

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій**

Курсова робота

з дисципліни «Програмування»

**на тему: «Інформаційна система для роботи із
групами студентів »**

Виконала:
студентка 1 курсу, групи ІА-34
Загорулько Єлизавета Вікторівна

(підпис)

Керівник:
асистент кафедри ІСТ
Тюляков Дмитро Ігорович

(підпис)

Київ – 2024 року

	ЗМІСТ	
ВСТУП		3
1 ВИМОГИ ДО СИСТЕМИ		4
1.1 Функціональні вимоги до системи		4
1.2 Нефункціональні вимоги до системи		4
2 СЦЕНАРІЇ ВИКОРИСТАННЯ СИСТЕМИ		5
2.1 Діаграма прецедентів		6
2.2 Опис сценаріїв використання системи		7
3 АРХІТЕКТУРА СИСТЕМИ		24
4 РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ		26
4.1 Загальна структура проекту		27
4.2 Компоненти рівня доступу до даних		27
4.3 Компоненти рівня бізнес-логіки		29
4.4 Компоненти рівня інтерфейсу користувача		30
ВИСНОВКИ		31
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ		33
ДОДАТОК А Лістінг програми		36
		38

ВСТУП

Все більше і більше накопичується даних про студентів, викладачів, успішності, кожного року інформація оновлюється і звичайно потребує ефективного рішення для пошуку та редагування. Наприклад, додати студента до групи не є важкою процедурою, як змінити всі відповідні дані та головне зберегти їх у правильному місці і ефективно ними користуватись. Виникають питання по особистим заявкам і тут перегортати усіх студентів у журналі чи навіть в таблиці, не буде зручним варіантом, використовуючи сторонні продукти не гарантується безпека збереження інформації, а особливо особистої. Так наприклад, ексель таблиці є дуже зручними для простого користувача, але не коли університет налічує майже 25 тисяч студентів *1.

Саме розробка системи, що має на меті полегшити доступ до інформації, редагування, перегляд студентів та груп студентів. Допоможе збільшити ефективність роботи деканату покладаючись на інтенсивний розвиток та автоматизувати рутинні задачі.

Які задачі вирішує система:

1. Створення функціоналу, що дозволить деканату редагування, видалення та додавання студентів у групи.
2. Розробка системи управління конкретними групами, яка дозволяє створювати нові, редагувати існуючі та призначати студентів до певних груп.
3. Доступ студентів до системи для перегляду інформації про свої групи, пошук за прізвищем та номером групи.
4. Створення зрозумілого інтерфейсу як для співробітника так і для студентів, з подальшою можливістю під'єднання додаткових джерел.

1 ВИМОГИ ДО СИСТЕМИ

1.1 Функціональні вимоги до системи

Система має відповідати наступним функціональним вимогам:

1. Зареєстрований користувач, як співробітник, має мати можливість редагувати, видаляти та додавати студентів у групи;
2. Зареєстрований користувач, як співробітник, має мати можливість редагувати групи цілком;
3. Зареєстрований користувач, як студент, має мати можливість переглядати групи за назвою групи та за прізвищем студента

1.2 Нефункціональні вимоги до системи

Система має відповідати наступним нефункціональним вимогам:

1. Система повинна мати відкриту архітектуру;
2. Система повинна мати веб-інтерфейс;
3. Інтерфейс користувача має бути зручним та інтуїтивно-зрозумілим;
4. Система повинна бути кросплатформною.

2 СЦЕНАРІЇ ВИКОРИСТАННЯ СИСТЕМИ

2.1 Діаграма прецедентів

Діаграма прецедентів системи представлена на рис. 2.1.

Акторами є користувачі системи: зареєстрований (співробітник) та зареєстрований(студент).

Співробітнику, доступний функціонал редагування груп. Студенту, недоступний функціонал збереження своєї групи на головній сторінці та пошук. Зареєстрований користувач має можливість пошуку інформації про групи та студентів, які їх налічують.(*2)



Рисунок 2.1 – Діаграма прецедентів

2.2 Опис сценаріїв використання системи

Детальні описи сценаріїв використання наведено у таблицях 2.1 – 2.17.

Таблиця 2.1 – Сценарій використання «Пошук по назві групи, по студенту»

Назва	Пошук по назві групи, по студенту
ID	1
Опис	Користувач використовує поле пошуку для введення назви групи або прізвища студента
Актори	Користувач
Вигоди компанії	Спрощує необхідність знайти дані про групи або студента
Частота користування	Постійно
Тригери	Користувач вводить пошуковий запит у полі для пошуку
Передумови	Пошукове поле доступне у будь-якому вікні
Постумови	Користувач потрапляє на вікно з результатами пошуку
Основний розвиток	Користувач вводить запит у пошукову строку, натискає на кнопку пошуку чи Enter
Альтернативні розвитку	Вибиває помилку перевірити коректність введення Вибиває, що не існує такої групи або студента
Виняткові ситуації	-

Таблиця 2.3 – Сценарій використання «Додавання або видалення студента у групи співробітником»

Назва	Додавання або видалення студента у групи співробітником
ID	3
Опис	Співробітник має можливість видалити або додати студента
Актори	Зареєстрований користувач, як співробітник
Вигоди компанії	Швидкий доступ до видалення та додавання(редагування) групи
Частота користування	Переважно під кінець семестрів
Тригери	Користувач шукає певну групу(або студента) і виконує редагування
Передумови	Доступна вкладка пошуку
Постумови	Користувач отримує список введених груп
Основний розвиток	Користувач знаходить потрібну групу або студента і редагує
Альтернативні розвитку	-
Виняткові ситуації	-

3 АРХІТЕКТУРА СИСТЕМИ

Загальна архітектура системи наведена на рис. 3.1.

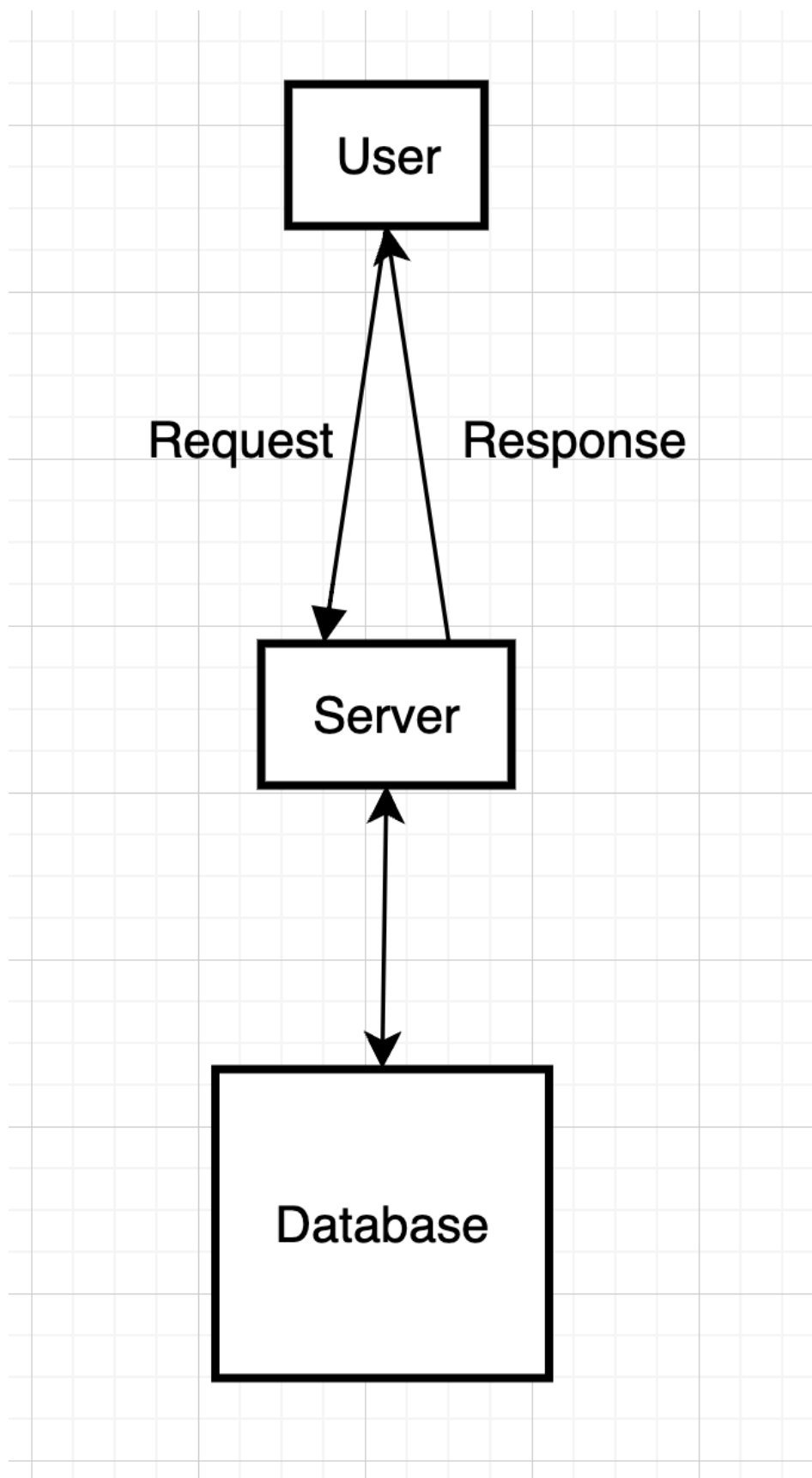


Рисунок 3.1 – Загальна архітектура системи

Система складається з наступних елементів:

- Інтерфейс користувача;
- серверна частина;
- база даних.

Графічний інтерфейс є ключовим для взаємодії з користувачами. Коли HTTP запит відправляється до сервера, він там обробляється, після чого сервер надсилає відповідь. Основна бізнес-логіка виконується на сервері. Дані, отримані через графічний інтерфейс, проходять валідацію та конвертацію на сервері. Сервер також ініціює запити до бази даних, обробляє отримані відповіді та передає їх назад до графічного інтерфейсу. База даних відповідає за зберігання інформації, яка була оброблена сервером, і видає ці дані за запитом.

4 РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ

4.1 Загальна структура проекту на рисунках 4.1,4.2

Рисунок 4.1

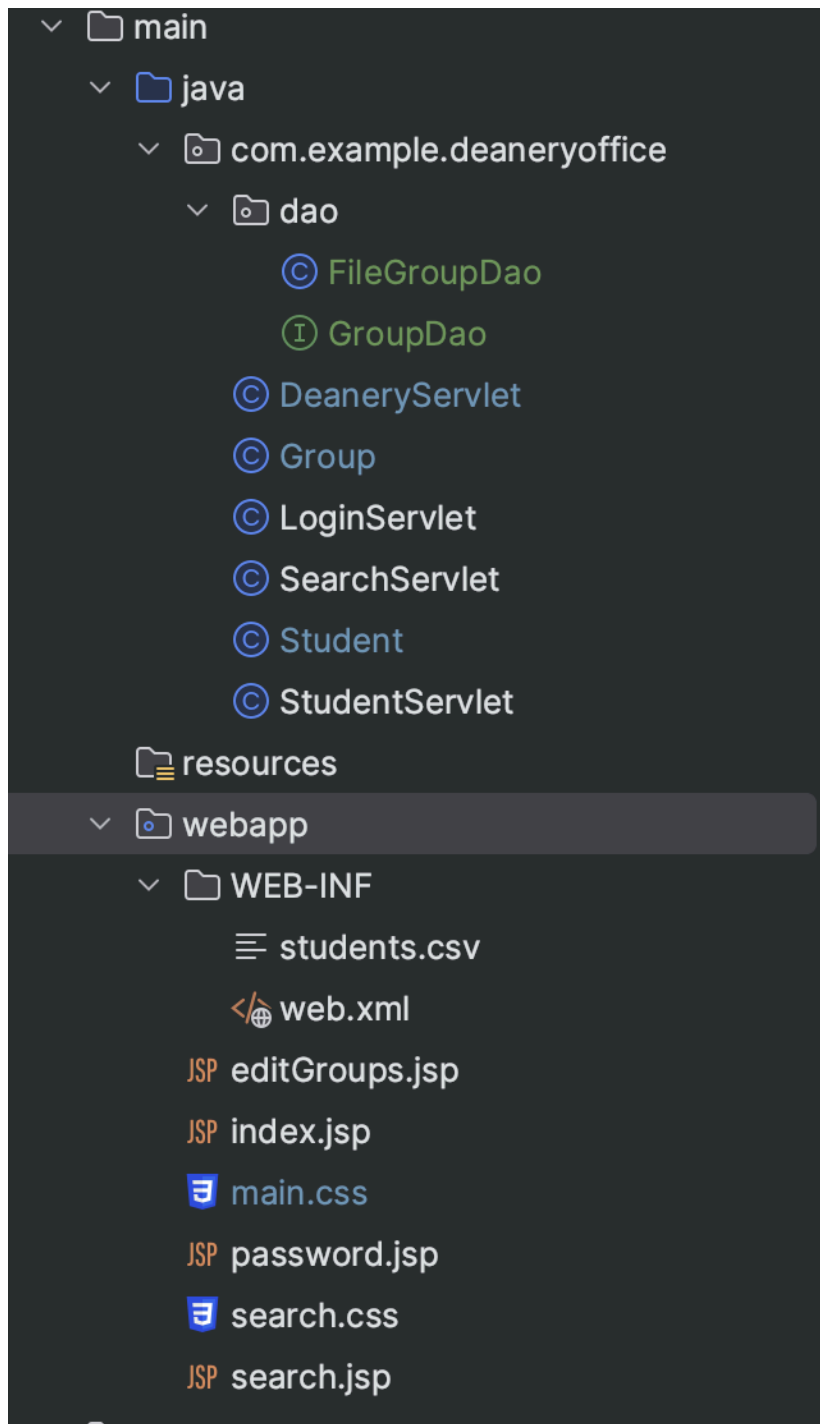


Рисунок 4.2



Проект складається з веб-ресурсів, бібліотек, та вихідного коду, який в свою чергу можна поділити на компоненти рівня доступу до даних, компоненти бізнес-логіки та веб-компоненти.

4.2 Компоненти рівня доступу до даних

Основні сутності та інтерфейси рівня доступу до даних наведені на рис. 4.3

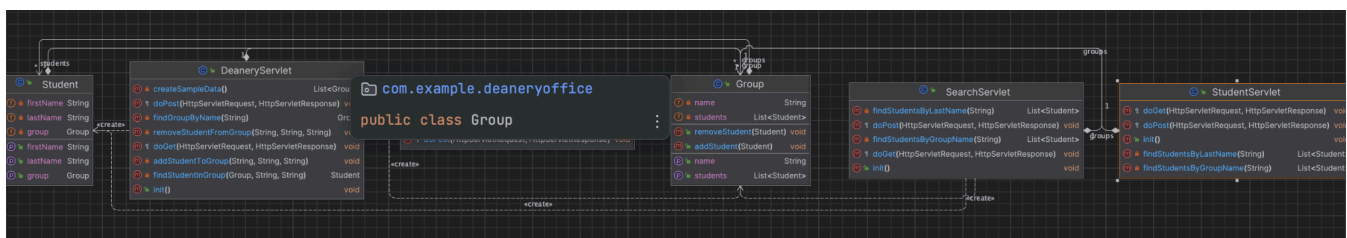


Рисунок 4.3

4.3 Компоненти рівня бізнес-логіки

Сервер для деканату призначений для управління групами студентів. Він забезпечує функціонал для додавання, видалення та редагування груп і студентів.

- Сервер для деканату містить кілька сервлетів, кожен з яких виконує специфічні завдання для управління групами студентів.
 - GroupServlet** - додає нову групу з вказаною назвою, додає студента до зазначеної групи, видаляє студента зі зазначеної групи.
 - SearchServlet** - шукає студентів за заданим прізвищем, шукає студентів, що належать до заданої групи.
- Для взаємодії користувача з сервером використовуються JSP-сторінки

- a. **manageGroups.jsp**: Інтерфейс для управління групами та студентами. Дозволяє додавати нові групи, додавати та видаляти студентів.
- b. **search.jsp**: Інтерфейс для пошуку студентів за прізвищем або назвою групи.

4.4 Компоненти рівня інтерфейсу користувача

Компоненти рівня інтерфейсу користувача

1. **index.jsp** - Головна сторінка, яка містить посилання на сторінки управління групами та пошуку студентів.
2. **manageGroups.jsp** - Сторінка для управління групами, яка дозволяє додавати та видаляти групи і студентів.
3. **search.jsp** - Сторінка для пошуку студентів за прізвищем або назвою групи.

CSS файли

main.css відповідає за стилізацію основних веб-сторінок, максимальний мінімалізм та зручність, також можливість повернутись на сторінку входу

search.css відповідає за сторінку із пошуком

ВИСНОВКИ

У цій курсовій роботі було пророблено аналіз потреб користувача та співробітника.

Першим кроком було сформулювати функціональні та нефункціональні вимоги до системи, що визначило очікувану поведінку системи. Наступним, обрано технології на яких буде написана система. Java була обрана в якості мови програмування, середа розробки – IntelliJ IDEA за її можливості та потужність. Для зберігання даних була використана структура dao (*3). Далі використовували сервлети для з'єднання серверу та jsp сторінки, на сервлетах були написані методи doPost doGet для отримання від користувача аргументів, які будуть оброблятися іншими сервлетами. Використовували jsp сторінки для скелету самої сторінки, усі блоки для вводу інформації, переходу на інші сторінки і для оформлення. Для оформлення використовували css файли(*4), щоб зробити більш функціональним та зручним для розуміння сайт.

Спочатку створено систему входу, для уникнення зайвих редагування, було поставлено загальний пароль для усіх співробітників, після введення правильного паролю, співробітник отримує доступ до всіх впринципі існуючих груп, можливість створити нову групу, додати до неї чи іншої групи студента, також видалити студента із будь-якої групи.

Студент входить на сторінку без будь-яких обмежень, просто обираючи свою роль як студент, після входу студент може переглянути студентів за назвою групи, або переглянути до яких груп належить студент знайшовши його за прізвищем.

У висновку було зроблено інформаційну систему для додавання, редагування та видалення груп, тобто деканат. Система має базовий набір функцій, також простий інтерфейс для зручності та швидкості доступу, що буде зручним для щоденного використання.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського". Структура університету. [Електронний ресурс] - Режим доступу до ресурсу: kpi.ua
2. draw.io - для побудови діаграм
3. Для підключення dao -
<https://javarush.com/ua/quests/lectures/ua.questspring.level03.lecture06>
4. Для більшого занурення до css -
https://developer.mozilla.org/ru/docs/Learn/CSS/First_steps/How_CSS_is_structured
5. Лекції Букасова Максима Михайловича
6. Посилання на гітхаб репозиторій
<https://github.com/YelyzavetaZahorulko/DeaneryOffice.git>