КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет комп'ютерних наук та кібернетики
Кафедра інтелектуальних програмних систем

Курсова робота

за спеціальністю 121 Інженерія програмного забезпечення
на тему:

СТВОРЕННЯ ВЕБ-ДОДАТКУ ДЛЯ ЗАБЕЗПЕЧЕННЯ ДИСТАНЦІЙНОГО НАВЧАННЯ

Виконав студент 3-го курсу
Микола ХЛОПИК


(підпис)

Науковий керівник:
асистент
Костянтин ЖЕРЕБ

(підпис)

Засвідчую, що в цій курсовій роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент


(підпис)

Київ – 2023

РЕФЕРАТ

Обсяг роботи 26 сторінок, 15 ілюстрацій, 8 джерел посилань.

HIBERNATE, JAVA, POSTGRESQL, SPRING, SQL, ВЕБ-ДОДАТОК, ПАТЕРН.

Об’єктом роботи є процес розробки веб додатків з використанням мови програмування Java і допоміжних фреймворків. Предметом роботи є веб-додаток для проведення онлайн навчання.

Мета роботи полягає в розробці веб-додатку для навчання на основі Spring Framework. Цей додаток має на меті забезпечити зручну та ефективну платформу для навчальної взаємодії між викладачами та студентами.

Методи розроблення: ООП; патерн: DI, MVC; принципи: SOLID, YAGNI, KISS та DRY. Інструменти розроблення: IntelliJ Idea безкоштовне, інтегроване середовище розробки для програмування мовами Java, JavaScript, Python і багато інших, мови програмування Java, JavaScript, мови для створення і стилізації веб-сторінок HTML і CSS, фреймворки: Hibernate, Thymeleaf, Spring (з якого були використані такі частини як :Spring Data JPA, Spring MVC, Spring Boot, Spring Security), Zoom API.

Результати роботи: реалізований веб-додаток з назвою «Tutoring platform», де користувач в залежності від ролі може бути викладачем або учнем. Викладач може приймати учнів на навчання, створювати уроки з різних доступних для нього предметів і з посиланням на урок за допомогою Zoom API, створювати групи для проведення уроку з кількома учнями, видалення учнів. Учень має змогу надсилати запит до обраного ним викладача, отримувати звістку про урок і посилання на конференцію при початку уроку.

Веб-додаток «Tutoring platform» може застосовуватися як і для проведення викладачем приватних уроків з учнями, так і бути інтегрованим в шкільну систему при необхідності онлайн навчання.

Проект також подається як практичне використання Spring Framework, для розробки ефективних і легко розширюваних веб-додатків, за рахунок

слабкої зв’язності його компонентів.

У майбутньому проект можна розширити за допомогою введення Google Calendar API, для зручнішого подання запланованих уроків. Добавити нові ролі користувачам, добавивши преміум ролі з більшим функціоналом. Це можна легко реалізувати за допомогою функціоналу, який надає Spring Security.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

API – Application Programming Interface;

DI– Dependency injection;

HTML – Hypertext Markup Language;

ID– Identity column;

IoC– Inversion of control;

JPA – Java Persistence API;

MVC – Model View Controller;

ORM – Object Relational Mapping;

POJO– Plain Old Java Object;

SQL – Structured Query Language;

URL – Uniform Resource Locator;

БД – база даних;

СУБД– система управління базами даних.

ВСТУП

Оцінка сучасного стану об’єкта розробки. У сучасному світі онлайн навчання стає все більш важливим і незамінним в сфері освіти. Пандемія стала серйозним рушієм для його розвитку. Через неї традиційна парадигма навчання неабияк змінилась, що надало безліч можливостей як для викладачів, так і для здобувачів освіти. Що ж до веб-технологій, вони також не стоять на місці і через кризу спричинену COVID-19, все глибше переплітаються з нашим життям. Spring Framework є найоптимальнішим вибором для розробки веб-додатків мовою програмування Java, і він має широкий набір функціональних інструментів, яких вистачить для будь-яких потреб у розробці. Це сприяє швидкому старту проекту, зменшує зв’язність, полегшує розширення проекту і подальшу підтримку.

Актуальність роботи та підстави для її виконання. Актуальність роботи визначається зростанням популярності онлайн навчання, потребою у гнучкому та індивідуальному підході до навчання, та зручністю управління навчальним процесом, що можна реалізувати за допомогою сучасних і ефективних фреймворків розробки,

Мета й завдання роботи. Мета роботи полягає в розробці веб-додатку для навчання на основі Spring Framework. Цей додаток має на меті забезпечити зручну та ефективну платформу для навчальної взаємодії між викладачами та студентами. Завдання роботи включатиме:

1. Розробка користувацького інтерфейсу використовуючи HTML, CSS JavaScript.
2. Реалізація функціональності яка забезпечуватиме авторизацію і автентифікацію користувача за допомогою Spring Security
3. Створення продуманої архітектури проекту
4. Реалізація функціональності яка забезпечуватиме взаємодію між студентами та викладачами.

Об'єкт , методи й засоби розроблення. Об’єктом роботи є процес розробки веб додатків з використанням мови програмування Java і допоміжних

фреймворків. Під час розробки додатку використовувалася IDE IntelliJ Idea, яка є одним з найкращих середовищ розробки на мові програмування Java. Було використано pgAdmin4 для створення БД і необхідних таблиць. Для зв’язування рядків таблиць з сутностями в Java коді, використовувалась Hibernate – один з найкращих ORM фреймворків. Щоб зберегти додаток слабозв’язаним для подальшого простого розширення проекту, використовувалось DI.

Можливі сфери застосування. Веб-додаток може бути використаний як і для приватного навчання, так і в школах, університетах, освітніх установах для забезпечення дистанційного навчання.

1 ОСНОВИ СТВОРЕННЯ ВЕБ-ДОДАТКІВ

1.1 Поняття веб-додатку

Веб-додаток (веб-застосунок) – це програмне забезпечення, розроблене для використання через веб-браузер. Він знаходиться на веб-сервері і доступний користувачам за допомогою Інтернету. Веб-додатки використовуються для надання різноманітних послуг та функціональності, таких як відправлення електронних листів, онлайн-покупки, соціальна мережа, банківські операції, управління контентом та багато іншого.

Роль веб-додатків у сучасному цифровому світі є надзвичайно важливою. Веб-додатки забезпечують зручну та доступну платформу для взаємодії користувачів з інформацією та послугами. Вони дозволяють отримувати доступ до потрібних даних та функціональності через веб-браузер, що робить їх універсальними та зручними для широкої аудиторії.

Веб-додатки забезпечують можливість доступу до інформації та послуг з будь-якого місця на планеті, де є доступ до Інтернету. Це робить їх надзвичайно зручними для користувачів, які можуть отримувати необхідну інформацію та виконувати різноманітні дії незалежно від свого місцезнаходження.

Завдяки веб-доступу та глобальній доступності, веб-додатки стають все більш популярними і забезпечують користувачам зручну взаємодію з послугами та інформацією в онлайн-середовищі.

1.2 Розробка веб-додатків

Розробка веб-додатків у сучасному світі відіграє важливу роль у цифровій трансформації бізнесу та забезпечує безперервний доступ до інформації та сервісів через Інтернет. Цей процес зазнає постійних змін та розвитку, оскільки веб-технології, фреймворки та інструменти швидко еволюціонують. Сучасна

веб-розробка поділяється на дві основні частини: frontend і backend, які відповідають за різні аспекти створення веб-додатків.

1.2.1 Frontend

Frontend відповідає за створення користувацького інтерфейсу, який відображається у веб-браузері. HTML, CSS і JavaScript і по сьогоднішній день є основними технологіями які використовуються у цій сфері. HTML використовується для структурування та розмітки вмісту веб-сторінки, CSS - для оформлення та стилізації елементів, а JavaScript - для динамічної взаємодії та програмної логіки. Ці технології працюють разом для створення багатофункціональних веб-додатків зі зручним інтерфейсом та різноманітними функціями. Крім основних технологій, для розробки frontend веб-додатків використовуються різні фреймворки та бібліотеки, такі як React, Angular, Vue.js і багато інших. Ці інструменти надають готові компоненти, шаблони та інструменти для зручного розроблення і підтримки веб-додатків.

1.2.2 Backend

Backend з іншого боку, відповідає за логіку та обробку даних веб-додатка. Розробники backend працюють з серверними мовами програмування, такими як Java, Python, PHP або Node.js. У сфері backend-розробки існує багато популярних фреймворків, таких як Spring (для Java), Django (для Python), Laravel (для PHP) та Express.js (для Node.js). Фреймворки дозволяють розробникам швидше створювати функціональність, забезпечують різноманітні модулі та інструменти для роботи з базами даних, маршрутизацією, аутентифікацією та багато іншого. Також важливим аспектом backend-розробки є робота з базами даних. Розробники backend використовують різноманітні СУБД такі як PostgreSQL, MySQL або MongoDB, для збереження та управління даними веб-додатків. Вони розробляють структуру бази даних, створюють запити для отримання, оновлення

та видалення даних, а також забезпечують безпеку та оптимальну продуктивність бази даних. Backend-розробники також використовують мови запитів, такі як SQL, для взаємодії з базою даних, виконання складних запитів і оптимізації їх продуктивності.

1.2.3 Комунікація між backend і frontend

Комунікація між фронтендом (frontend) і бекендом (backend) є важливою складовою розробки веб-додатків. Існує кілька способів, якими фронтенд та бекенд можуть взаємодіяти між собою: HTTP запити (полягає у відправленні запитів GET, POST, PUT, DELETE до API backend-у і отримання відповідей), REST або GraphQL API, WebSockets.

Крім цього, також можна використовувати Thymeleaf, який є шаблонним двигуном для розробки веб-додатків у Java-середовищі. Thymeleaf дозволяє розміщувати HTML-код в шаблонах і динамічно заповнювати їх даними з серверної сторони.

2 ДОСЛІДЖЕННЯ ФРЕЙМВОРКУ SPRING ДЛЯ РОЗРОБКИ ВЕБ-ДОДАТКУ

2.1 Spring Framework як сучасний інструмент розробки веб-додатків

Spring є одним з найпопулярніших і потужних фреймворків для розробки веб-додатків у сучасній веб-розробці. Він надає широкий спектр функцій та інструментів, що допомагають розробникам швидко і ефективно будувати як прості, так і складні веб-додатки.

Spring дозволяє легко створювати корпоративні програми Java. Він надає все необхідне для використання мови Java у корпоративному середовищі, з підтримкою Groovy і Kotlin як альтернативних мов у JVM, а також з гнучкістю для створення багатьох типів архітектур залежно від потреб програми. [1]

Spring фокусується на принципах IoC (Inversion of control) і DI (Dependency injection), що дозволяє розробникам легко управляти компонентами додатку завдяки розділенню обов'язків та модульності. Фреймворк надає механізми для створення, конфігурації та управління компонентами, що дозволяє зосередитися на бізнес-логіці додатку, забезпечуючи його масштабованість та підтримку.

У сучасній веб-розробці використання Spring є актуальним і практичним, оскільки він надає потужні інструменти та підходи для швидкої і ефективної розробки веб-додатків з високою масштабованістю, надійністю та безпекою.

2.1.1 Spring Boot

Одним із найважливіших компонентів Spring є Spring Boot. Він дозволяє легко створювати автономні додатки на основі Spring, які можна «просто запускати»(“just run”).[2] Він забезпечує початкові налаштування веб-додатків, автоматичну конфігурацію та управління залежностями. Spring Boot дозволяє розробникам швидко створювати готові до використання веб-додатки, зменшуючи необхідність в ручному налаштуванні та конфігурації.

2.1.2 Spring Security

Spring Security – це потужний інструмент для забезпечення безпеки веб-додатків. Як і в усіх проектах Spring, справжня сила Spring Security полягає в тому, наскільки легко його можна розширити, щоб відповідати користувацьким вимогам[3]. Spring Security забезпечує надійний контроль доступу до ресурсів у веб-додатку. Завдяки можливості визначення ролей та дозволів користувачів, розробники можуть точно налаштувати, які ресурси мають бути доступними для окремих груп користувачів. Це забезпечує високий рівень безпеки та захисту конфіденційної інформації.

Забезпечення безпеки веб-додатків також включає захист від різних атак, таких як перехоплення сесії, міжсайтовий скриптінг та впровадження коду. Spring Security автоматично застосовує різноманітні заходи безпеки, такі як фільтри безпеки та захист від CSRF-атак, що дозволяє розробникам зосередитись на розробці додатку, маючи впевненість у його безпеці.

2.1.3 Spring MVC

Spring MVC — це оригінальний веб-фреймворк, побудований на API Servlet, який був включений у Spring Framework із самого початку. Формальна назва «Spring Web MVC» походить від назви вихідного модуля (spring-webmvc), але більш відома як «Spring MVC».[4] Це є один з ключових компонентів фреймворку Spring. Це потужний і гнучкий інструмент, призначений для розробки веб-додатків на основі шаблону проектування MVC. Цей шаблон розділяє логіку додатку на три основні компоненти: Model, View та Controller. Кожен з цих компонентів має свою відповідальність і взаємодіє з іншими для створення повноцінного веб-додатку.

2.1.4 Spring JPA

Spring Data JPA, частина сімейства Spring Data, дозволяє легко впроваджувати репозиторії на основі JPA. Цей модуль стосується розширеної підтримки рівнів доступу до даних на основі JPA. Це полегшує створення додатків на основі Spring, які використовують технології доступу до даних.[5] Spring Data JPA автоматично генерує SQL-запити на основі методів репозиторію. Це дозволяє розробникам працювати з базою даних, не займаючись написанням власних SQL-запитів. Він також підтримує різні типи запитів, такі як запити за критеріями, нативні SQL-запити тощо.

3 ПРОЄКТУВАННЯ ВЕБ-ДОДАТКУ

3.1 Засоби розробки

Для реалізації backend частини веб-додатку для забезпечення онлайн навчання, була вибрана мова програмування Java, з використанням фреймворку Spring. Середовище розробки IntelliJ IDEA яка надає велику кількість корисного функціоналу. Обрана СУБД – PostgresSQL, графічний інтерфейс для роботи з нею – PgAdmin 4. Він надає зручний графічний інтерфейс для створення і обробки БД. Як реалізація ORM був використаний фреймворк Hibernate для об'єктно-реляційного відображення даних. Hibernate дозволяє працювати з базою даних, використовуючи об'єктно-орієнтований підхід, безпосередньо з коду програми.

Для реалізації frontend частини роботи використовувались HTML, CSS, JavaScript, у вищезгаданому середовищі розробки IntelliJ IDEA. Для поєднання обох частин (frontent I backend), був обраний шаблонізатор Thymeleaf - сучасний серверний движок шаблонів Java як для веб, так і для автономних середовищ.[6]

3.2 Проектування сутностей

Для подальшої реалізації бізнес логіки програми, визначимо сутності, які представлятимуть користувачів програми. Всі класи сутностей є POJO класами, тому над ними прописана анотація @Data, зручна анотація швидкого доступу, яка об’єднує функції @ToString, @EqualsAndHashCode, @Getter/@Setter та @RequiredArgsConstructorразом[7].

3.2.1 Класи для представлення користувачів

Головний клас User представляє собою інформацію про користувача. Поля класу: email, ім’я, прізвище, стать, пароль, ID, номер телефону, username, ролі.

Також він містить посилання на класи Teacher і Student, і в залежності від обраної ролі може бути або викладачем або учнем. Ця інформація задається при автентифікації. Клас User буде використовуватись для поєднання з Spring Security. Повна інформація про клас у форматі UML(наведені поля і методи), наведено на рис. 3.1

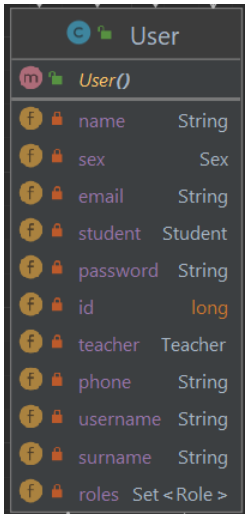

Рисунок 3.1 – Клас для представлення користувача

Клас Teacher представляє собою викладача. Поля класу: ID, досвід, предмети які викладає, групи які навчає, місце де надає перевагу для проведення уроку(онлайн чи офлайн), рівень студентів, коли був зареєстрований, студенти яких навчає, студенти яких він не прийняв ще на навчання, для рейтингу викладача створені поля загальна оцінка (сума всіх оцінок) і кількість оцінок, щоб в результаті отримувати середній бал, оплата за годину у випадку приватних уроків, зв'язок з юзером який містить інформацію про особисті дані викладача. Повна інформація про клас у форматі UML наведено на рисунку 3.2.

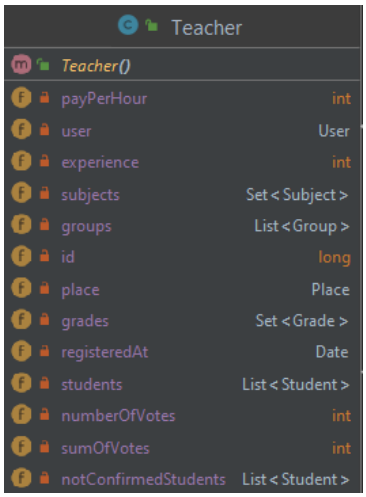

Рисунок 3.2 – Клас для представлення викладача

Клас Student представляє собою учня. Поля класу: ID, групи в яких навчається, викладачі які прийняли цього учня на навчання, викладачі яким був відправлений запит на навчання, зв'язок з юзером який містить інформацію про особисті дані учня. Повна інформація про клас у форматі UML наведено на рисунку 3.3.

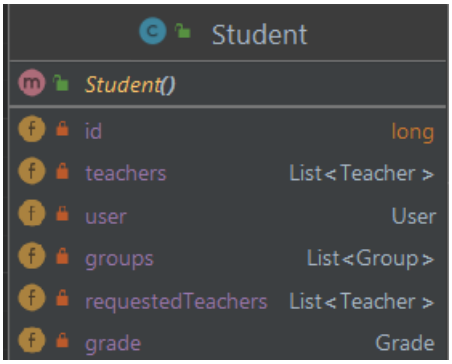

Рисунок 3.3 – Клас для представлення учня

3.2.2 Класи для представлення зв’язків

Клас Group представляє собою групу – множину яка зберігає набір учнів і пов’язує її з одним викладачем. Клас містить наступні поля: ID, назва групи, викладач який проводить уроки для цієї групі, список запланованих уроків і множина студентів яка представляє групу. Повна інформація про клас у форматі UML наведено на рисунку 3.4.

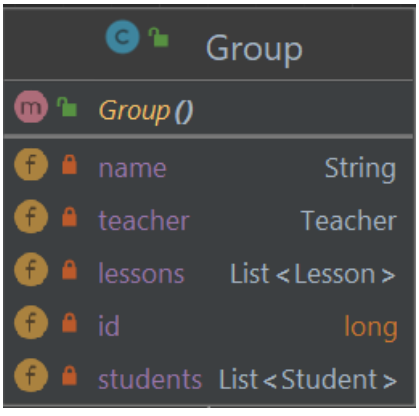

Рисунок 3.4 – Клас для представлення групи

Клас Lesson представляє собою урок – запланована подія викладачем.

Клас містить наступні поля: ID, дата початку уроку, тривалість уроку, групу в якій запланований цей урок, дисципліна яка буде викладатись на уроці. Повна інформація про клас у форматі UML наведено на рисунку 3.5.

Рисунок 3.5 – Клас для представлення уроку

3.2.3 Допоміжні класи для бізнес логіки

Для зберігання таких даних як стать(Sex), роль(Role), місце проведення уроку(Place), предмет(Subject), рівень освіти(Grade), були створені спеціальні типи даних для представлення набору констант. Всі вони зображені на рис. 3.6

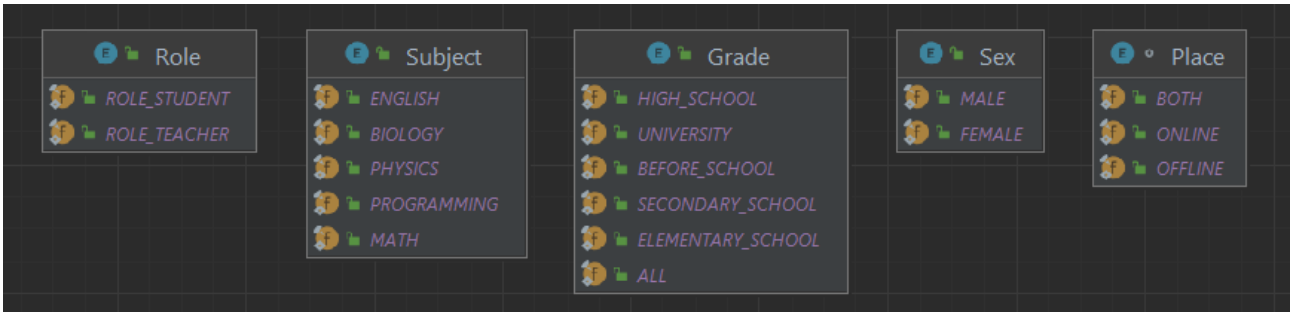

Рисунок 3.6 – Необхідні enums

Також реалізовані кастомні помилки для більш інформативного висвітлення винятків, які можуть виникати у програмі. Всі вони зображені на рис. 3.7

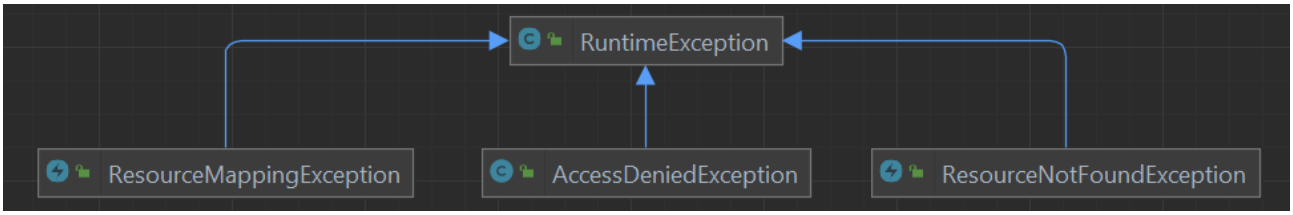

Рисунок 3.7 – Кастомні винятки

3.2.4 Архітектура моделей

Описані вище класи утворюють архітектуру моделі веб-додатку. Використовуючи функціонал IntelliJ Idea, була побудована UML діаграма класів для представлення зв'язків між сутностями. Зв'язки відповідають наступному формату

- Відношення один до одного: 1 – 1
- Відношення один до багато: 1 – *
- Відношення багато до багато: * – *

Діаграма класів з відношеннями зображені на рис 3.8

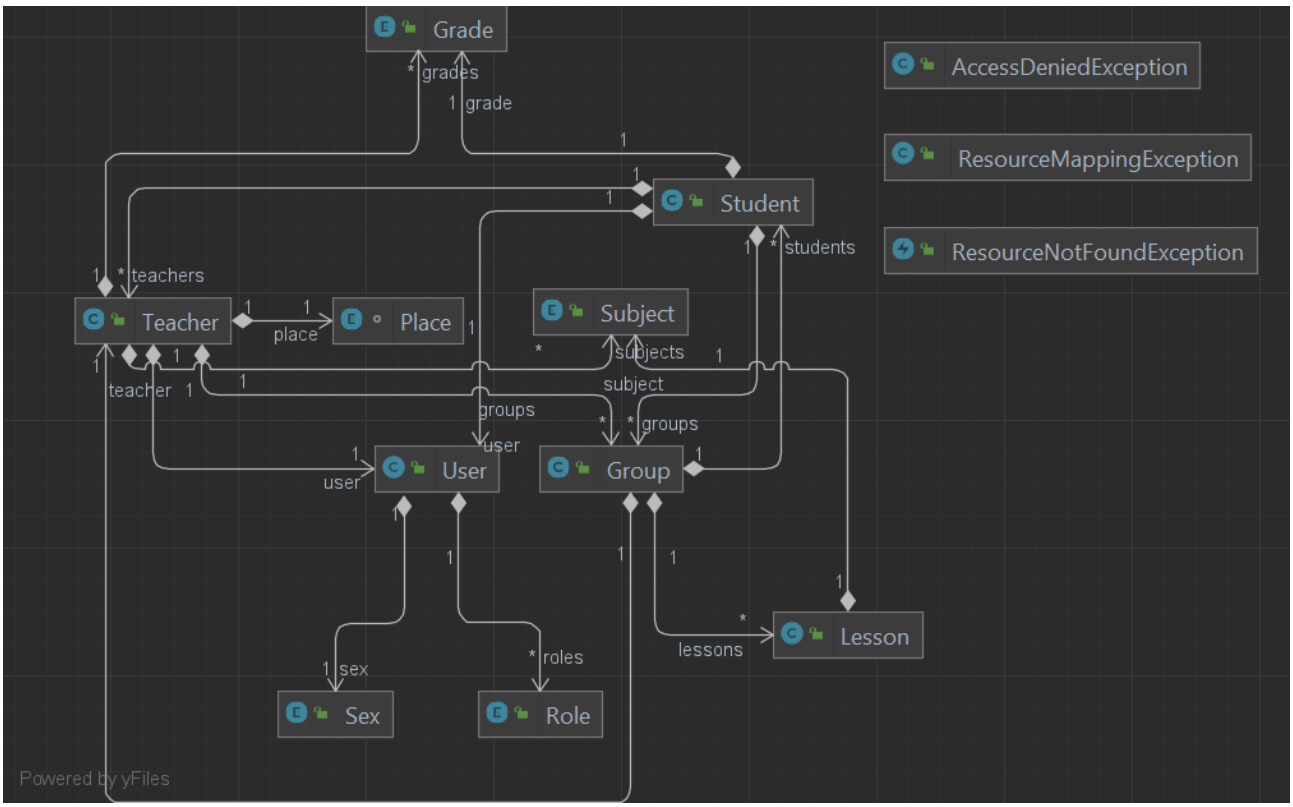

Рисунок 3.8 – Діаграма класів

3.3 Проектування сутностей в БД

Проектування бази даних є важливою частиною розробки програмного додатку, і для цього можна використовувати фреймворк Hibernate. Hibernate є

реалізацією ORM для мови програмування Java, , яка допомагає вирішувати проблеми об'єктно-реляційного відображення.

Hibernate спрощує створення бази даних та дозволяє розробникам зосередитися на бізнес-логіці додатку. Він надає можливість використовувати різні можливості для створення баз даних, а завдяки інтерфейсам для провайдерів баз даних, можна використовувати Hibernate з різними СУБД.

З використанням анотації @Entity над класом-моделлю, ми позначаємо, що цей клас повинен мати відповідну таблицю в базі даних. Анотацією @Table(name = "…") прописується з якою таблицею в БД встановлюється зв'язок.

Поля класу використовуються як поля таблиці. Назви полів за замовчуванням співпадають з назвами полів, але за допомогою анотації @Column ми можемо коригувати цей процес. Так як naming convention в Java і SQL відрізняється, у класах @Column була застосована для всіх полів.

Важливою частиною бази даних є зовнішні та головні ключі, а також зв'язки з іншими таблицями. Hibernate надає можливість встановлювати ці зв'язки за допомогою анотацій. Були використані зв'язки @OneToOne - зв'язок один до одного, @OneToMany - зв'язок один до багатьох, @ManyToOne - зв'язок багато до одного, і @ManyToMany - зв'язок багато до багатьох. Для останньої в БД будується допоміжна таблиця яка зберігає ключі як пару значень ID-шок таблиць, які вона зв'язує. Наприклад сутності Teacher і Student відповідають такому зв'язку оскільки викладач може мати багато учнів, так і учень може мати багато викладачів. Допоміжна таблиця наведена на рис 3.9.

```
create table if not exists teachers_students
(
    teacher_id bigint not null,
    student_id   bigint not null,
    primary key (teacher_id, student_id),
    foreign key (teacher_id) references teachers (id),
    foreign key (student_id) references students (id)
);
```

Рисунок 3.9 – Таблиця для представлення зв'язку багато до багато

Зв'язок прописаний використовуючи анотації Hibernate, буде виглядати наступним чином (рис 3.10, рис 3.11)

```
@ManyToMany(mappedBy = "students")
private List<Teacher> teachers;
```

Рисунок 3.10 Зв’язок з класу Student до класу Teacher

```
@ManyToMany
@JoinTable(
        name = "teachers_students",
        joinColumns = @JoinColumn(name = "teacher_id"),
        inverseJoinColumns = @JoinColumn(name = "student_id"))
private List<Student> students;
```

Рисунок 3.11 Зв’язок з класу Teacher до класу Student

Для поля, яке виконує роль унікального ідентифікатора, використовується анотація @Id. Також можна налаштувати стратегію інкрементації ідентифікатора, передаючи відповідний параметр у цю анотацію.

Загальний список створених таблиць наведений на рис. 3.12

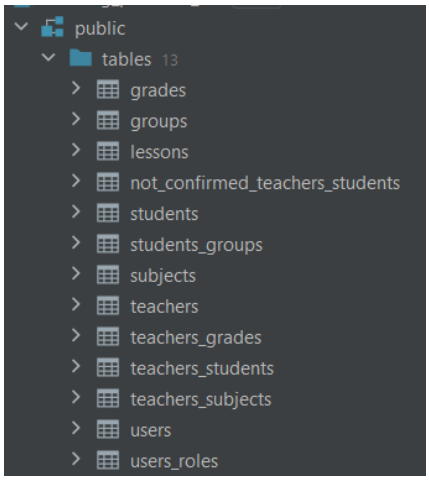

Рисунок 3.12 Всі створені таблиці в БД

4 РОЗРОБКА ВЕБ-ДОДАТКУ «TUTORING PLATFORM»

4.1 Бізнес логіка

Веб-додаток розроблявся дотримуючись патерну MVC, тому розробку можна розділити на три основні частини: модель, представлення та контролер. На початку необхідно визначити бізнес логіку проекту. Всі класи в моделі позначенні анотацією @Data що робить їх POJO класами.

Було вирішено щоб інформацію про користувача і інформацію необхідну для навчального процесу розмістити по різним класам: User зберігає інформацію про користувача що більш потрібно під час автентифікації, Teacher і Student зберігають інформацію необхідну для навчального процесу.

Оскільки викладач має сам обирати учнів у яких хоче викладати, і відповідно учень обирати викладачів, логіка цього процесу наступна. Учень робить запит до викладача(список всіх викладачів надається учню), і тільки після погодження викладачем навчання цього учня, встановиться зв'язок між ними.

Оскільки викладач мусить мати можливість проводити урок не тільки з одним учнем, був створений клас Group. Викладач матиме можливість зі списку своїх учнів зібрати групу і вже створеній групі запланувати необмежену кількість уроків. Для представлення уроків реалізований клас Lesson з часовими межами його проведення. Якщо урок закінчився, тоді він автоматично видаляється. Видалити його може і викладач.

При реєстрації користувача, логіка коректності даних покладається на анотації надані Hibernate-ом. Наприклад поле username для представлення унікального імені користувача визначена наступним патерном, рис 4.1

```
@NotNull
@Size(min = 6, max = 16, message = "Username should be between 6 and 16 characters")
@Pattern(regexp = "(?<![^\\W_])\\b[a-z]+(?:_[a-z]+)*\\b(?![^\\W_])", message = "user
private String username;
```

Рисунок 4.1 Зв’язок представлений з класу Student

Отже якщо поле не було заповнене, або не відповідатиме розміру, або не

відповідатиме патерну (тільки маленькі букви і нижній дефіс між ними), тоді користувача поверне на попередню форму реєстрації і повідомить про помилку.

4.2 Класи сервіси і класи репозиторії

Наступною частиною для реалізації є реалізація шару сервісу. Класи-сервіси є важливими компонентами веб-додатка, розробленого за паттерном MVC. Вони містять логіку роботи додатка, обробляють запити від класів-контролерів та взаємодіють з класами-репозиторіями. У додатку можна виокремити різні задачі та поділити їх логічно. Наприклад, логіку обробки інформації про користувача можна розмістити в окремому сервісі, який буде містити методи для роботи з цими даними. Такий сервіс допоможе зберігати, видаляти, редагувати та додавати користувачів.

Класи-сервіси позначаються анотацією @Service, яка вказує на їх роль у додатку. Ця анотація поліпшує читабельність коду та допомагає зрозуміти бізнес-процеси, які реалізує програма. Методи класів-сервісів використовують класи-репозиторії, позначені анотацією @Repository, щоб взаємодіяти з базою даних. Це необхідно для виконання операцій додавання, видалення, редагування та отримання даних. Репозиторії значно спрощують процес розробки, оскільки надають реалізацію простих запитів SELECT, UPDATE, INSERT, DELETE, що дозволяє розробнику на вищому рівні абстракції, що, як результат, пришвидшує час розробки програми.

Після обробки даних з класів-репозиторіїв, класи-сервіси можуть виконати додаткові маніпуляції та повернути результат класу-контролеру, або в викинути виняток який вище буде перехоплений Exception Handler-ом.Список класів сервісів наведений на рис. 4.2

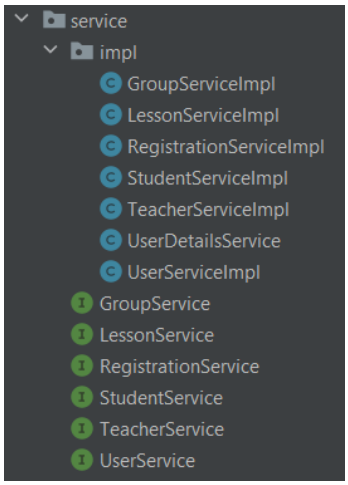

Рисунок 4.2 Список сервісів представлених як інтерфейси і їхня реалізація
Список класів репозиторіїв наведений на рис. 4.3

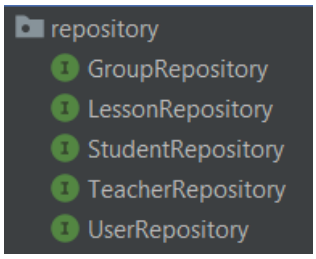

Рисунок 4.3 Список репозиторіїв для класів моделей

4.3 Класи контролерів

Класи-контролери виконують роль вхідних точок програми та оброблюють HTTP запити. Вони діляться на прошарки, відповідно до логіки, яку вони втілюють. Щоб позначити клас-контролер як компонент Spring, йому додається анотація @Controller, яка дозволяє Spring розпізнати його та обробляти його залежності.

Класи-контролери також використовують анотації @GetMapping, @PostMapping, @DeleteMapping та @PatchMapping, які надають можливість вказати, які HTTP методи та URL-шаблони пов'язані з відповідними методами класу. Анотація @RestController поєднує в собі анотації @Controller та @ResponseBody і вказує, що клас-контролер повинен повертати результати у

вигляді JSON-об'єктів.

Анотація @RequestMapping("/teacher") узагальнює початок запитів і всі запити які потрапляють у контролер мають починатись з /teacher у своєму url-і.

Релізовано було 4 контролери:

- AuthController – контролер у якому реалізована логіка реєстрації користувачів.
- TeacherController – контролер у якому реалізована логіка яка дозволена користувачу, який є викладачем.
- StudentController – контролер у якому реалізована логіка яка дозволена користувачу, який є учнем.
- MainContorller – контролер у якому реалізована переадресація користувача, в залежності від його ролі.

4.4 Інтерфейс веб-додатку

В розробці веб-додатку був використаний HTML, CSS і JavaScript для реалізації фронтенду. Був використаний шаблонізатор Thymeleaf для полегшення роботи з генерацією HTML-сторінок у серверній частині додатку. Thymeleaf дозволяє вбудовувати динамічні дані у HTML-шаблони, що спрощує відображення даних з сервера на сторінках додатку. Це дозволяє ефективно керувати виглядом та поведінкою користувацького інтерфейсу і забезпечити зручну взаємодію з користувачами додатку.

ВИСНОВКИ

У сучасному світі онлайн навчання стає все більш важливим і необхідним в сфері освіти, особливо з урахуванням впливу пандемії COVID-19. Розвиток веб-технологій та наявність ефективних фреймворків розробки, таких як Spring Framework, створюють унікальні можливості для створення зручних та інтерактивних платформ для навчання.

Метою даної роботи було розробити веб-додаток для навчання на основі Spring Framework, який забезпечує зручну та ефективну платформу для взаємодії між викладачами та студентами. Для досягнення цієї мети були вирішені завдання, такі як розробка користувацького інтерфейсу з використанням HTML, CSS та JavaScript, забезпечення авторизації та автентифікації користувачів за допомогою Spring Security, створення продуманої архітектури проекту та реалізація взаємодії між студентами та викладачами. Під час розробки додатку використовувалися такі засоби, як IntelliJ IDEA для розробки на мові програмування Java, pgAdmin4 для створення бази даних та таблиць, а також Hibernate для зв'язування об'єктів Java з таблицями бази даних. Використання DI дозволило створити слабко зв'язаний додаток, що спрощує подальше розширення та підтримку проекту.

Отриманий веб-додаток "Tutoring Platform" може бути успішно використаний як для приватного навчання, так і в освітніх установах, школах та університетах для забезпечення дистанційного навчання.

Готовий проект «Tutoring platform» можна знайти на GitHub[8]

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Spring. [Електронний ресурс] // Spring framework overview. – Режим доступу: https://docs.spring.io/spring-framework/reference/overview.html
2. Spring. [Електронний ресурс] // Spring boot. – Режим доступу: https://spring.io/projects/spring-boot#overview
3. Spring. [Електронний ресурс] // Spring security. – Режим доступу: https://spring.io/projects/spring-security
4. Spring. [Електронний ресурс] // Spring web MVC. – Режим доступу: https://docs.spring.io/spring-framework/reference/web/webmvc.html
5. Spring. [Електронний ресурс] // Spring data JPA. – Режим доступу: https://spring.io/projects/spring-data-jpa
6. Thymeleaf. [Електронний ресурс] // Thymeleaf. – Режим доступу: https://www.thymeleaf.org/.
7. ProjectLombok. [Електронний ресурс] // @Data. – Режим доступу: https://projectlombok.org/features/Data
8. GitHub. [Електронний ресурс] // Репозиторій проекту. – Режим доступу: https://github.com/mykolaKhlopyk/TutoringPlatform