>();
    static {
        logger.info("Your session begin at: " + new Date());
    }
    @Override
    public void init() throws ServletException {
        try {
            extractProcessorNamesFromXMLFile("/WEB-INF/resources/processors.xml");
            logger.info(namesOfAllProcessors.size() + " processors were initialized.");
        } catch (IOException e) {
            logger.error("Threw a IOException in DispatcherServlet class::" + e.getMessage(), e);
            throw new ServletException(e);
        }
        for (String name : namesOfAllProcessors) {
            try {
                listOfAllProcessors.add((Processor) Class.forName(name).newInstance());
            } catch (InstantiationException e) {
                logger.error("Threw a InstantiationException in DispatcherServlet class::" +
                    e.getMessage(), e);
                throw new ServletException(e);
            } catch (IllegalAccessException e) {
                logger.error("Threw a IllegalAccessException in DispatcherServlet class::" +
                    e.getMessage(), e);
                throw new ServletException(e);
            } catch (ClassNotFoundException e) {
                logger.error("Threw a ClassNotFoundException in DispatcherServlet class::" +
                    e.getMessage(), e);
                throw new ServletException(e);
            }
        }
    }
}
```

```

    }
}
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
IOException {
    this.process(request, response);
}
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
IOException {
    this.process(request, response);
}
private void process(HttpServletRequest request, HttpServletResponse response) throws
IOException {

    logger.info("Current value of action parameter: " + request.getParameter("action"));
    for (Processor processor : listOfAllProcessors) {
        try {
            if(processor.canProcess(request.getParameter("action"))){
                ProcessorResult result = processor.process(request);
                if(result.getIncludedPage() != null) {
                    request.getSession().setAttribute("includedJSPName", result.getIncludedPage());
                }
                if (result.isForward()) {
                    try {
                        RequestDispatcher rd = request.getRequestDispatcher(result.getUrl());
                        rd.forward(request, response);
                    } catch (ServletException e) {
                        logger.error("Throw a ServletException in DispatcherServlet class: " +
e.getMessage(), e);
                    }
                }
            }
            else {
                response.sendRedirect("DispatcherServlet" + result.getUrl());
            }
        }
    }
}

```

```

        }
        break;
    }
} catch (DataAccessException e) {
    logger.error("Threw a DataAccessException in DispatcherServlet class:." +
e.getMessage(), e);
}
}
}
private void extractProcessorNamesFromXMLFile(String path) throws IOException {

    String fullPath = getServletContext().getResource(path).getPath();
    File xmlFile = new File(fullPath);
    DocumentBuilderFactory docFactory = DocumentBuilderFactory.newInstance();
    DocumentBuilder docBuilder = null;
    try {
        docBuilder = docFactory.newDocumentBuilder();
    } catch (ParserConfigurationException e) {
        logger.error("Threw a ParserConfigurationException in DispatcherServlet class:." +
e.getMessage(), e);
        throw new IOException(e);
    }
    try {
        Document document = docBuilder.parse(xmlFile);
        document.getDocumentElement().normalize();
        NodeList list = document.getElementsByTagName("processor");
        String className;
        for (int i = 0; i < list.getLength(); i++) {
            Node node = list.item(i);
            if(node instanceof Element) {
                Element element = (Element)list.item(i);
                className =
element.getElementsByTagName("processorName").item(0).getTextContent();

```

```

        namesOfAllProcessors.add(className);
    }
}
} catch (SAXException e) {
    logger.error("Threw a SAXException in DispatcherServlet class::" + e.getMessage(), e);
    throw new IOException(e);
}
}
}
}

```

Class `PaginationController` to implement pagination for big result set.

```

public class PaginationController {
    private int totalItems;
    private int itemsPerPage;
    private int currentPageNumber;
    public PaginationController(int totalItems, int itemsPerPage, int currentPageNumber) {
        this.totalItems = totalItems;
        this.itemsPerPage = itemsPerPage;
        this.currentPageNumber = currentPageNumber;
    }
    public void setTotalItems(int totalItems) {
        this.totalItems = totalItems;
    }
    public int getCurrentPageNumber() {
        return currentPageNumber;
    }
    public void setCurrentPageNumber(int currentPageNumber) {
        this.currentPageNumber = currentPageNumber;
    }
    public int getTotalItems() {
        return totalItems;
    }
    public int getItemsPerPage() {

```



```

    return itemsPerPage;
}
public void setItemsPerPage(int itemsPerPage) {
    this.itemsPerPage = itemsPerPage;
}
public String makePagingLinks(String pageLocation, String addToRequest) {
    String ret = "";
    if (totalItems <= itemsPerPage)
        return ret;
    int totalPages = (totalItems / itemsPerPage);
    if (totalItems % itemsPerPage != 0)
        totalPages++;
    if(currentPageNumber > totalPages)
        currentPageNumber = 1;
    if (totalItems <= currentPageNumber * itemsPerPage)
        currentPageNumber = totalPages;
    int start = currentPageNumber - 3;
    if (start <= 0)
        start = 1;
    int end = currentPageNumber + 3;
    if (end >= totalPages)
        end = totalPages;
    if (start > 1){
        ret += "<a href=\"" + pageLocation + "&page=1"
            + addToRequest + "\">1</a> ";
    }
    if (start > 2)
        ret += "... ";
    for (int i = start; i <= end; i++) {
        if (i == currentPageNumber){
            ret += "<span style=\"font-weight: bold;\" +
                \"color: red;\">" + (i) + "</span>";
        }else{
            ret += "<a href=\"" + pageLocation + "&page="
                + (i) + addToRequest + "\">" + (i) + "</a>";
        }
    }
    if (i < totalPages)
        ret += " ";
}

```

```

if (end + 1 < totalPages)
    ret += "... ";
if (end < totalPages) {
    if (totalPages == currentPageNumber){
        ret += "<strong>" + totalPages + "</strong>";
    }else{
        ret += "<a href=\"" + pageLocation + "&page=\""
            + totalPages + addToRequest + "\">"
            + totalPages + "</a>";
    }
}
ret += " ";
return ret;
}
}

```

Class ModelDataAccess – connection to Data Base.

```

public interface ModelDataAccess {
    List<Student> getAllStudents() throws DataAccessException;
    List<Student> getAllStudentsInCourse(int courseId) throws DataAccessException;
    List<Student> getAllStudentsOutOfCourse(int courseId) throws DataAccessException;
    List<Location> getAllLocationsByParentId(int locationId) throws DataAccessException;
    List<Grade> getAllGradesByCourseAndStudent(int courseId, int studentId) throws
DataAccessException;
    void addStudent(Student student) throws DataAccessException;
    void addLocation(Location location) throws DataAccessException;
    void addCourse(Course course) throws DataAccessException;
    void addGrade(Grade grade) throws DataAccessException;
    void removeStudent(int studentId) throws DataAccessException;
    void removeLocation(int locationId) throws DataAccessException;
    void removeCourse(int courseId) throws DataAccessException;
    void removeGrade(int gradeId) throws DataAccessException;
    void updateStudent(Student student) throws DataAccessException;
}

```

```
void updateLocation(Location location) throws DataAccessException;
void updateCourse(Course course) throws DataAccessException;
void updateGrade(Grade grade) throws DataAccessException;
void enrollStudent(int studentId, int courseId) throws DataAccessException;
void removeEnrollment(int studentId, int courseId) throws DataAccessException;
Student getStudentById(int studentId) throws DataAccessException;
Course getCourseById(int courseId) throws DataAccessException;
Location getLocationById(int locationId) throws DataAccessException;
Grade getGradeById(int gradeId) throws DataAccessException;
List<Course> getCoursesByStudentId(int studentId) throws DataAccessException;
List<Course> getCoursesByLocationId(int locationId) throws DataAccessException;
Map<Integer,String> getLocationHierarchy(int locationId) throws DataAccessException;
Map<Integer,String> getAllLocations() throws DataAccessException;
int getTotalCountOfStudents() throws DataAccessException;
List<Student> getAllStudentsByPage(int startIndex, int range) throws DataAccessException;
}
```