

ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЄКТ

на тему:

«Програмний додаток мережі спортивних клубів»

Студента 4 курсу, 7 групи,
спеціальності 121 «Інженерія
програмного забезпечення»
освітньої програми «Інженерія
програмного забезпечення»

підпис студента

Чеботаєва
Деніса
Олеговича

Науковий керівник
кандидат технічних наук, доцент
кафедри інженерії програмного
забезпечення та кібербезпеки

підпис керівника

Власенко
Лідія
Олександрівна

Гарант освітньої програми
кандидат технічних наук,
доцент кафедри інженерії
програмного забезпечення та
кібербезпеки

підпис керівника

Рзаєва
Світлана
Леонідівна

Державний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь бакалавр

Спеціальність 121 «Інженерія програмного забезпечення»

Затверджую

Зав. кафедри інженерії програмного
забезпечення та кібербезпеки

Криворучко О. В.

«14» листопада 2022 р.

Завдання

на випускний кваліфікаційний проєкт студентіві

Чеботаєву Денису Олеговичу

(прізвище, ім'я, по батькові)

1. Тема випускного кваліфікаційного проєкту «Програмний додаток мережі спортивних клубів»

Затверджена наказом ректора від «6» грудня 2022 р. № 3288

2. Строк здачі студентом закінченого проєкту 5 червня 2023

3. Цільова установка та вихідні дані до проєкту

Мета проєкту: розробити програмний додаток мережі спортивних клубів

Об'єкт дослідження: інформаційна система взаємодії з клієнтами спортивного залу

Предмет дослідження: сучасні технології для розробки веб-орієнтованої інформаційної системи приватного підприємства

4. Консультанти проєкту із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускного кваліфікаційного проєкту (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1 АНАЛІЗ ІНФОРМАЦІЙНИХ СИСТЕМ

1.1 Загальна характеристика інформаційних систем

1.2 Огляд існуючих рішень

1.3 Постановка задачі

1.4 Висновки до розділу 1

РОЗДІЛ 2 ДЕТАЛЬНИЙ АНАЛІЗ ПРЕДМЕТА ДОСЛІДЖЕННЯ

2.1. Основні положення

2.2. Аналіз функціональних вимог

2.3. Специфікація вимог до програмного продукту

2.4. Специфікація нефункціональних вимог

2.5. Технічне завдання

2.6. Висновок до розділу 2

РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Вибір мови програмування та додаткових фреймворків

3.2 Вибір реляційної бази даних

3.3 Проектування бази даних

3.4 Розробка серверної частини системи

3.5 Розробка клієнтської частини системи

3.6 Демонстрація розробленої системи

3.7 Висновок до розділу 3

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

6. Календарний план виконання проєкту

№ пор.	Назва етапів випускного кваліфікаційного проєкту	Строк виконання етапів проєкту	
		за планом	фактично
1	2	3	4
1.	<i>Вибір теми випускного кваліфікаційного проєкту</i>	21.09.2022	21.09.2022
2.	<i>Розробка та затвердження завдання на проєкт</i>	14.11.2022	14.11.2022
3.	<i>Вступ та перелік літературних джерел</i>	23.12.2022	23.12.2022
4.	<i>Розділ 1. Аналіз інформаційних систем</i>	27.01.2023	27.01.2023
5.	<i>Розділ 2. Детальний аналіз предмета дослідження</i>	03.03.2023	03.03.2023
6.	<i>Розділ 3. Реалізація програмного забезпечення</i>	14.04.2023	14.04.2023
7.	<i>Висновки</i>	28.04.2023	28.04.2023
8.	<i>Здача випускного кваліфікаційного проєкту на кафедрі (перша перевірка)</i>	17.05.2023	17.05.2023
9.	<i>Підготовка автореферату та презентації доповіді</i>	26.05.2023	26.05.2023
10.	<i>Попередній захист випускного кваліфікаційного проєкту</i>	29.05.2023 – 02.06.2023	30.05.2023
11.	<i>Зовнішнє рецензування випускного кваліфікаційного проєкту</i>	05.06.2023	05.06.2023
12.	<i>Здача прошого випускного кваліфікаційного проєкту на кафедрі</i>	05.06.2023	05.06.2023
13.	<i>Публічний захист випускного кваліфікаційного проєкту</i>	21.06.2023	21.06.2023

7. Дата видачі завдання «14» листопада 2022 р.

8. Науковий керівник випускного кваліфікаційного проєкту

Власенко Л.О

(прізвище, ініціали, підпис)

9. Гарант освітньої програми

Рзаєва С.Л.

(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент

Чеботаєв Д. О.

(прізвище, ініціали, підпис)

АННОТАЦІЯ

Відповідно до мети дослідження робота присвячена розробці програмного додатку мережі спортивних клубів.

В результаті порівняльного аналізу аналогічних рішень визначено, що на ринку програмних додатків мережі спортивних клубів відсутні, або не є оптимальними готові рішення.

Розробка серверної і клієнтської частини програмного продукту виконано в середовищі IntelliJ IDEA з використанням Java, Spring Boot, JavaScript.

Ключові слова: інформаційна система, WEB, мережа, спортивний клуб, управління, автоматизація, зберігання даних, веб-розробка, аналітика, інтеграція.

ABSTRACT

In accordance with the research objective, this work is dedicated to the development of a software application for a network of sports clubs.

Through a comparative analysis of similar solutions, it has been determined that there is a lack of, or suboptimal, ready-made solutions in the market for software applications in the network of sports clubs.

The development of the server-side and client-side components of the software product was performed using IntelliJ IDEA environment, utilizing Java, Spring Boot, and JavaScript.

Keywords: information system, WEB, network, sports club, management, automation, data storage, web development, analytics, integration.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ІС - Інформаційна система

WEB - Веб-додаток або Веб-сайт

ІТ (англ. Information Technologies) Інформаційні технології — це система методів, процесів та способів використання обчислювальної техніки і систем зв'язку для створення, збору, передачі, пошуку, оброблення та поширення інформації з метою ефективної організації діяльності людей

ПЗ – програмне забезпечення

XML (англ. Extensible Markup Language) – розширювана мова розмітки

Backend – код серверної частини клієнт-серверного або веб-додатка

Frontend – код клієнтської частини додатка, з якою взаємодіє користувач.

СУБД – система управління базами даних.

<i>ДТЕУ 121 07-26.БР</i>				
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>
Зав. каф.		Криворучко О.В.		14.04.23
Керівник		Власенко Л. О.		14.04.23
Гарант		Рзасва С.Л.		14.04.23
Розробив		Чеботаєв Д. О.		14.04.23
<i>Програмний додаток мережі спортивних клубів</i>				
<i>Перелік умовних скорочень</i>				
<i>Стадія</i>		<i>Аркуш</i>		<i>Аркушів</i>
<i>ПС</i>		<i>2</i>		<i>65</i>
<i>Факультет інформаційних технологій 4 курс, 7 група</i>				

ЗМІСТ

ВСТУП	4
РОЗДІЛ 1 АНАЛІЗ ІНФОРМАЦІЙНИХ СИСТЕМ	6
1.1. Загальна характеристика інформаційних систем	6
1.2. Огляд існуючих рішень	8
1.3. Постановка задачі	17
1.4. Висновки до розділу 1	19
РОЗДІЛ 2 ДЕТАЛЬНИЙ АНАЛІЗ ПРЕДМЕТА ДОСЛІДЖЕННЯ	21
2.1. Основні положення.....	21
2.2. Аналіз функціональних вимог	28
2.3. Специфікація вимог до програмного продукту	29
2.4. Специфікація не функціональних вимог	311
2.5. Технічне завдання.....	322
2.6. Висновки до розділу 2	333
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	37
3.1. Вибір мови програмування та додаткових фреймворків	37
3.2. Вибір реляційної бази даних.....	39
3.3. Проектування бази даних.....	40
3.4. Розробка серверної частини системи.....	45
3.5. Розробка клієнтської частини системи	54
3.6. Демонстрація розробленої системи	56
3.7. Висновки до розділу 3	60
ВИСНОВКИ ТА ПРОПОЗИЦІЇ	62
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	64
ДОДАТКИ	66

					<i>ДТЕУ 121 07-26.БР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Програмний додаток мережі спортивних клубів	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Зав. каф.		Криворучко О.В.		23.12.22		3	3	65
Керівник		Власенко Л.О.		23.12.22		Факультет інформаційних технологій 4 курс, 7 група		
Гарант		Рзаєва С.Л.		23.12.22				
Розробив		Чеботаєв Д.О.		23.12.22	<i>Зміст</i>			

ВСТУП

Дипломна робота присвячена розробці програмного додатку для мережі спортивних клубів. У сучасному світі все більшою популярністю користується здоровий спосіб життя та заняття спортом, тому попит на спортивні заклади зростає. Однак, збільшення кількості клієнтів призводить до потреби вдосконалювати систему їх обслуговування та автоматизувати багато процесів. Саме тому, розробка програмного додатку для мережі спортивних клубів може значно полегшити життя клієнтів та працівників закладів.

Незважаючи на те, що на сьогоднішній день існує велика кількість програмних додатків для мережі спортивних клубів, є кілька загальних проблем, з якими часто зустрічаються користувачі та працівники цих клубів:

- Недостатня функціональність: більшість програмних додатків для спортивних клубів не надають достатньої кількості функцій для задоволення потреб клієнтів та працівників, таких як бронювання занять, спілкування з тренерами та інші.
- Складність використання: деякі програмні додатки можуть бути складні для використання користувачами, особливо тими, хто не має досвіду в користуванні комп'ютерами або мобільними додатками.
- Низька надійність та стабільність: деякі програмні додатки можуть мати проблеми з надійністю та стабільністю, що може призвести до помилок, втрати даних та інших проблем.

					<i>ДТЕУ 121 07-26.БР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Програмний додаток мережі спортивних клубів	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Зав. каф.		Криворучко О.В.		23.12.22		<i>В</i>	<i>4</i>	<i>65</i>
Керівник		Власенко Л.О.		23.12.22		<i>Факультет інформаційних технологій 4 курс, 7 група</i>		
Гарант		Рзаєва С.Л.		23.12.22				
Розробив		Чеботаєв Д.О.		23.12.22				
					<i>Вступ</i>			

- Недостатня інтеграція з іншими системами: іноді програмні додатки для спортивних клубів не можуть взаємодіяти з іншими системами, такими як системи оплати, розклади занять, або бази даних клієнтів.
- Високі вимоги до апаратного забезпечення: деякі програмні додатки можуть вимагати високих технічних вимог до апаратного забезпечення, що може зробити їх недоступними для деяких користувачів.
- Метою дипломної роботи є розробка програмного додатку, що дасть можливість клієнтам мережі спортивних клубів бронювати заняття, стежити за своїми тренуваннями та результатами, а працівникам - керувати клієнтською базою, зберігати дані про заняття, контролювати оплату послуг та вести звітність.

Об'єкт дослідження - інформаційна система взаємодії з клієнтами спортивного залу. Предметом дослідження є сучасні технології для розробки веб-орієнтованої інформаційної системи приватного підприємства.

Для розробки програмного додатку буде використано сучасні технології та програмні інструменти, що дозволять розробити ефективний та функціональний додаток, здатний задовольнити потреби клієнтів та працівників мережі спортивних клубів.

Таким чином, дипломна робота має на меті розробити програмний додаток, який спростить процес обслуговування клієнтів та керування мережею спортивних клубів.

						ДТЕУ 121 07-26.БР	Аркуш 5
Зм.	Аркуш	№ докум	Підпис	Дата			

РОЗДІЛ 1

АНАЛІЗ ІНФОРМАЦІЙНИХ СИСТЕМ

1.1. Загальна характеристика інформаційних систем

Люди сприймають інформацію за допомогою різних органів почуттів: слуху, зору, дотику і нюху. Вони вивчили складні системи мов, щоб інтерпретувати ці сигнали, які отримують, і розпізнавати об'єкти, розуміти повідомлення і реагувати на різні стимули.

Ці інформаційно-передавальні знаки передаються за допомогою енергетичних явищ, таких як звукові хвилі, світлові хвилі, хімічні та електрохімічні стимули. У термінах інженерії, люди виступають як рецептори аналогових сигналів, а повідомлення, що передаються через ці сигнали, називаються аналоговою інформацією.

До появи цифрових комп'ютерів, когнітивна інформація зберігалася і оброблялася переважно у формі аналогових сигналів за допомогою технологій друку, фотографії та телефонії.

Коли мова йде про обробку інформації, яка зберігається в пам'яті, люди виявляються дуже вправними. Однак, обробка аналогової інформації, яка зберігається поза межами розуму, є складним завданням. Сучасні інформаційні технології спрощують цей процес шляхом перетворення інформації на цифрові сигнали, які відображають наявність або відсутність енергії, такої як електрика, світло або магнетизм. Ця форма інформації, представлена у вигляді двійкових станів, часто називається цифровою інформацією.

Зм.	Аркуш	№ докум.	Підпис	Дата	<i>ДТЕУ 121 07-26.БР</i>			
Зав. каф.		Криворучко О.В.		27.01.23	Програмний додаток мережі спортивних клубів	Стадія	Аркуш	Аркушів
Керівник		Власенко Л.О.		27.01.23		РІ	6	65
Гарант		Рзаєва С.Л.		27.01.23		Факультет інформаційних технологій		
Розробив		Чеботаєв Д.О.		27.01.23		4 курс, 7 група		

Сучасні інформаційні системи пройшли значні зміни віданалогової до цифрової форми інформації. Цей перехід відбувся настільки широко, що призвів до історичної трансформації способу створення, доступу та використання інформації людиною.

Інформаційна система є комплексом компонентів, які взаємодіють для збору, обробки, зберігання та поширення інформації. Вона включає апаратне забезпечення, програмне забезпечення, засоби зв'язку та саму інформацію. Комп'ютерна система складається з апаратних компонентів, таких як персональні комп'ютери та смартфони, програмного забезпечення, такого як операційні системи та програми, а також зв'язку, який дозволяє обмінюватись даними між компонентами системи.

Інформаційна система також включає саму інформацію, яка зберігається, обробляється та передається в рамках системи. Це можуть бути бази даних, тексти, зображення, відео та інші формати інформації. Користувачі інформаційної системи використовують її компоненти для взаємодії з інформацією та здійснення різних завдань, від комунікації з друзями до управління бізнес-процесами.

Кожна інформаційна система є унікальною в своєму роді, оскільки вона враховує специфіку певної діяльності або організації, для якої вона використовується. Це дозволяє використовувати інформаційну систему для різноманітних цілей, починаючи від управління ланцюгами поставок і закінчуючи взаємодією з цифровими ринками.

Унікальність кожної інформаційної системи полягає в тому, як вона впроваджується, налаштовується та використовується для конкретних потреб. Підприємства та корпорації використовують інформаційні системи для різних цілей, включаючи взаємодію з постачальниками та клієнтською базою, здійснення операцій, управління організацією та проведення маркетингових кампаній. Ці системи можуть використовуватися для управління ланцюгами

					ДТЕУ 121 07-26.БР	Аркуш 7
Зм.	Аркуш	№ докум	Підпис	Дата		

поставок, взаємодії з цифровими ринками та іншими різноманітними завданнями.

Люди також надають перевагу використанню інформаційних систем для спілкування зі своїми однолітками та друзями через соціальні мережі. Вони також залежать від цих систем для здійснення щоденних дій, таких як банківські операції, покупки або просто пошук знань і інформації.

Ця унікальна комбінація використання інформаційних систем підприємствами та користувачами для досягнення своїх цілей свідчить про важливість і вплив цих систем на сучасне суспільство. Вони дозволяють зберігати, обробляти та передавати інформацію з більшою ефективністю і спрощують багато аспектів нашої повсякденної діяльності. Основні з них наведені на рис. 1.1.



Рис. 1.1 – Загальна характеристика інформаційної системи

Джерело: побудовано автором

1.2. Огляд існуючих рішень

Коли робочий процес оптимізований, це означає, що він ефективно структурований і виконується з максимальною продуктивністю при

					ДТЕУ 121 07-26.БР	Аркуш 8
Зм.	Аркуш	№ докум	Підпис	Дата		

оптимальному використанні ресурсів. Оптимізація робочого процесу може включати усунення зайвих кроків, автоматизацію повторюваних завдань, розподіл завдань між співробітниками та використання ефективних інструментів та технологій.

Оптимізація робочого процесу має декілька переваг. По-перше, вона дозволяє знизити витрати, зменшити час, необхідний для виконання завдань, та покращити якість виконаної роботи. По-друге, оптимізований робочий процес сприяє підвищенню задоволеності співробітників, оскільки вони можуть більш ефективно виконувати свої обов'язки без непотрібного напруження і затримок. Крім того, оптимізований робочий процес сприяє покращенню взаємодії з клієнтами та партнерами, оскільки забезпечує швидку та якісну реалізацію їхніх потреб.

З постійним розвитком технологій та зростанням обсягу доступної інформації, підприємства шукають способи оптимізувати свої робочі процеси. Вони шукають ефективні способи використання інформаційних технологій для покращення продуктивності та ефективності своєї діяльності.

Автоматизація управління є одним із найпродуктивніших використань інформаційних технологій на підприємствах. Застосування програмного забезпечення та технічних засобів дозволяє підвищити ефективність управлінської діяльності. Сучасні керівники стикаються з великим обсягом інформації, необхідністю швидких рішень та ефективного комунікування з підлеглими, а також з постійним контролем фінансових, економічних та технічних процесів.

Проте, просто встановлення персональних комп'ютерів не є вирішенням цих проблем. Це лише перший крок до комплексної автоматизації підприємства. Основна мета автоматизації полягає в створенні потужного та універсального інструменту, який забезпечує швидкий аналіз діяльності

					ДТЕУ 121 07-26.БР	Аркуш
						9
Зм.	Аркуш	№ докум	Підпис	Дата		

компанії та допомагає у прийнятті ефективних рішень. Важливо, що керівник встановлює завдання та визначає стратегію, використовуючи цей інструмент.

Автоматизація управління на підприємстві є актуальним та перспективним напрямом, що дозволяє здійснити ряд важливих переваг:

- Заміна ручної праці шляхом використання спеціальних технологій та програмного забезпечення.
- Зниження загальних витрат підприємства шляхом ефективного використання ресурсів та оптимізації процесів.
- Підвищення продуктивності та ефективності діяльності організації за рахунок швидкого доступу до необхідної інформації та автоматизованого контролю над процесами.
- Спрощення виконання повсякденних та повторюваних завдань завдяки автоматичному виконанню процесів.

Правильне поєднання комп'ютерного обладнання та спеціалізованого програмного забезпечення дозволяє оптимізувати різні аспекти роботи підприємства, що призводить до збільшення його прибутковості та конкурентоспроможності.

Центральним аспектом автоматизації є не тільки встановлення сучасного обладнання, але й вирішення складних завдань шляхом застосування інноваційних підходів. В процесі вдосконалення та структурування роботи підприємства неможливо уявити його функціонування без кваліфікованих фахівців. Для успішної реалізації автоматизації часто необхідні спеціально розроблені програми, які адаптовані до конкретних потреб та особливостей підприємства.

Автоматизація є актуальним напрямом для всіх типів бізнесу, включаючи спортивні клуби. Вона дозволяє скоротити час та збільшити рівень обслуговування клієнтів, полегшує контроль за роботою працівників та

					ДТЕУ 121 07-26.БР	Аркуш
						10
Зм.	Аркуш	№ докум	Підпис	Дата		

забезпечує ефективну взаємодію між підрозділами підприємства. Для цього потрібні спеціальні програми, розроблені для конкретних потреб підприємства.

У більшості виробничих підприємств автоматизація допомагає ефективно проводити всі аспекти діяльності. В останні роки спортивні клуби стали дуже популярними, і з'явилася потреба в інформаційних системах, що дозволяють вирішувати специфічні завдання в цій галузі. Такі системи допомагають забезпечити ефективну діяльність клубів та підвищити рівень обслуговування клієнтів.

На ринку доступний широкий вибір програмних продуктів, призначених для обліку завдань, характерних для різних видів діяльності. З метою огляду було відібрано кілька інформаційних систем, які були проаналізовані за декількома критеріями, такими як функціональність, вартість використання, а також переваги і недоліки кожної системи.

Під час аналізу звернено увагу на функціональні можливості кожної системи, включаючи їх здатність виконувати потрібні завдання та надавати корисну інформацію для управління. Крім того, була врахована вартість використання кожної системи, включаючи вартість придбання, розгортання і підтримки.

У результаті аналізу були виявлені як переваги, так і недоліки кожної інформаційної системи. Наприклад, деякі системи можуть мати розширені можливості звітності і аналітики, що дозволяють краще контролювати діяльність підприємства. Однак, деякі системи можуть бути дорогими у використанні або вимагати додаткових ресурсів для налагодження і підтримки.

Загальна мета аналізу полягала в тому, щоб добитися розуміння того, яка інформаційна система найкраще відповідає потребам та обмеженням підприємства. Це дозволить зробити обґрунтований вибір і розробити

					ДТЕУ 121 07-26.БР	Аркуш
						11
Зм.	Аркуш	№ докум	Підпис	Дата		

програмний продукт, який найбільш ефективно підтримує діяльність організації.

Для аналізу було обрано декілька інформаційних систем, такі як: «SportManage», «FitPro», «ClubMaster», «SportsClub Pro», «FitBiz». Всі п'ять програмних продуктів позиціонуються як програмне забезпечення для автоматизації діяльності спортивних клубів, фітнес центрів, тренажерних залів тощо.

SportManage - це комплексна інформаційна система для управління спортивним клубом, яка надає широкий спектр функціональності, включаючи облік членства, розклади тренувань, фінансовий облік та звітність. Вона пропонує зручний інтерфейс та можливість інтеграції з онлайн-платформами.

Переваги:

- Широкий спектр функцій для управління клубом, включаючи облік членства, розклади тренувань, фінансовий облік і звітність.
- Інтеграція з онлайн-платформами для реєстрації і оплати членських внесків.
- Зручний інтерфейс користувача та зручний доступ до інформації через мобільний додаток.

Недоліки:

- Висока вартість придбання та підтримки системи.
- Обмежені можливості настройки під потреби конкретного клубу.
- Потребує навчання персоналу для повного використання функціональності системи.

FitPro - це проста в управлінні система, яка надає можливість відстежувати прогрес учасників, включаючи тренувальні плани та результати. Вона також інтегрується з фітнес-трекерами та мобільними додатками для зручного моніторингу. Розробки цієї системи наведені на рис. 1.2.

					ДТЕУ 121 07-26.БР	Аркуш
						12
Зм.	Аркуш	№ докум	Підпис	Дата		

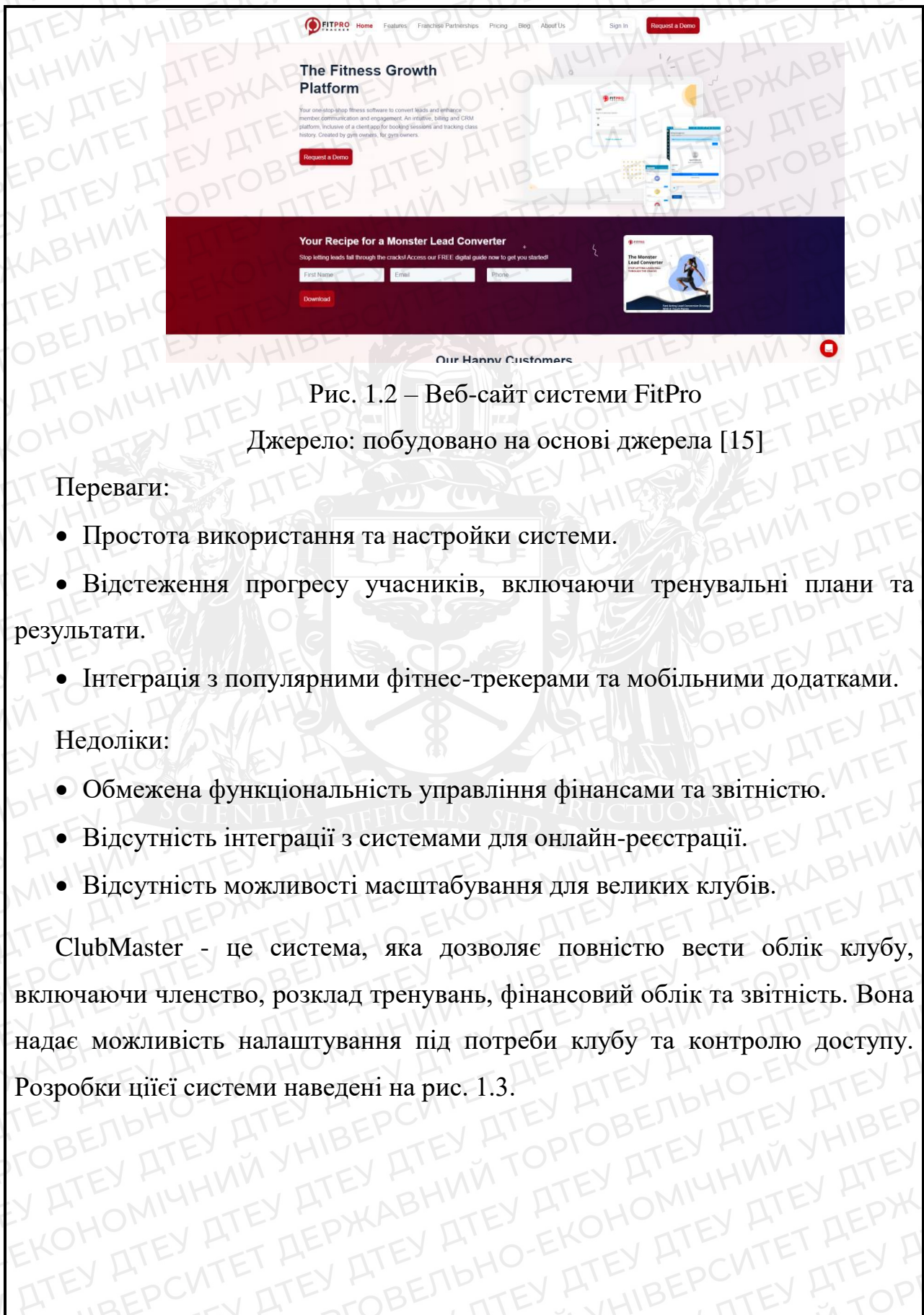


Рис. 1.2 – Веб-сайт системи FitPro

Джерело: побудовано на основі джерела [15]

Переваги:

- Простота використання та настройки системи.
- Відстеження прогресу учасників, включаючи тренувальні плани та результати.
- Інтеграція з популярними фітнес-трекерами та мобільними додатками.

Недоліки:

- Обмежена функціональність управління фінансами та звітністю.
- Відсутність інтеграції з системами для онлайн-реєстрації.
- Відсутність можливості масштабування для великих клубів.

ClubMaster - це система, яка дозволяє повністю вести облік клубу, включаючи членство, розклад тренувань, фінансовий облік та звітність. Вона надає можливість налаштування під потреби клубу та контролю доступу.

Розробки цієї системи наведені на рис. 1.3.

					ДТЕУ 121 07-26.БР	Аркуш
						13
Зм.	Аркуш	№ докум	Підпис	Дата		

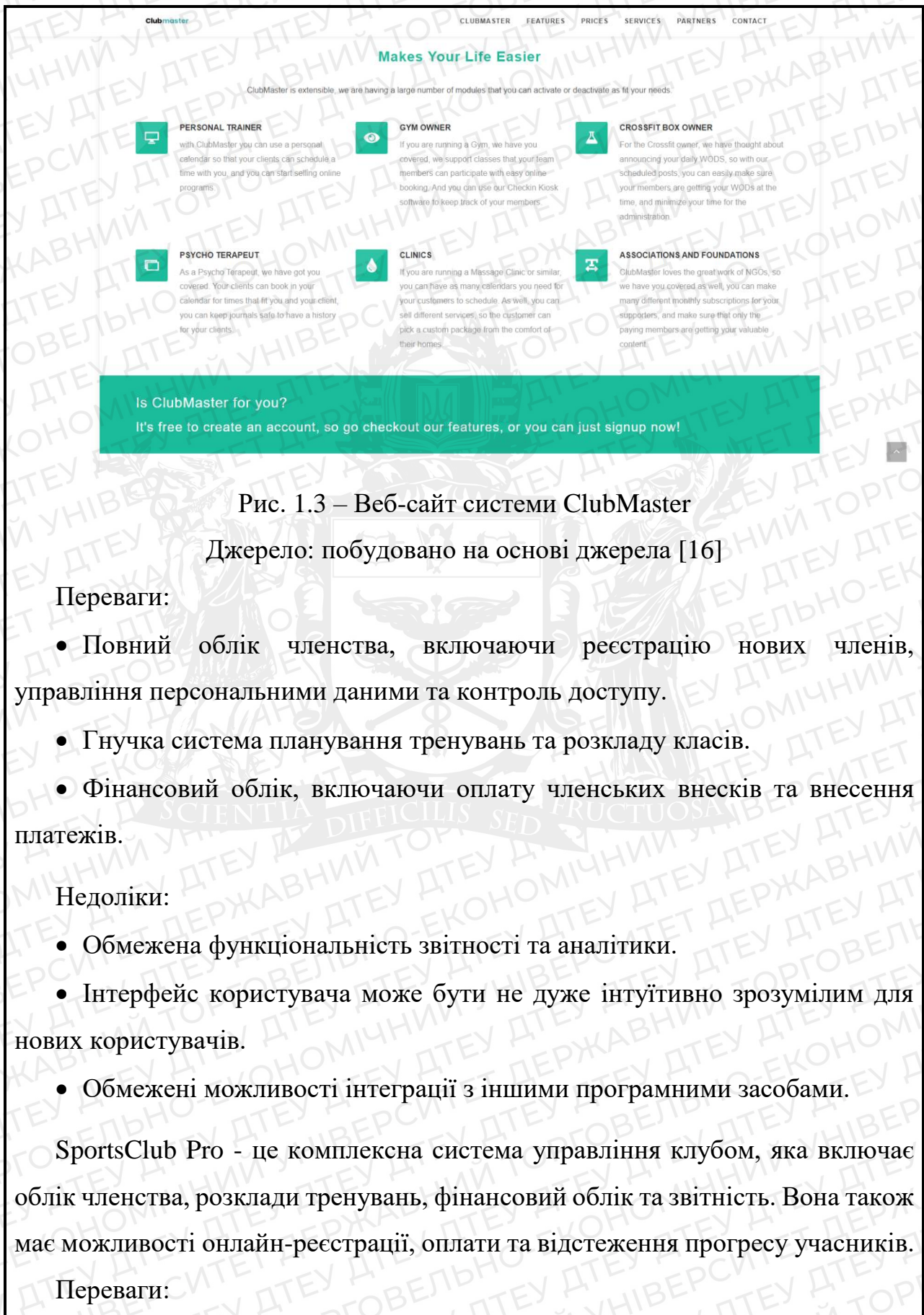


Рис. 1.3 – Веб-сайт системи ClubMaster

Джерело: побудовано на основі джерела [16]

Переваги:

- Повний облік членства, включаючи реєстрацію нових членів, управління персональними даними та контроль доступу.
- Гнучка система планування тренувань та розкладу класів.
- Фінансовий облік, включаючи оплату членських внесків та внесення платежів.

Недоліки:

- Обмежена функціональність звітності та аналітики.
- Інтерфейс користувача може бути не дуже інтуїтивно зрозумілим для нових користувачів.
- Обмежені можливості інтеграції з іншими програмними засобами.

SportsClub Pro - це комплексна система управління клубом, яка включає облік членства, розклади тренувань, фінансовий облік та звітність. Вона також має можливості онлайн-реєстрації, оплати та відстеження прогресу учасників.

Переваги:

					ДТЕУ 121 07-26.БР	Аркуш
						14
Зм.	Аркуш	№ докум	Підпис	Дата		

- Комплексна система управління клубом, включаючи облік членства, розклади тренувань, фінансовий облік і звітність.
- Можливість реалізації онлайн-реєстрації, оплати та бронювання тренувань.
- Інтеграція з системами контролю доступу та відстеженням прогресу учасників.

Недоліки:

- Висока вартість придбання та підтримки системи.
- Необхідність налагодження системи та навчання персоналу для повного використання функціоналу.
- Вимоги до технічних можливостей комп'ютерів та серверів для ефективної роботи системи.

FitBiz - це гнучка система, яка може бути настроєна під потреби конкретного клубу. Вона надає можливість ведення обліку відвідуваності, розкладу тренувань і платежів, а також інтегрується з системами контролю доступу.

Переваги:

- Повний цикл управління клубом, включаючи реєстрацію членів, тренувальні плани, фінансовий облік та звітність.
- Інтеграція з онлайн-платформами для реєстрації та оплати.
- Аналітичні інструменти для відстеження та оцінки продуктивності клубу.

Недоліки:

- Вартість використання системи може бути високою для невеликих клубів.

					ДТЕУ 121 07-26.БР	Аркуш
						15
Зм.	Аркуш	№ докум	Підпис	Дата		

- Деякі функції можуть бути складними для освоєння користувачами без попереднього досвіду.
- Обмежені можливості настройки та індивідуалізації системи.

Після огляду різноманітних програмних продуктів, що присутні на ринку, були виявлені як переваги, так і недоліки кожної з них. Проте, виявилось, що деякі з цих продуктів мають широкий спектр функціоналу, який включає можливості складського та бухгалтерського обліку, а також цінову політику, яка виглядає привабливо. Однак, зазначені характеристики не відповідають вимогам для розробки інформаційної системи спортивного залу.

У контексті спортивного залу, для якого має бути розроблений продукт, відсутність потреби в такому функціоналі, який присутній у більшості існуючих рішень, стає першочерговою. Отже, розроблена інформаційна система не має включати можливостей для бухгалтерського та податкового обліку, оскільки ці завдання виконує головний бухгалтер спортивного залу. Крім того, модуль складського обліку не є необхідним, оскільки спортивний зал не має складського простору. Також немає потреби в реалізації додаткових послуг, таких як солярій або масаж, оскільки їх в спортивному залі немає.

Отже, інформаційна система буде спрямована на задоволення специфічних потреб спортивного залу, забезпечуючи ефективне управління клієнтською базою, розкладом тренувань, контролем доступу та іншими необхідними функціями. Вона буде легкою у використанні та налаштуванні, а також відповідатиме вимогам законодавства і забезпечуватиме конфіденційність даних клієнтів.

По-друге, важливо враховувати вимоги замовника, серед яких основними є простота впровадження і мінімальні витрати на продукт. У зв'язку з цим, специфіка розробки інформаційної системи для спортивного залу не дозволяє використовувати готові рішення, і вимагає створення нового продукту, який враховуватиме всі особливості та потреби цього закладу.

					ДТЕУ 121 07-26.БР	Аркуш
						16
Зм.	Аркуш	№ докум	Підпис	Дата		

1.3. Постановка задачі

Велика частка користувачів спортивного залу використовує смартфони, що робить WEB-додаток найзручнішим варіантом доступу до системи. WEB-додатки не потребують додаткового встановлення на пристрої користувача, що спрощує доступ і зменшує бар'єри для використання.

Статистичні дані та опитування показують, що користувачі частіше користуються веб-версіями програмних застосунків. Це пов'язано зі спрощеною установкою і налаштуванням, а також з можливістю отримати доступ до додатку з будь-якого пристрою, на якому є веб-браузер.

WEB-додатки забезпечують більшу гнучкість в розробці та оновленні. Адміністратор може оновлювати інформацію на сайті безпосередньо через адміністративний інтерфейс, не потребуючи редагування вихідного коду. Це робить процес підтримки та розвитку додатку більш ефективним та швидким.

WEB-додатки можуть бути легко оптимізовані для швидкої роботи та ефективного використання ресурсів. Веб-сервери та технології кешування дозволяють зменшити навантаження на сервер та покращити швидкість відгуку додатку.

Адаптація до різних розмірів екранів, зокрема до смартфонів, є важливим фактором, оскільки це дозволяє користувачам зручно використовувати додаток незалежно від пристрою, який вони використовують.

Отже, вибір WEB-додатку для створення програмного рішення спортивного залу обґрунтований його зручністю, доступністю, ефективністю та спрощеною підтримкою.

Ціллю випускної роботи є розробка WEB-додатку для автоматизації роботи спортивного залу з метою поліпшення організації роботи з відвідувачами та спрощення процесів тренувань та адміністрування. Додаток повинен надавати

					ДТЕУ 121 07-26.БР	Аркуш
						17
Зм.	Аркуш	№ докум	Підпис	Дата		

механізм авторизації та аутентифікації користувачів, зберігати програми тренувань у цифровому форматі, дозволяти додавати, видаляти та переглядати програми тренувань, проводити валідацію введених даних, реєструвати записи на тренування в онлайн режимі, забезпечувати можливість покупки абонементів користувачами, мати приємний та зручний інтерфейс для залучення більшої кількості користувачів, дозволяти адміністратору оновлювати більшість інформації на сайті без редагування вихідного коду, бути добре оптимізованим та адаптованим до різних розмірів дисплеїв.

Вимоги до функціональності:

Механізм авторизації та аутентифікації користувача, що забезпечує безпечний доступ до системи.

Можливість створювати, зберігати, видаляти та переглядати програми тренувань у цифровому форматі.

Реалізація механізму додавання та видалення записів на тренування в онлайн режимі.

Реалізація можливості покупки абонементів користувачами.

Валідація введених даних для запобігання помилкам та забезпечення коректної роботи додатку.

Забезпечення приємного та зручного інтерфейсу для залучення користувачів і забезпечення їх зручного користування додатком.

7. Можливість адміністратору оновлювати більшість інформації на сайті без редагування вихідного коду, забезпечуючи швидке оновлення даних та зручні інструменти для адміністрування.

Добра оптимізація додатку, забезпечуючи швидку реакцію та ефективну роботу навіть при великій кількості користувачів та обсязі даних.

Адаптація до різних розмірів дисплеїв, забезпечуючи зручне відображення та користування додатком на різних пристроях, зокрема на смартфонах.

Вимоги до WEB-додатку:

					ДТЕУ 121 07-26.БР	Аркуш
						18
Зм.	Аркуш	№ докум	Підпис	Дата		

Додаток повинен бути розроблений з використанням веб-технологій, таких як HTML, CSS, JavaScript.

Додаток повинен бути сумісний з різними сучасними веб-браузерами, зокрема Google Chrome, Mozilla Firefox, Safari, Microsoft Edge.

Веб-сайт повинен бути респонсивним і адаптованим до різних розмірів екранів, забезпечуючи зручне користування на різних пристроях, включаючи смартфони.

Додаток повинен бути розроблений з урахуванням принципів безпеки і захисту даних, забезпечуючи захист персональної інформації користувачів та конфіденційність їх даних.

Потрібно використовувати зручну та інтуїтивно зрозумілу навігацію та інтерфейс, щоб користувачі могли легко зорієнтуватись і використовувати функціонал додатку.

1.4. Висновки до розділу 1

У даному розділі було проведено загальну характеристику інформаційних систем, а також були оглянуті існуючі рішення інформаційних систем, спрямованих на надання послуг власникам спортивних клубів.

В результаті аналізу існуючих рішень, було встановлено, що на ринку програмних продуктів для спортивних клубів існує певна кількість інформаційних систем, які пропонують функціонал для управління клубами, зберігання даних про клієнтів, планування тренувань, оплати членства та інші пов'язані процеси.

Проте, в ході постановки задачі для власної інформаційної системи для спортивних клубів, було визначено конкретні потреби та вимоги. Було визначено, що система повинна надавати зручний інтерфейс для власників спортивних клубів, щоб вони могли ефективно управляти клубами, керувати

					ДТЕУ 121 07-26.БР	Аркуш
						19
Зм.	Аркуш	№ докум	Підпис	Дата		

реєстрацією клієнтів, плануванням тренувань, контролювати оплати та забезпечувати звітність.

Також, вимоги до системи включають інтеграцію з іншими системами, такими як платіжні шлюзи або системи електронної комерції, для забезпечення зручності в оплаті членства та інших послуг.

Отже, розробка власної інформаційної системи для спортивних клубів дозволить вирішити конкретні потреби власників клубів і забезпечити ефективне управління клубами, зручну реєстрацію клієнтів, планування тренувань та звітність. Крім того, інтеграція з іншими системами покращить функціональність та зручність системи, спрощуючи процеси оплати та забезпечуючи більш широкі можливості для клієнтів і власників клубів.

Розробка власної інформаційної системи також дозволить покращити ефективність управління клубами шляхом автоматизації рутинних завдань, зменшення часу на адміністративні процеси і забезпечення точності та цілісності даних.

Описані вимоги до системи створюють базу для подальшої розробки і реалізації інформаційної системи для спортивних клубів. Впровадження такої системи має потенціал покращити ефективність роботи спортивних клубів, полегшити їхнє управління та забезпечити кращий досвід для клієнтів.

У подальшому дослідженні необхідно зосередитися на розробці та впровадженні інформаційної системи з урахуванням вказаних вимог. Детальніший аналіз функціональності, архітектури та інших аспектів розробки системи дозволить досягти поставлених цілей і створити ефективний інструмент для управління спортивними клубами.

					ДТЕУ 121 07-26.БР	Аркуш
						20
Зм.	Аркуш	№ докум	Підпис	Дата		

РОЗДІЛ 2

ДЕТАЛЬНИЙ АНАЛІЗ ПРЕДМЕТА ДОСЛІДЖЕННЯ

2.1. Основні положення

Для ефективної організації роботи з відвідувачами спортивного залу необхідно розробити веб-сайт, який виступатиме потужним інструментом формування позитивного іміджу організації. Кожен елемент, розміщений на сайті, має значення, оскільки впливає на сприйняття відвідувачами організації як сучасної та надійної компанії. Будучи вірогідним джерелом інформації, сайт може бути використаний для підсилення бренду та просування його в інтернет-середовищі.

Створення веб-сайту надає широкі можливості для привертання цільової аудиторії за допомогою різноманітних інструментів інтернет-маркетингу, таких як пошукова оптимізація (SEO), контекстна реклама та електронна розсилка. Це лише деякі засоби, які можна використовувати для залучення потенційних клієнтів та збільшення їх зацікавленості у співпраці з компанією.

Отже, розроблений веб-сайт буде не лише інформаційним ресурсом, але й потужним інструментом маркетингу, який допоможе підвищити увагу та інтерес потенційних клієнтів до організації.

Фронтенд частина веб-додатку відіграє важливу роль у відображенні даних та створенні інтерфейсу для користувача. Однак, для досягнення оптимальної продуктивності та швидкості завантаження, фронтенд повинен бути легким і ефективним. Це дозволяє забезпечити швидкий доступ до інформації для користувачів.

					<i>ДТЕУ 121 07-26.БР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Програмний додаток мережі спортивних клубів	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Зав. каф.		Криворучко О.В.		03.03.23		P2	21	65
Керівник		Власенко Л.О.		03.03.23		Факультет інформаційних технологій 4 курс, 7 група		
Гарант		Рзаєва С.Л.		03.03.23				
Розробив		Чеботаєв Д.О.		03.03.23	Детальний аналіз предмета дослідження			

У розробці фронтенду використовуються різноманітні технології, такі як HTML, CSS і JavaScript. Для створення сучасних інтерфейсів часто використовуються фреймворки, такі як React або Angular, які допомагають зручно та ефективно організувати роботу з компонентами та взаємодію з даними.

Кожна веб-сторінка може мати свій унікальний1 вигляд, але вони, як правило, включають загальні стандартні компоненти. Ці компоненти забезпечують зручну навігацію, відображення вмісту та взаємодію з користувачем. Використання таких стандартних компонентів допомагає забезпечити зрозумілість та зручність використання веб-додатку для користувачів.

В цілому, фронтенд частина веб-додатку відіграє ключову роль у створенні зручного та привабливого інтерфейсу для користувачів. І використання сучасних технологій та стандартних компонентів допомагає забезпечити якість та ефективність розробки. Структура веб-додатку наведена на рис. 2.1.

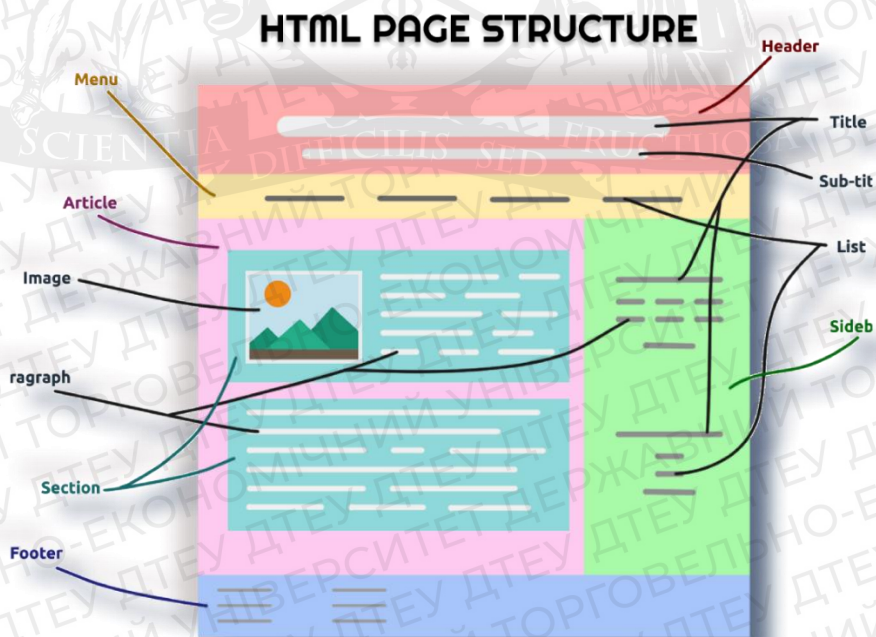


Рис. 2.1 – Загальна структура WEB-сторінки

Джерело: побудовано на основі джерела [17]

					ДТЕУ 121 07-26.БР	Аркуш
						22
Зм.	Аркуш	№ докум	Підпис	Дата		

Веб-сторінка є ключовим елементом сучасного інтернет-простору, який впливає на враження користувачів і формує їхню сприйнятливість до вмісту та послуг, що надаються. Розумна організація та структурування веб-сторінки мають велике значення для досягнення ефективності та зручності використання.

Меню навігації є розділом веб-сторінки, що містить посилання на основні секції сайту. Зазвичай це представлено у вигляді кнопок, посилань або вкладок і залишається незмінним на всіх сторінках. Важливо, щоб навігація була зрозумілою та логічною, щоб уникнути заплутання та розчарування відвідувачів.

Основний вміст веб-сторінки представляє собою значну площу в центрі, де розміщується унікальний контент: відео, історії, карти, заголовки новин та інше. Цей розділ зазвичай змінюється в залежності від конкретної сторінки, надаючи відвідувачам різноманітний та цікавий вміст.

Сайдбар або бічна панель є додатковою інформацією, посиланнями, цитатами, оголошеннями та іншим контентом. Зазвичай вона пов'язана з основним контентом (наприклад, на сторінці новинної статті бічна панель може містити інформацію про автора або посилання на інші статті), але іноді вона також може містити додаткові елементи, наприклад, систему навігації.

Нижній колонтитул - це панель, яка знаходиться внизу сторінки і зазвичай містить менш значну інформацію, таку як авторські права або контактну інформацію. Це місце для загальної інформації, хоча вона не є найважливішою частиною веб-сайту. Нижній колонтитул іноді також використовується з метою покращення пошукової оптимізації, надаючи посилання для швидкого доступу до популярного контенту.

Завдяки постійному розвитку сучасних веб-технологій, більшість веб-сайтів сьогодні використовують різноманітні інтерфейсні бібліотеки, фреймворки і плагіни, що полегшують управління HTML, CSS і JavaScript, які

					ДТЕУ 121 07-26.БР	Аркуш
						23
Зм.	Аркуш	№ докум	Підпис	Дата		

відображаються в браузері. Загалом, ці інструменти можуть бути розподілені на три основні категорії.

Перша категорія - це бібліотеки JavaScript, які надають широкий спектр функціональних можливостей і підходів. Серед найпопулярніших бібліотек варто згадати jQuery, Angular і React.

Друга категорія - це адаптивні бібліотеки, які отримали популярність завдяки зростанню використання мобільних пристроїв. Ці фреймворки надають готові компоненти і адаптивні ґрид-системи, що сприяють створенню зручного веб-сайту для мобільних пристроїв. Два найбільш популярних фреймворки в цій категорії - це Bootstrap і Foundation.

Третя категорія - це плагіни, які додають додаткову функціональність до веб-сайту. Ці плагіни можуть включати різноманітні функції, такі як анімація, галереї зображень, слайдери тощо. Вони допомагають покращити взаємодію з користувачем і забезпечують більш привабливий та функціональний веб-сайт.

Завдяки використанню інтерфейсних бібліотек, фреймворків і плагінів розробка веб-сайтів стає значно простішою, а результатом є привабливий та функціональний інтерфейс для користувачів. Додаткова категорія інструментів - плагіни, включає безліч корисних віджетів користувацького інтерфейсу, таких як слайдери, діаграми, сітки даних, датчики, анімація та елементи управління введенням. Відповідно до потреб проекту, ці плагіни можуть бути використані на всіх сторінках веб-сайту або обмежитися лише декількома сторінками.

Хоча зазвичай сучасні веб-сайти використовують одну основну бібліотеку JavaScript і один адаптивний фреймворк, вони часто мають багато плагінів на основі JavaScript. Для великих або дуже інтерактивних сайтів кількість плагінів може сягати декількох десятків. Ці плагіни розширюють можливості веб-сайту і забезпечують більш багатий та динамічний користувацький досвід.

					ДТЕУ 121 07-26.БР	Аркуш
						24
Зм.	Аркуш	№ докум	Підпис	Дата		

Використання цих інтерфейсних інструментів значно полегшує розробку веб-сайтів, дозволяючи розробникам швидко і ефективно реалізовувати різноманітні функції та ефективно керувати виглядом та поведінкою веб-сторінок. В результаті отримується унікальний та інноваційний веб-інтерфейс, який привертає увагу користувачів та забезпечує їм зручне взаємодію зі сторінкою.

Бекенд та фронтенд тісно пов'язані між собою і спільно створюють повноцінний веб-додаток. Фронтенд відповідає за візуальне представлення додатку, взаємодію з користувачем та передачу даних до бекенду. Він використовує API, надане бекендом, для отримання необхідної інформації та виконання дій.

Бекенд, з свого боку, обробляє запити від фронтенду, виконує бізнес-логіку, здійснює доступ до бази даних, забезпечує безпеку та автентифікацію користувачів. Він зберігає та обробляє дані, необхідні для функціонування додатку, і повертає фронтенду результати, які потім відображаються користувачеві.

Існує багато інструментів та технологій, які використовуються для розробки бекенду. Основні з них наведені на рис. 2.2. Деякі з популярних включають:

- Мови програмування: Наприклад, Python, Java, PHP, Ruby, JavaScript (Node.js) використовуються для написання серверної логіки.
- Фреймворки: Наприклад, Django, Flask, Spring, Laravel, Express.js, які надають готові рішення та інструменти для розробки бекенду.
- Бази даних: Наприклад, MySQL, PostgreSQL, MongoDB, Redis, які використовуються для зберігання та управління даними.
- Інструменти для розгортання: Наприклад, Docker, Kubernetes, Apache, Nginx, які допомагають забезпечити розгортання та масштабування бекенду.

					ДТЕУ 121 07-26.БР	Аркуш
						25
Зм.	Аркуш	№ докум	Підпис	Дата		

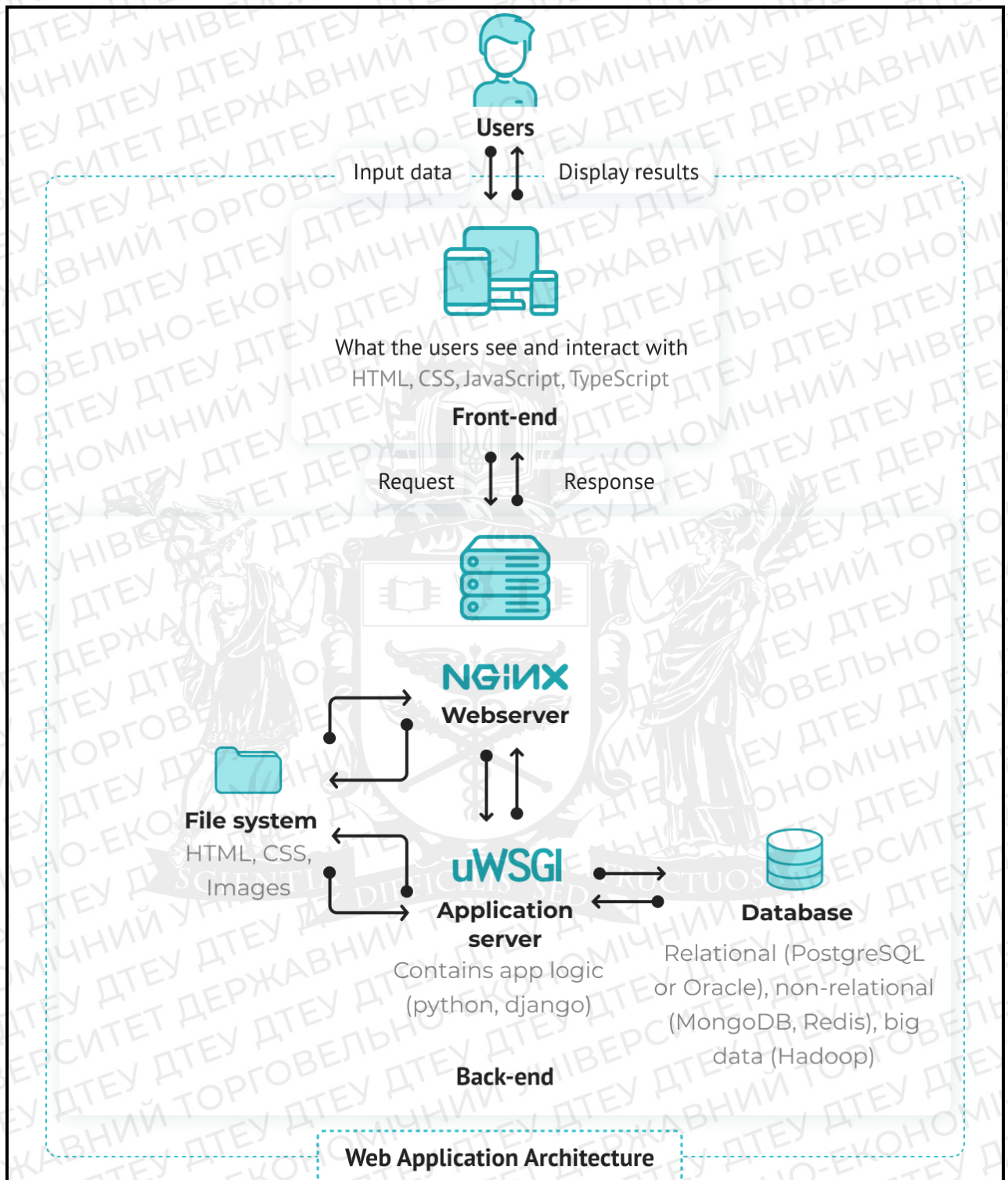


Рис. 2.2 – Архітектурна схема WEB – застосунку

Джерело: побудовано на основі джерела [18]

Бекенд є важливим компонентом додатку, який дозволяє забезпечити його функціональність, безпеку та ефективність. Бекенд частина додатку є

					ДТЕУ 121 07-26.БР	Аркуш
						26
Зм.	Аркуш	№ докум	Підпис	Дата		

важливою складовою веб-додатків і відповідає за обробку та збереження даних, бізнес-логіку, безпеку, аутентифікацію, авторизацію та взаємодію з базою даних. Основні принципи розробки бекенд частини включають:

Розділення обов'язків: Бекенд відповідає за логіку додатку, тоді як фронтенд займається візуальним представленням та взаємодіє з користувачем. Цей принцип розділення дозволяє покращити модульність, підтримку та розширюваність системи.

API-орієнтованість: Бекенд часто надає API (Application Programming Interface), яке дозволяє фронтенду та іншим зовнішнім системам взаємодіяти з ним. Це може бути реалізовано через REST (Representational State Transfer) або GraphQL, що дозволяє передавати та отримувати дані між фронтендом та бекендом.

Безпека: Бекенд відповідає за захист даних та виконання автентифікації та авторизації користувачів. Він повинен мати механізми для перевірки доступу до ресурсів, шифрування даних та захисту від зловживань.

Управління базою даних: Бекенд забезпечує збереження та взаємодію з базою даних. Це може включати створення схеми бази даних, виконання запитів, маніпуляцію даними та забезпечення цілісності даних.

Масштабованість: Бекенд повинен бути готовим до масштабування, щоб забезпечити ефективну роботу додатку навіть при зростанні обсягів даних та навантаження.

Після того, як було досліджено декілька десятків сайтів спортивних клубів, було виявлено з проблему відсутності необхідного функціоналу на більшості з них. Майже на кожному сайті не можна було знайти всіх функцій, які б хотілось мати. Наприклад, було б корисно мати можливість бронювання занять онлайн та перегляду доступних тренерів і послуг, але не всі сайти це пропонували.

					ДТЕУ 121 07-26.БР	Аркуш
						27
Зм.	Аркуш	№ докум	Підпис	Дата		

Ця проблема може бути особливо дратівливою для клієнтів, які хочуть бути в курсі всього, що відбувається в спортивному клубі. Відсутність необхідного функціоналу може стати перешкодою для спілкування з тренерами, планування занять та контролювання власного прогресу.

Тому, щоб вирішити цю проблему, потрібно створити свою систему, в якій потрібно реалізувати всі задумані функції, які допоможуть клієнтам та співробітникам клубів ефективно спілкуватися та планувати свої заняття. Розробка нової системи дозволить зробити все, що хотілось зробити на інших сайтах. До того ж, додавання зручного для користувача функціоналу дозволяє підвищити конкурентоспроможність на ринку.

Отже, висновок попереднього повідомлення полягає в тому, що відсутність необхідного функціоналу на багатьох сайтах спортивних клубів є серйозною проблемою, яка може вплинути на задоволення клієнтів та ефективність роботи клубів. Розробка власної системи дозволить мені реалізувати всі задумані функції, які не були доступні на інших сайтах. Таким чином розробка власної системи є найкращим виходом з даної ситуації.

2.2. Аналіз функціональних вимог

Основні функціональні вимоги до розробки програмного продукту для мережі спортивних клубів можуть включати наступне:

- Реєстрація клієнтів: можливість реєструватись в програмі для користувачів клубу з метою використання його послуг.
- Управління абонементами: можливість купувати та користуватись абонементами, перевіряти строк їх дії та продовжувати їх.
- Управління розкладом: можливість переглядати розклад занять клубу, записуватись на заняття та скасовувати раніше зроблені записи.

					ДТЕУ 121 07-26.БР	Аркуш
						28
Зм.	Аркуш	№ докум	Підпис	Дата		

- **Управління фінансами:** можливість вносити оплату за послуги клубу, перевіряти історію оплати та отримувати рахунки.
- **Управління тренерським складом:** можливість переглядати інформацію про тренерів клубу, записуватись на заняття з певним тренером та переглядати історію занять.
- **Онлайн-підтримка:** можливість спілкування клієнтів з адміністраторами та тренерами клубу у режимі реального часу.

Це лише деякі з основних функціональних вимог до програмного продукту для мережі спортивних клубів. Конкретний перелік може бути розширений або змінений в залежності від потреб та вимог користувачів.

2.3. Специфікація вимог до програмного продукту

У системі є два актори: адміністратор і клієнт. Згідно їхніх ролей система буде надавати їм відповідний функціонал. Функціональні можливості продукту будуть доступні для клієнта та адміністратора. Серед них основними є: керування користувачами, моніторинг системи, взаємодія з контентом, персоналізація та взаємодія з системою.

Складемо таблицю специфікації функціональних вимог для програмного продукту.

					ДТЕУ 121 07-26.БР	Аркуш
						29
Зм.	Аркуш	№ докум	Підпис	Дата		

Таблиця 2.1

СПЕЦИФІКАЦІЯ ФУНКЦІОНАЛЬНИХ ВИМОГ

Номер вимоги	Назва вимоги	Атрибути вимог	
		Пріоритет	Складність
1	Авторизація користувача	Обов'язково	Середня
2	Додавання послуги адміністратором	Обов'язково	Висока
3	Редагування послуги	Обов'язково	Середня
4	Запис на послугу	Обов'язково	Середня
5	Зворотній зв'язок	Не обов'язково, можна лишити контакти у вигляді номеру телефону	Висока
6	Видалення послуги	Обов'язково	Низька
7	Розподіл послуг на групи	Обов'язково	Висока
8	Особистий кабінет користувача	Обов'язково	Висока
9	Оплата послуги	Обов'язково	Висока
10	Пагінація	Не обов'язково у разі малої кількості послуг	Висока

Джерело: побудовано автором

За розробкою програмного продукту для мережі спортивних клубів було сформовано низку обов'язкових функціональних вимог, включаючи авторизацію, додавання, редагування та видалення послуг, зворотній зв'язок,

						ДТЕУ 121 07-26.БР	Аркуш 30
Зм.	Аркуш	№ докум	Підпис	Дата			

розподіл послуг на групи, оплата послуг та особистий кабінет. Ці функції є ключовими для забезпечення зручного та ефективного використання веб-сайту клієнтами тренажерного залу.

Проте функція пагінації не є обов'язковою, оскільки не є критично необхідною для роботи програмного продукту. Наявність або відсутність цієї функції залежить від конкретних потреб до продукту. Отже, розробка програмного продукту, що задовольняє всі обов'язкові функціональні вимоги, є критично важливою для забезпечення задоволення користувачів та ефективної роботи бізнесу.

2.4. Специфікація не функціональних вимог

Окрім функціональних вимог, важливим аспектом розробки веб-сайту є визначення нефункціональних вимог. Нефункціональні вимоги - це вимоги, які визначають якість та характеристики додатка, а не його функціональність.

Ось деякі можливі нефункціональні вимоги для веб-сайту спортивного клубу:

Користувальницький інтерфейс (UI): вимога до користувальницького інтерфейсу полягає в тому, що веб-сайт повинен мати привабливий та зрозумілий інтерфейс, який привертає увагу користувача, забезпечує зручність навігації та легкість використання додатка.

Відповідність вимогам безпеки: вимога до безпеки полягає в тому, що веб-сайт повинен бути захищений від несанкціонованого доступу до даних користувачів, а також від зловживань з боку користувачів.

Швидкодія та продуктивність: вимога до швидкодії та продуктивності полягає в тому, що веб-сайт повинен працювати швидко та ефективно, щоб користувачі не відчували затримок та неполадок під час використання додатка.

					ДТЕУ 121 07-26.БР	Аркуш
						31
Зм.	Аркуш	№ докум	Підпис	Дата		

Масштабованість: вимога до масштабованості полягає в тому, що веб-сайт повинен бути легко масштабований та готовий до збільшення обсягу роботи при зростанні кількості користувачів.

Надійність: вимога до надійності полягає в тому, що веб-сайт повинен бути стійким до збоїв та аварій та відновлювати свою роботу швидко у разі виникнення проблем.

Легкість супроводу та розвитку: вимога до легкості супроводу та розвитку полягає в тому, що веб-сайт повинен бути легко змінюватись та доповнюватись новими функціями, а також підтримувати документацію та допоміжні інструменти для розробки та тестування.

2.5. Технічне завдання

Структура технічного завдання:

1. Загальні відомості:
 - 1.1 Найменування системи: Програмний додаток мережі спортивних клубів.
 - 1.2 Планові терміни початку та закінчення робіт: початок робіт планується на 28.01.2023, а реліз проєкту на 28.05.2023.
 - 1.3 Порядок оформлення і пред'явлення результатів робіт: вказано в календарному плані виконання проєкту.
 - 1.4 Потенційні користувачі системи: приватні підприємці, які мають спортивний клуб, або мережу спортивних клубів, а також потенційні відвідувачі спортивних клубів.
2. Мета та призначення системи:
 - 2.1 Призначення системи: призначення програмного продукту полягає в спрощенні роботи менеджерів з обслуговування клієнтів. Також система додає приховану рекламну кампанію для спортивного комплексу,

					ДТЕУ 121 07-26.БР	Аркуш
						32
Зм.	Аркуш	№ докум	Підпис	Дата		

оскільки інформація про систему стає загальнодоступною, і буде видаватись користувачам через пошукову систему при відповідному запиті.

2.2 Метою створення системи є розробка швидкого, інтуїтивно зрозумілого веб орієнтованого застосунку для зручної взаємодії користувача з послугами спортивного клубу.

3. Вимоги до системи:

3.1 Вимоги до структури та функціонування системи, перелік підсистем: програмний модуль має включати в себе серверну частину (backend), клієнтську частину (frontend), базу даних.

3.2 Вимоги до режимів функціонування системи: для коректної роботи системи необхідне підключення до мережі інтернет.

3.3 Вимоги до діагностування системи: всі дані мають зберігатися до бази даних.

3.4 Вимоги до режимів управління системою: буде розроблено два види доступу, такі як «Адміністратор» і «Користувач».

3.5 Вимоги до програмного забезпечення: програмний модуль має працювати стабільно на будь якій операційній системі, яка підтримує графічні інтерфейси та доступ до мережі інтернет.

3.6 Вимоги до технічного забезпечення: оскільки програмний модуль є веб орієтованим застосунком, то працюватиме на будь якому пристрої, який має екран для відображення інформації та вихід до мережі інтернет.

3.7 Вимоги до методичного забезпечення: інтуїтивно зрозумілий інтерфейс користувача, з підказками, в разі некоректної взаємодії з системою.

2.6. Висновки до розділу 2

У даному розділі було розглянуто основні положення, пов'язані з аналізом вимог та проектування програмного продукту для мережі спортивних клубів.

					ДТЕУ 121 07-26.БР	Аркуш
						33
Зм.	Аркуш	№ докум	Підпис	Дата		

Було проведено аналіз функціональних вимог, що визначають необхідний функціонал програмного продукту. Вимоги включають такі функції, як управління клієнтами, реєстрація, планування тренувань, оплата членства та інші процеси, пов'язані з управлінням спортивними клубами.

Специфікація до програмного продукту була розроблена для уточнення вимог та опису конкретних можливостей, які має надавати система. Це допомагає уточнити функціональність та обсяг робіт, необхідних для реалізації програмного продукту.

Також було розроблено специфікацію нефункціональних вимог, яка визначає технічні та проектні обмеження, такі як продуктивність, надійність, безпека, масштабованість та інші аспекти системи. Це дозволяє забезпечити високу якість та ефективність програмного продукту.

У рамках проектування була спроектована база даних, яка включає сутності, атрибути та зв'язки, необхідні для зберігання і обробки даних системи. Проектування бази даних відповідає вимогам функціональності та дозволяє ефективно зберігати та керувати даними клієнтів, тренувань, оплати та інших аспектів спортивного клубу.

У підсумку, проведений аналіз вимог та проектування вказує на те, що розробка програмного продукту для мережі спортивних клубів має великий потенціал для поліпшення управління та надання послуг власникам спортивних клубів. Аналіз функціональних та нефункціональних вимог дав чітке уявлення про необхідні функції та параметри, які повинна мати інформаційна система.

Специфікація до програмного продукту визначила деталізований набір функціональності, що включає управління клієнтами, реєстрацію, планування тренувань, оплату членства та інші важливі функції для спортивних клубів. Це створило чітку основу для розробки програмного продукту, який відповідатиме потребам клієнтів і забезпечить ефективне управління клубами.

					ДТЕУ 121 07-26.БР	Аркуш
						34
Зм.	Аркуш	№ докум	Підпис	Дата		

Проектування бази даних було проведено з урахуванням потреб системи в зберіганні та обробці даних. Структура бази даних дозволяє зручно зберігати і організовувати інформацію про клієнтів, тренування, оплату та інші важливі дані для спортивних клубів.

В результаті проведеного аналізу та проектування була створена основа для подальшої розробки інформаційної системи для спортивних клубів. Розробка та впровадження такої системи має потенціал поліпшити управління клубами, спростити адміністративні процеси і підвищити задоволення клієнтів.

У подальшому дослідженні необхідно зосередитися на реалізації розробленої специфікації та впровадженні інформаційної системи в спортивні клуби. Очікується, що така система принесе значні користі власникам клубів та їх клієнтам, поліпшить ефективність та якість надання послуг.

					ДТЕУ 121 07-26.БР	Аркуш
						35
Зм.	Аркуш	№ докум	Підпис	Дата		

РОЗДІЛ 3

РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1. Вибір мови програмування та додаткових фреймворків

Вибір мови програмування для розробки веб-сервісу залежить від багатьох факторів, таких як потреби проекту, досвід, популярність мови та інші технічні фактори. Проаналізувавши технічні вимоги до майбутньої системи та всі за і проти, вибір випав на мову програмування Java.

- Java - це мова програмування, яка має декілька переваг для розробки веб-сервісів порівняно з іншими мовами, такими як:
 - Кросплатформенність: Java може працювати на будь-якій операційній системі, що дозволяє розробникам розробляти веб-сервіси для будь-якої платформи.
 - Надійність та безпека: Java має вбудовану систему обробки помилок, що дозволяє зменшити кількість помилок в програмному коді. Крім того, Java має вбудовану систему безпеки, яка дозволяє забезпечити захист веб-сервісу від різних видів атак.
 - Об'єктно-орієнтованість: Java є повністю об'єктно-орієнтованою мовою програмування, що дозволяє розробникам створювати класи та об'єкти, що полегшує розробку та підтримку веб-сервісу.
 - Велика спільнота: Java має велику спільноту розробників, що дозволяє розробникам отримувати допомогу та підтримку від інших розробників, які працюють з цією мовою.

Зм.	Аркуш	№ докум.	Підпис	Дата	<i>ДТЕУ 121 07-26.БР</i>			
Зав. каф.		Криворучко О.В.		14.04.23	Програмний додаток мережі спортивних клубів	Стадія	Аркуш	Аркушів
Керівник		Власенко Л.О.		14.04.23		РЗ	37	65
Гарант		Рзаєва С.Л.		14.04.23		Факультет інформаційних технологій 4 курс, 7 група		
Розробив		Чеботаєв Д.О.		14.04.23				
					<i>Реалізація програмного забезпечення</i>			

На відміну від інших мов програмування, Java має достатньо високий рівень абстракції, що дозволяє розробникам сконцентруватися на розв'язанні проблеми, а не на реалізації деталей. Крім того, Java має багато вбудованих функцій та бібліотек, що дозволяє розробникам створювати застосунки в першу чергу просто та швидко, оскільки мова програмування Java в першу чергу піклується саме про людей, які нею користуються.

Оскільки в сучасній розробці рідко обмежуються однією чистою мовою програмування, потрібно обрати фреймворк, який допоможе скоротити час розробки програмного забезпечення.

Існує багато фреймворків на Java для розробки веб-додатків, і кожен з них має свої переваги та недоліки. Проте, Spring Boot - один з найбільш популярних та широко використовуваних фреймворків на Java, особливо для розробки веб-додатків.

Ось деякі переваги Spring Boot порівняно з іншими фреймворками на Java:

1. Spring Boot має більш широкий спектр функціональності та підтримки, ніж інші фреймворки. Він має багато модулів та бібліотек, що дозволяє розробникам легко додавати нові функції та інтегрувати їх з існуючим кодом.
2. Spring Boot має просту та зрозумілу структуру, що дозволяє розробникам швидко розібратися з фреймворком та створити веб-додаток за короткий час.
3. Spring Boot має велику та активну спільноту розробників, яка допомагає розробникам вирішувати проблеми та надає документацію та приклади коду.
4. Spring Boot підтримує масштабування веб-додатків, що дозволяє розробникам легко додавати нові сервери та збільшувати продуктивність додатка.
5. Spring Boot має вбудовану підтримку тестування, що дозволяє розробникам швидко та ефективно тестувати свій код.

						ДТЕУ 121 07-26.БР	Аркуш 38
Зм.	Аркуш	№ докум	Підпис	Дата			

У даному випадку, розробляється програмний додаток для спортивного клубу, тому це може вимагати великої кількості ресурсів та динамічної обробки даних. Одним з кращих варіантів для розробки бекенду веб-додатку є використання мови програмування Java та фреймворку Spring Boot.

Java - це популярна мова програмування, яка має велику спільноту розробників та широкий спектр бібліотек та інструментів. Java є стабільною та безпечною мовою програмування, що є важливим для розробки додатків, які потребують високої продуктивності та надійності.

Spring Boot - це фреймворк, який спрощує розробку веб-додатків на базі Spring Framework. Spring Boot дозволяє розробникам швидко та легко створювати високоякісні веб-додатки з використанням інструментів, які допомагають збільшувати продуктивність та швидкість розробки.

Однією з головних переваг Spring Boot є автоматична конфігурація, яка дозволяє розробникам швидко створювати прості та складні веб-додатки, не затрачаючи багато часу на ручну конфігурацію. Spring Boot також має велику спільноту розробників та детальну документацію, що дозволяє легко вирішувати будь-які проблеми, які можуть виникнути під час розробки.

3.2. Вибір реляційної бази даних

До аналізу бази даних було залучено такі системи керування базами даних: MS SQL Server, MySQL, PostgreSQL. MySQL, Microsoft SQL Server та PostgreSQL - це всі реляційні бази даних, які мають схожий базовий функціонал, але мають деякі відмінності.

Однією з переваг MySQL є його простота в налаштуванні та використанні, що робить його досить популярним серед розробників. Крім того, MySQL є відкритим продуктом і може бути використаний безкоштовно, що робить його більш доступним для невеликих проектів або підприємств з обмеженим

					ДТЕУ 121 07-26.БР	Аркуш
						39
Зм.	Аркуш	№ докум	Підпис	Дата		

бюджетом. Нарешті, MySQL має широку спільноту користувачів, яка надає багато підтримки та допомоги.

У порівнянні з Microsoft SQL Server, MySQL може бути менш потужним і менш масштабованим для деяких великих проєктів, особливо коли йдеться про обробку дуже великих обсягів даних. Однак, враховуючи те, що веб-сервіс тренажерного залу не потребує роботи з дуже великими обсягами даних, MySQL є досить відповідним варіантом для реалізації цього проєкту.

Щодо порівняння з PostgreSQL, MySQL має дещо більш простий синтаксис та менш потужний набір функцій. Однак, як і MySQL, PostgreSQL є відкритим продуктом та може бути використаний безкоштовно. Також PostgreSQL має деякі функції, які можуть бути корисними для веб-сервісу, такі як вбудована підтримка JSON та геопросторових запитів. Однак, у зв'язку з тим, що MySQL є більш популярним та простим використовувати, він є більш підходящим варіантом для розробки веб-сервісу мережі спортивних клубів.

3.3. Проектування бази даних

У сучасному світі програмні веб-застосунки використовуються для забезпечення різноманітних функцій та послуг, включаючи управління даними. Важливою складовою будь-якого програмного веб-застосунку є база даних, яка дозволяє зберігати, організувати та управляти інформацією, необхідною для його функціонування. Основні з них наведені на рис. 3.1.

					ДТЕУ 121 07-26.БР	Аркуш
						40
Зм.	Аркуш	№ докум	Підпис	Дата		

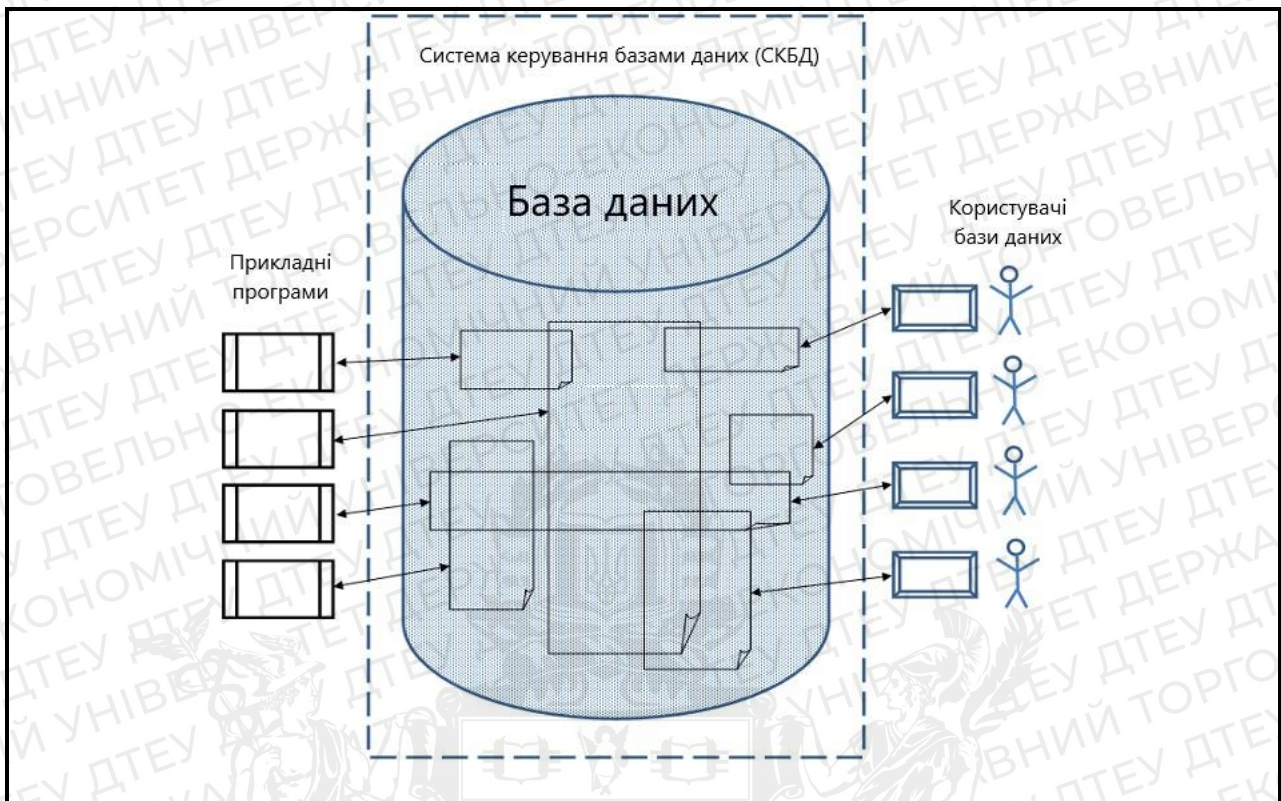


Рис. 3.1 – Схема системи керування базами даних (СКБД)

Джерело: побудовано на основі джерела [19]

У даному розділі роботи розглянемо процес проектування бази даних для програмного веб-застосунку "Спортивний клуб". Наша мета - створити структуровану та ефективну базу даних, яка дозволить зберігати і управляти інформацією про користувачів, ролі, тренерів, обладнання, вправи, заняття з тренерами, плани тренувань та записи на тренування.

Проектування бази даних включає в себе аналіз вимог до системи, ідентифікацію сутностей, визначення взаємозв'язків між сутностями, вибір відповідних атрибутів та визначення їх типів даних. Також враховується ефективність збереження та операцій з даними для досягнення швидкості та надійності роботи веб-застосунку.

У процесі проектування бази даних для "Спортивного клубу" будуть враховані особливості функціональності веб-застосунку. Наприклад, оскільки користувачі мають розділятися на певні ролі (користувачі, адміністратори,

					ДТЕУ 121 07-26.БР	Аркуш
						41
Зм.	Аркуш	№ докум	Підпис	Дата		

тренери), тому буде введена додаткова таблиця для зберігання ролей користувачів та встановлення їх відповідності.

На цьому етапі будемо детально розглядати кожен крок, необхідний для створення бази даних, включаючи визначення сутностей, атрибутів, зв'язків між ними та встановлення правильних типів даних.

Перед початком створення бази даних потрібно визначити основні сутності, які будуть зберігатись у системі. Наприклад, у випадку спортивного клубу можуть бути такими сутностями: користувачі, тренери, обладнання, вправи, заняття та інші.

Для кожної сутності потрібно визначити характеристики, які будуть зберігатись у базі даних. Наприклад, для сутності "користувачі" можуть бути атрибути, такі як ім'я, прізвище, електронна пошта, пароль та інші. Важливо врахувати особливості кожної сутності та їхні зв'язки між собою.

Після визначення сутностей та атрибутів необхідно встановити зв'язки між ними. Наприклад, у випадку спортивного клубу може існувати зв'язок між сутностями "користувачі" та "тренери", де один користувач може мати багато тренерів або навпаки. Також можуть існувати зв'язки з іншими сутностями, наприклад, "заняття" може бути пов'язане з "тренерами" та "вправами".

Кожен атрибут має свій тип даних, який визначає, яку інформацію можна зберігати в цьому атрибуті. Наприклад, для атрибута "ім'я" та "прізвище" можна використовувати тип даних "рядок" (string), для атрибута "електронна пошта" - тип "рядок", а для атрибута "пароль" - тип "рядок" або "хеш" (hash), що забезпечує безпеку даних. Для кожного атрибута потрібно обрати правильний тип даних, щоб забезпечити точність, ефективність та цілісність бази даних.

Після визначення сутностей, атрибутів, зв'язків та типів даних можна створити таблицю для кожної сутності. У випадку першої таблиці, "користувачі", створимо таблицю з відповідними стовпцями, такими як "ід",

					ДТЕУ 121 07-26.БР	Аркуш
						42
Зм.	Аркуш	№ докум	Підпис	Дата		

"емейл", "підключення розсилки", "ім'я", "прізвище", "пароль" та "логін". Кожен стовпець таблиці буде мати відповідний тип даних, визначений на попередньому кроці.

В базі даних необхідно встановити первинний ключ, який унікально ідентифікує кожен запис у таблиці. Наприклад, стовпець "ід" може бути встановлений як первинний ключ для таблиці "користувачі". Також можна встановити обмеження, наприклад, обмеження на нульові значення або обмеження на унікальність деяких атрибутів, якщо це необхідно.

Для забезпечення ефективності та надійності бази даних важливо провести оптимізацію та нормалізацію. Нормалізація допомагає уникнути дублювання даних та забезпечує структурованість бази даних. Вона розбиває таблиці на менші, декомпонентні атрибути у відповідні сутності та встановлюючи зв'язки між ними. Наприклад, якщо виявляється, що інформація про тренерів також міститься в таблиці "користувачі", можна розбити цю інформацію в окрему таблицю "тренери" та встановити зв'язок між цими таблицями.

Індекси використовуються для прискорення пошуку та доступу до даних. При проектуванні бази даних можна встановити індекси на поля, які часто використовуються у пошуках або сортуваннях, наприклад, на поле "електронна пошта" або "логін" у таблиці "користувачі". Це дозволить зменшити час, необхідний для виконання запитів до бази даних.

Зовнішні ключі встановлюють зв'язки між таблицями в базі даних. Наприклад, в таблиці "заняття" може бути стовпець, який посилається на первинний ключ таблиці "тренери", щоб вказати, який тренер проводить дане заняття. Це допомагає підтримувати цілісність даних та забезпечує зв'язаність між таблицями.

У веб-застосунку важливо встановити права доступу до бази даних, щоб забезпечити безпеку і захист від несанкціонованого доступу до даних. Це можна зробити шляхом визначення різних рівнів доступу для різних ролей,

					ДТЕУ 121 07-26.БР	Аркуш
						43
Зм.	Аркуш	№ докум	Підпис	Дата		

наприклад, адміністратора, тренера та користувача. Права доступу можуть включати можливість перегляду, редагування або видалення даних залежно від ролі користувача.

Для забезпечення захисту даних важливо регулярно створювати резервні копії бази даних та мати механізм відновлення даних в разі виникнення проблем. Це можна зробити шляхом планування регулярних резервних копій та зберігання їх на надійних зовнішніх носіях або в хмарних сховищах.

Після створення бази даних важливо провести тестування та оптимізацію, щоб переконатися, що вона працює ефективно та надійно. Це включає виконання різних запитів, перевірку швидкості виконання та оптимізацію структури бази даних, індексів та запитів, якщо це потрібно.

Для більшої захищеності, також варто забезпечити шифрування даних, особливо якщо вони містять конфіденційну інформацію. Використовуючи високоякісне шифрування, можна знизити ризик небажаного доступу до ваших даних.

Безпека системи повинна бути перевірена регулярно з допомогою аудиту безпеки, який включає перевірку виконання всіх протоколів захисту і оновлень. Це гарантує, що захист даних залишається актуальним у змінних умовах.

Ретельно проаналізувавши предметну область дослідження та функціональні вимоги до системи, було розроблено наступну базу даних, яка відображає необхідну інформацію. Таблиці та їх відношення можна побачити на рис. 3.2.

					ДТЕУ 121 07-26.БР	Аркуш
						44
Зм.	Аркуш	№ докум	Підпис	Дата		

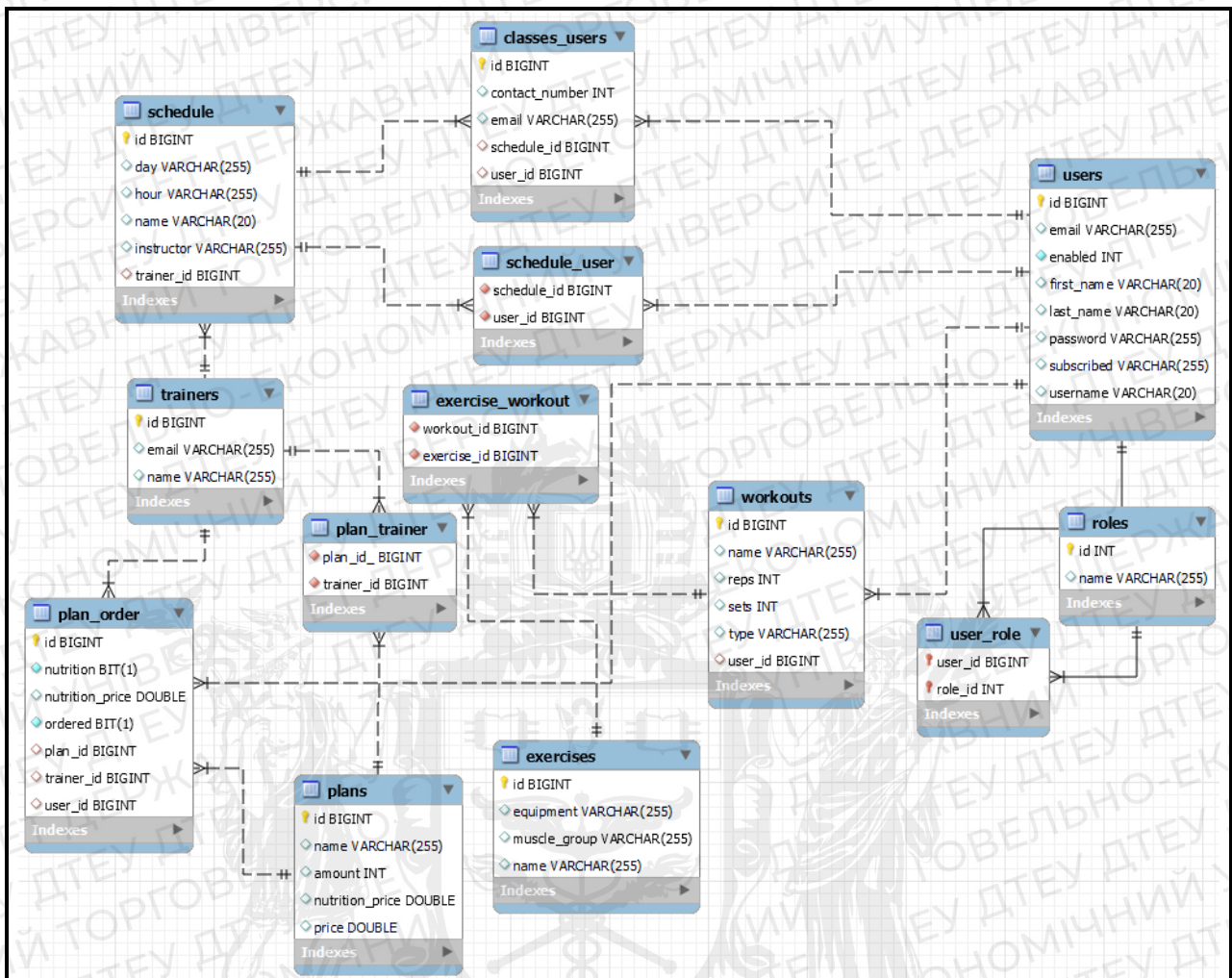


Рис. 3.2 – Фізична модель реляційної бази даних

Джерело: побудовано автором

3.4. Розробка серверної частини системи

Серверна частина відповідає за обробку бізнес-логіки та виконання різноманітних операцій над даними. Наприклад, це може бути створення, оновлення, видалення користувачів, обробка запитів на запис на тренування, генерація планів тренувань тощо. Для цього можна використовувати фреймворк Spring, який надає багато готових інструментів для реалізації бізнес-логіки, таких як інверсія керування (Inversion of Control - IoC) та управління транзакціями.

						Аркуш
					ДТЕУ 121 07-26.БР	45
Зм.	Аркуш	№ докум	Підпис	Дата		

Процес створення серверної частини інформаційної системи спортивного клубу з використанням мови програмування Java, фреймворків Spring Boot та Hibernate був розділений на кілька етапів.

1) Налаштування проекту та створення базової структури:

Починаємо зі створення нового проекту в середовищі розробки, налаштування залежностей для використання Spring Boot та Hibernate. Після цього створюємо основні пакети та класи для організації логіки проекту. Після створення проекту відразу потрібно провести певні налаштування безпеки. У фреймворку Spring Boot є спеціальні класи для цього. Діаграма security класів наведена на рис. 3.3.

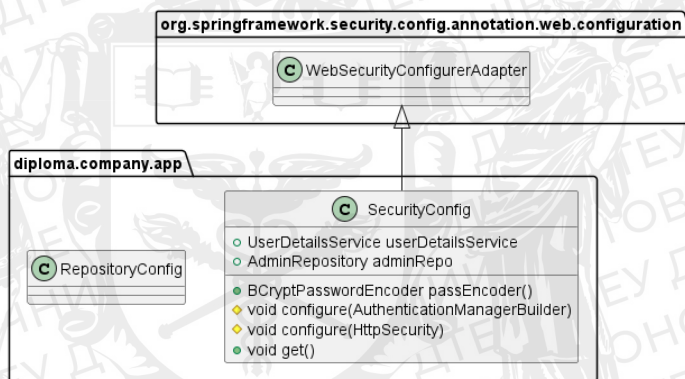


Рис. 3.3 – UML-діаграма для security класів

Джерело: побудовано автором

2) Визначення сутностей та створення моделей:

Аналізуючи предметну область системи спортивного клубу, ідентифікуємо основні сутності, такі як користувачі, тренери, вправи, заняття тощо. Для кожної сутності створюємо відповідні моделі (POJO-класи), що відображають структуру даних цих сутностей. Вони наведені на рис. 3.4.

						ДТЕУ 121 07-26.БР	Аркуш 46
Зм.	Аркуш	№ докум	Підпис	Дата			

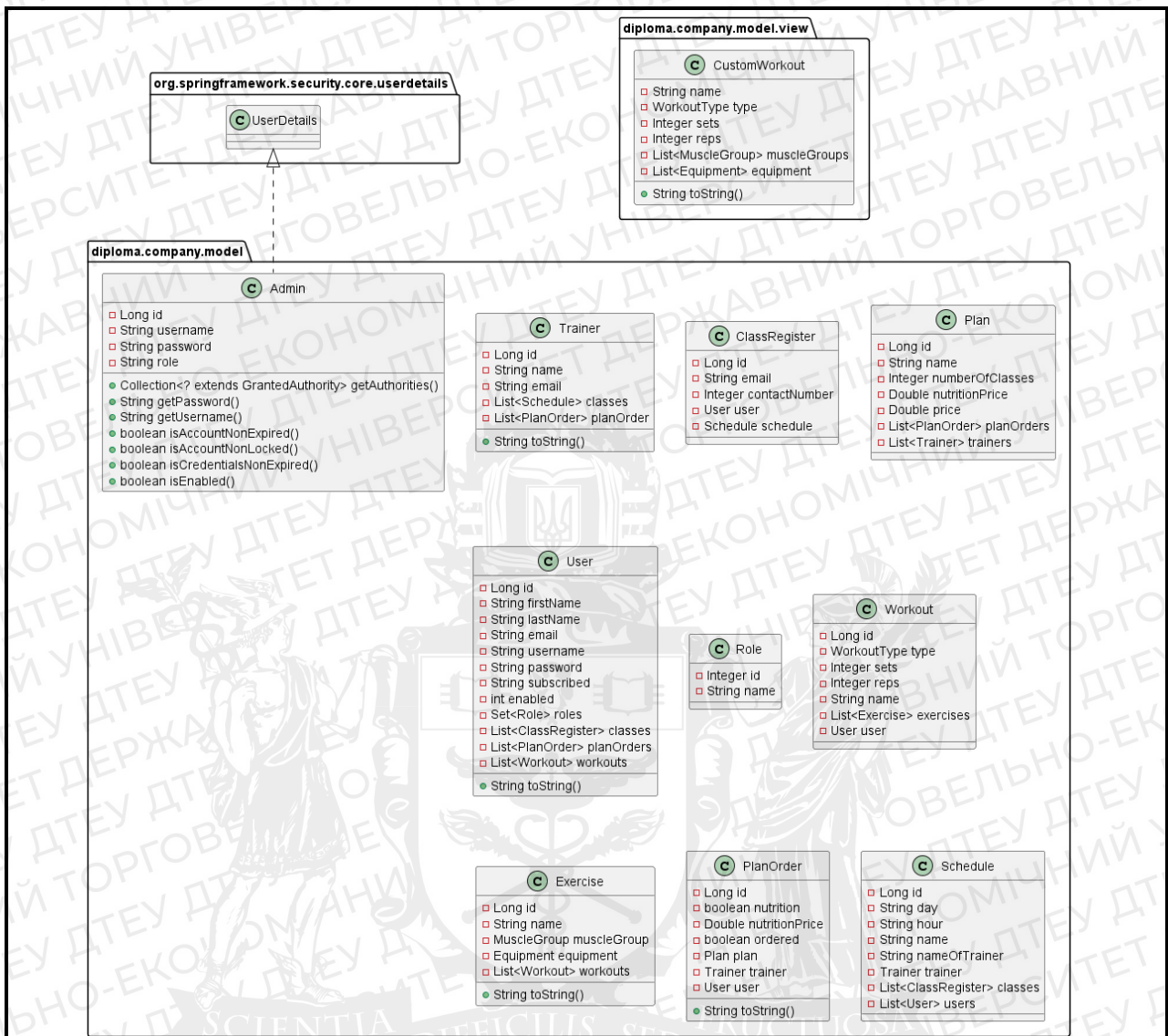


Рис. 3.4 – UML діаграма для класів сутностей

Джерело: побудовано автором

3) Налаштування доступу до бази даних:

Використовуючи Hibernate, налаштуємо з'єднання з базою даних і конфігуруємо параметри доступу до неї. Визначаємо налаштування ORM (Object-Relational Mapping), такі як тип бази даних, URL, ім'я користувача, пароль тощо.

4) Розробка репозиторіїв:

Створюємо репозиторії, які відповідають за доступ до даних в базі даних. Вони надають методи для збереження, оновлення, видалення та отримання

						ДТЕУ 121 07-26.БР	Аркуш 47
Зм.	Аркуш	№ докум	Підпис	Дата			

даних для кожної сутності. Використовуючи анотації Spring та Hibernate, забезпечуємо взаємодію з базою даних, яка наведена на рис. 3.5.

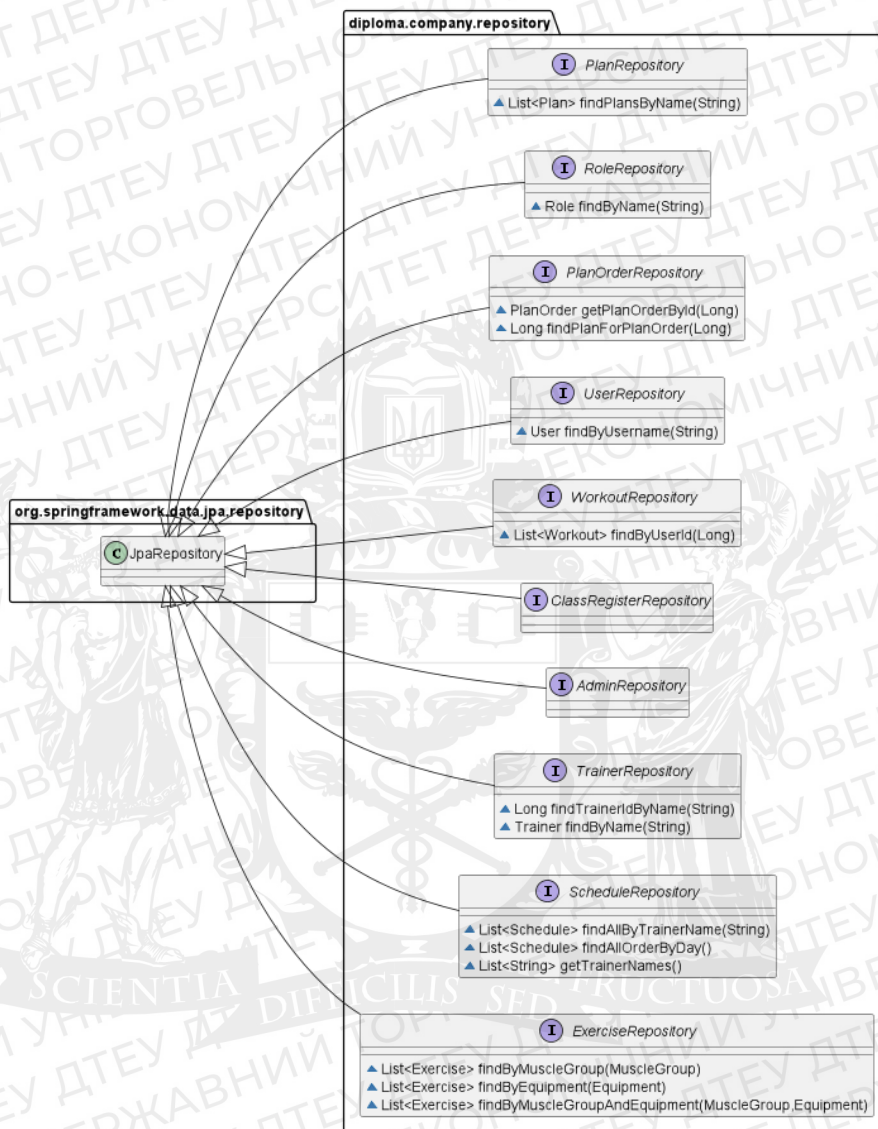


Рис. 3.5 – UML діаграма для класів-репозиторіїв

Джерело: побудовано автором

5) Розробка контролерів:

Створюємо контролери, які відповідають за обробку вхідних запитів від клієнта. Використовуючи анотації Spring, маємо URL-шляхи до відповідних методів контролера. У цих методах виконується обробка запиту, взаємодія з сервісами та повернення відповідей клієнту у відповідному форматі, наприклад, JSON. Вони наведені на рис. 3.6.

						ДТЕУ 121 07-26.БР	Аркуш 48
Зм.	Аркуш	№ докум	Підпис	Дата			

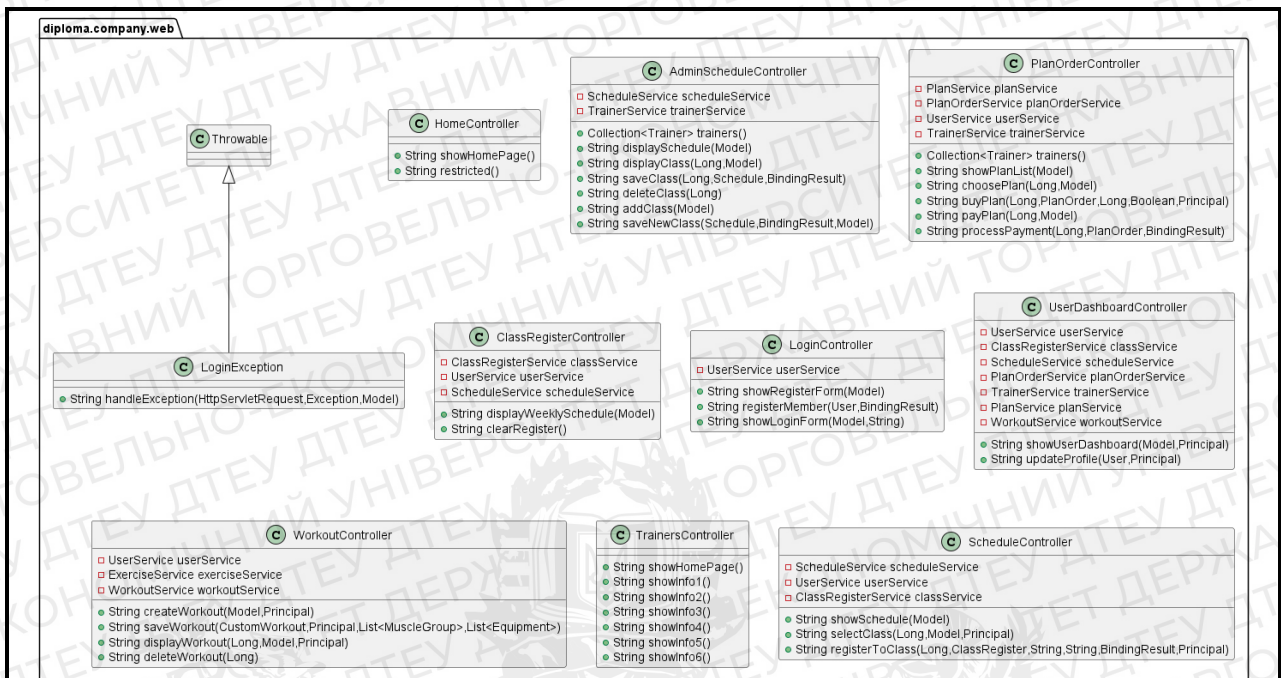


Рис. 3.6 – UML діаграма для класів контролерів

Джерело: побудовано автором

Поміщати всю логіку до методів в контролері є поганою практикою, тому окрім самих контролерів створюється додатковий прошарок, який називається сервіс. Саме сервіс відповідає за взаємозв'язок з базою даних та бізнес-логікою. На рис. 3.7 наведена UML діаграма для прошарку Service.

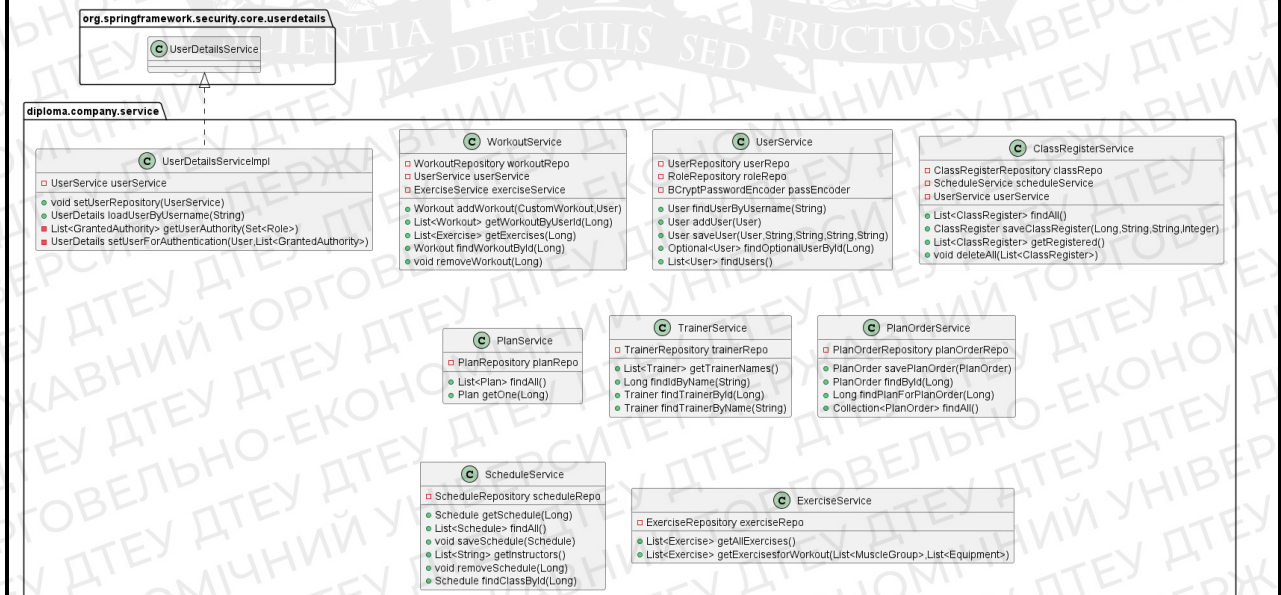


Рис. 3.7 – UML-діаграма для прошарку Service

Джерело: побудовано автором

						Аркуш
					ДТЕУ 121 07-26.БР	49
Зм.	Аркуш	№ докум	Підпис	Дата		

6) Налаштування маршрутизації:

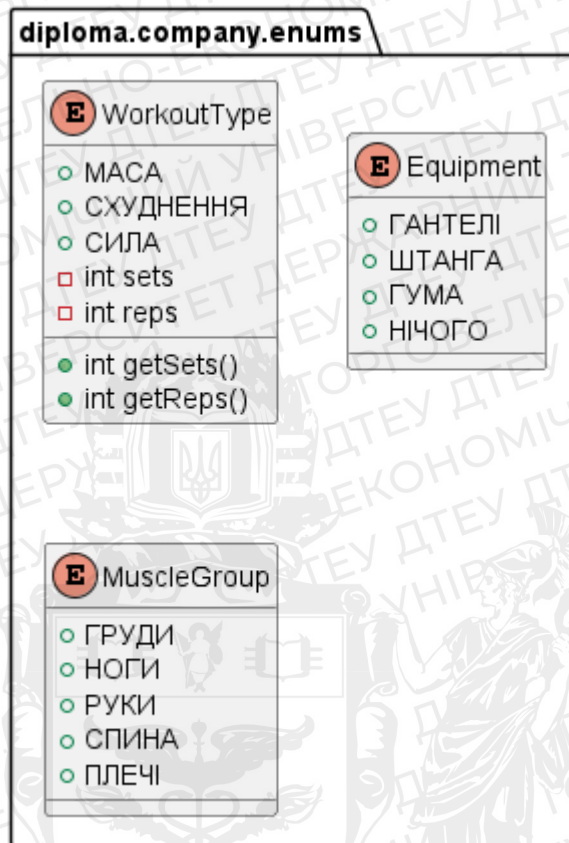


Рис. 3.8 – UML діаграма присутніх перелічень

Джерело: побудовано автором

Використовуючи Spring, налаштуємо маршрутизацію, щоб забезпечити правильну адресацію та перенаправлення запитів до відповідних контролерів.

7) Валідація даних:

Реалізуємо валідацію вхідних даних, щоб перевіряти їх на коректність та забезпечити цілісність бази даних. Використовуючи анотації Spring або створюючи власні валідатори, які наведені на рис. 3.9, проводимо перевірку даних на основі заданих правил.

					ДТЕУ 121 07-26.БР	Аркуш
						50
Зм.	Аркуш	№ докум	Підпис	Дата		

diploma.company.model.view

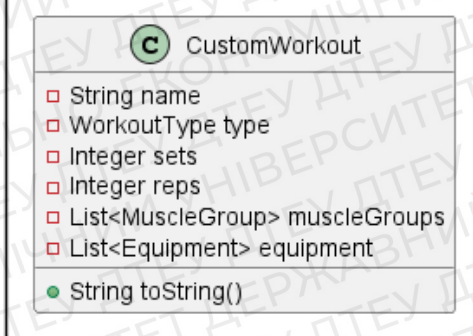


Рис. 3.9 – UML-діаграма класу, який відповідає за відображення тренування

Джерело: побудовано автором

8) Обробка помилок та винятків:

Розробляємо обробники помилок та винятків, які забезпечують коректну обробку непередбачених ситуацій і відповідну відповідь клієнту. Це дозволяє зробити систему більш надійною та стійкою до помилок.

9) Тестування та налагодження:

Після реалізації серверної частини проводимо тестування, включаючи юніт-тести, інтеграційні тести та валідацію. Забезпечуємо, що система працює належним чином та відповідає вимогам специфікації.

10) Розгортання та підтримка:

Після успішного тестування серверної частини можна розгорнути її на веб-сервері або хмарному середовищі. Залежно від вимог інфраструктури, можна використовувати різні інструменти для розгортання, такі як Docker або платформи для хостингу застосунків.

Після розгортання серверної частини необхідно забезпечити її підтримку та надання послуг користувачам. Це включає моніторинг працездатності сервера, вирішення можливих проблем, оновлення програмного забезпечення та надання технічної підтримки користувачам. Він наведений на рис. 3.10.

						ДТЕУ 121 07-26.БР	Аркуш 51
Зм.	Аркуш	№ докум	Підпис	Дата			

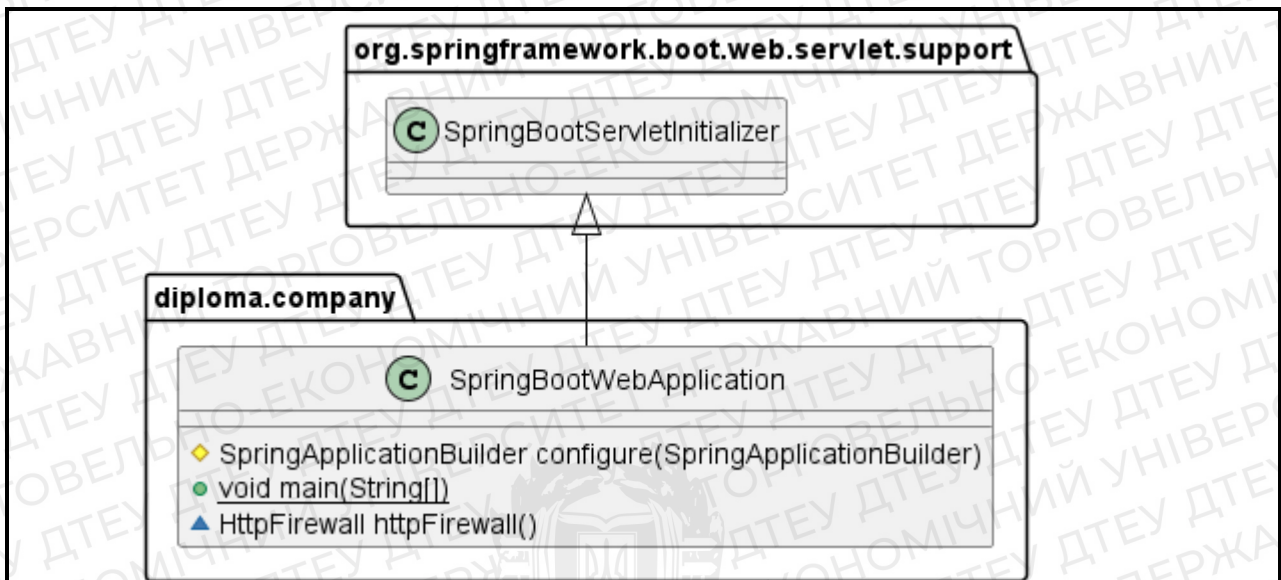


Рис. 3.10 – UML-діаграма для головного класу застосунку (точка входу)

Джерело: побудовано автором

Крім того, рекомендується регулярно вдосконалювати серверну частину, вносячи нові функції, вдосконалюючи безпеку та продуктивність. Також слід вести журнали діагностичних повідомлень для аналізу роботи системи та виявлення можливих проблем в процесі експлуатації.

Окремим блоком виділимо розмежування ролей. Spring Security є потужним інструментом для забезпечення безпеки веб-додатків. Він дозволяє контролювати доступ до ресурсів додатку, включаючи захищені URL-адреси, та здійснювати автентифікацію та авторизацію користувачів. Один з важливих аспектів Spring Security - це можливість розподілу ролей та прав доступу між користувачами.

Для розподілу ролей спочатку потрібно визначити список ролей, які існують у системі, наприклад, "користувач" та "адміністратор". Далі потрібно налаштувати Spring Security для використання цих ролей. Це можна зробити через конфігураційний файл або за допомогою анотацій.

Після налаштування ролей, можна задавати обмеження доступу до певних ресурсів на основі ролей. Наприклад, можна визначити, що тільки користувачі

						ДТЕУ 121 07-26.БР	Аркуш 52
Зм.	Аркуш	№ докум	Підпис	Дата			

з роллю "адміністратор" мають доступ до адміністративної панелі. Це забезпечує виконання принципу нестрашності (Least Privilege), де користувачам надаються лише необхідні права доступу відповідно до їх ролі.

Spring Security також надає можливість налаштувати автентифікацію користувачів, включаючи різні методи автентифікації, такі як використання логіна/пароля, JWT-токенів тощо. За допомогою Spring Security можна виконувати перевірку автентифікації перед доступом до захищених ресурсів.

Загалом, Spring Security забезпечує потужні інструменти для забезпечення безпеки і розподілу ролей у веб-додатках. Вони наведені на рис. 3.11.

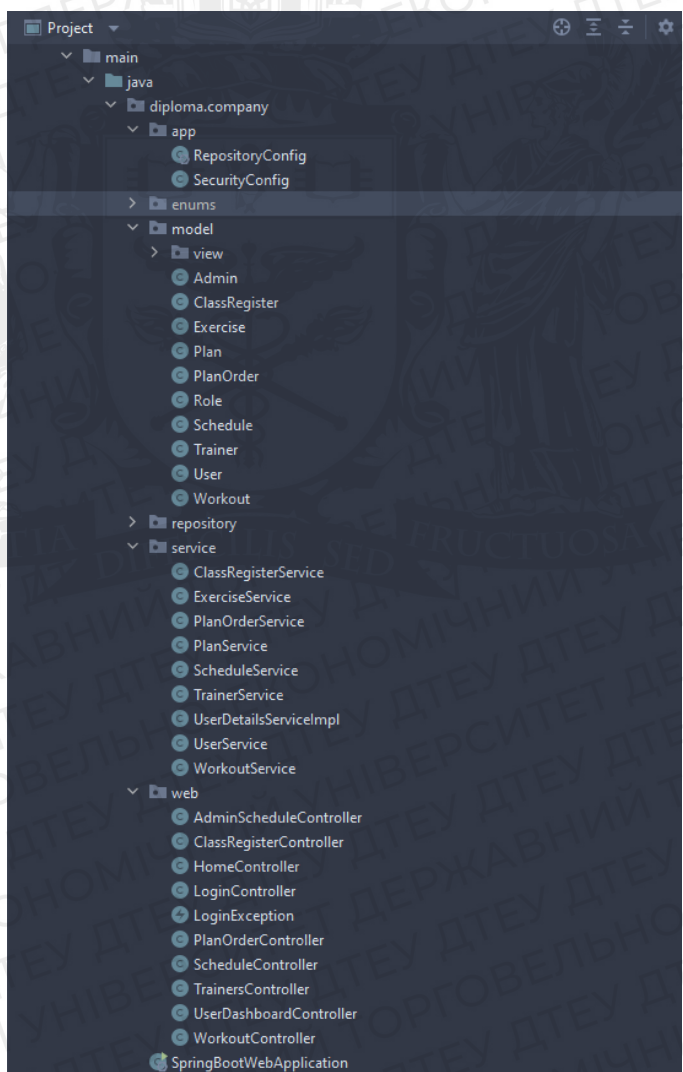


Рис. 3.11 – Розроблена система класів серверної частини

Джерело: побудовано автором

						ДТЕУ 121 07-26.БР	Аркуш 53
Зм.	Аркуш	№ докум	Підпис	Дата			

3.5. Розробка клієнтської частини системи

У цьому розділі дипломної роботи розглядається процес розробки клієнтської частини системи. Клієнтська частина веб-додатків та систем відіграє важливу роль у створенні зручного та ефективного інтерфейсу для користувачів. Вона відповідає за відображення інформації, взаємодію з користувачем та передачу даних на серверну частину.

Першим кроком у створенні клієнтської частини є розробка дизайну та макету інтерфейсу. Це включає в себе визначення структури сторінок, розташування елементів, використання кольорів та інших візуальних елементів. На цьому етапі можна використовувати різні інструменти та технології, наприклад, Photoshop, Sketch, Figma, або навіть папір та олівець.

JSP (JavaServer Pages) є технологією, яка дозволяє вбудовувати код Java в сторінки HTML. На цьому етапі потрібно створити JSP-шаблони для різних сторінок системи, включаючи реєстрацію, вхід в систему, головну сторінку тощо. У шаблонах JSP можна використовувати теги JSTL (JavaServer Pages Standard Tag Library) для динамічного відображення даних та управління потоком виконання.

JavaScript є мовою програмування, яка використовується для створення динамічних елементів на веб-сторінках. На цьому етапі потрібно розробити клієнтський JavaScript-код для валідації форм, взаємодії з сервером за допомогою AJAX-запитів, анімації елементів тощо.

CSS (Cascading Style Sheets) використовується для стилізації веб-сторінок. На цьому етапі потрібно визначити стилі для елементів інтерфейсу, такі як кольори, шрифти, розміри, відступи та інші властивості. Важливо забезпечити консистентність стилів у всьому додатку, щоб сторінки мали єдиний зовнішній вигляд.

					ДТЕУ 121 07-26.БР	Аркуш
						54
Зм.	Аркуш	№ докум	Підпис	Дата		

Після створення клієнтської частини необхідно провести тестування, щоб переконатися в правильному відображенні та функціонуванні інтерфейсу. Виконайте тестування на різних пристроях та браузерах, щоб переконатися, що інтерфейс адаптується належним чином і працює коректно.

Під час розробки клієнтської частини важливо звернути увагу на оптимізацію та покращення продуктивності. Це включає усунення зайвого коду, мінімізацію та стиснення файлів CSS та JavaScript, використання кешування ресурсів, що не змінюються часто, та інші оптимізаційні підходи. Це допоможе забезпечити швидку та ефективну роботу клієнтської частини системи.

Окремо також слід виділити процес творення адаптивного інтерфейсу, який став одним з найбільш трудомістких процесів розробки інформаційної системи. Процес відбувався в кілька етапів:

Адаптивний інтерфейс передбачає зміну вигляду та розміщення елементів в залежності від розміру екрану пристрою. Перший крок - це визначення точок перелому, або breakpoints, коли інтерфейс буде змінюватися. Наприклад, можна визначити точку перелому для мобільних пристроїв, планшетів та настільних комп'ютерів.

Медіа-запити дозволяють застосовувати різні стилі до елементів в залежності від розміру екрану. Використовуючи CSS, можна визначити правила стилізації, які будуть застосовуватися лише в певних діапазонах розмірів екрану. Наприклад, можна змінювати розміри, позицію або видимість елементів при досягненні певних breakpoints.

Використання гнучкої сітки допомагає адаптувати розміщення елементів на сторінці в залежності від розміру екрану. За допомогою CSS-фреймворків, таких як Bootstrap або Foundation, можна створити сітку, яка автоматично пристосовуватиметься до розміру екрану. Це дозволить елементам розміщуватися відповідно до доступного простору.

					ДТЕУ 121 07-26.БР	Аркуш
						55
Зм.	Аркуш	№ докум	Підпис	Дата		

Резиновий дизайн передбачає використання відносних одиниць вимірювання (наприклад, відсотків) для визначення розмірів та розміщення елементів. Такий підхід дозволяє елементам пристосовуватися до розміру екрану.

Адаптивний інтерфейс може включати скриття або показ елементів залежно від розміру екрану. Наприклад, на менших пристроях можна приховувати деякі навігаційні елементи або замінювати їх бургер-меню для економії місця на екрані. З використанням CSS та JavaScript можна керувати видимістю та поведінкою елементів на різних breakpoints.

Після розробки адаптивного інтерфейсу важливо протестувати його на різних пристроях та розмірах екрану. Використовуйте емулятори пристроїв або реальні пристрої, щоб переконатися, що інтерфейс належним чином адаптується та виглядає коректно на різних пристроях.

Розробка адаптивного інтерфейсу є ітеративним процесом. Почніть з базового дизайну та функціоналу для найширшого екрану, потім поступово додавайте зміни для менших розмірів екрану. Здійсніть тестування та виправлення помилок на кожному етапі, забезпечуючи коректну роботу інтерфейсу на різних пристроях.

3.6. Демонстрація розробленої системи

На рис. 3.12 – 3.17 продемонстрована робота програмного додатку.

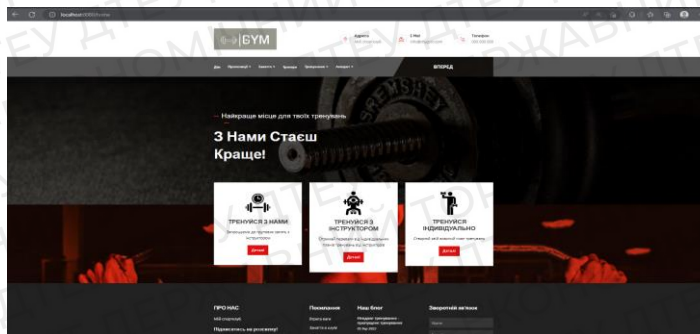


Рис. 3.12 – Головна сторінка при розширенні 1920 x 1080 px

					ДТЕУ 121 07-26.БР	Аркуш
						56
Зм.	Аркуш	№ докум	Підпис	Дата		

Джерело: побудовано автором

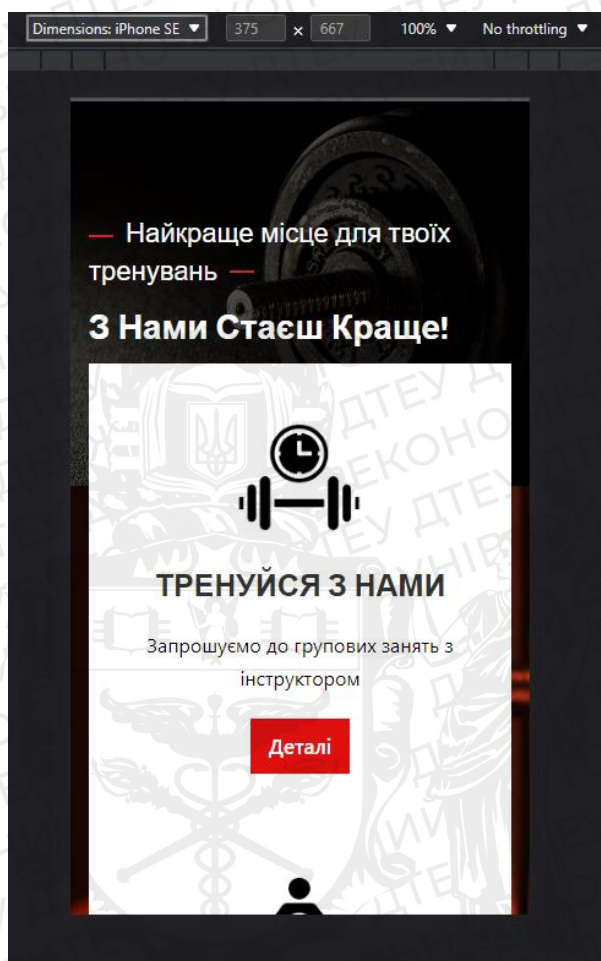


Рис. 3.13 – Головна сторінка при відображенні на дисплеї iPhone SE

Джерело: побудовано автором

Абсолютно всі сторінки сайту містять однакову шапку сторінки header, на якій розміщено основні кнопки для навігації по сайту, такі як домашня сторінка, реєстрація або авторизація, програми занять, плани тренувань, покупка планів тренувань, розділ з інформацією про тренерів. На нижній панелі всіх сторінок знаходиться колонтитул (footer) на якому розміщена інформація про авторські права, також блог спортивного клубу, форма для відправки повідомлень та зворотного зв'язку, декілька корисних посилань.

					ДТЕУ 121 07-26.БР	Аркуш
						57
Зм.	Аркуш	№ докум	Підпис	Дата		

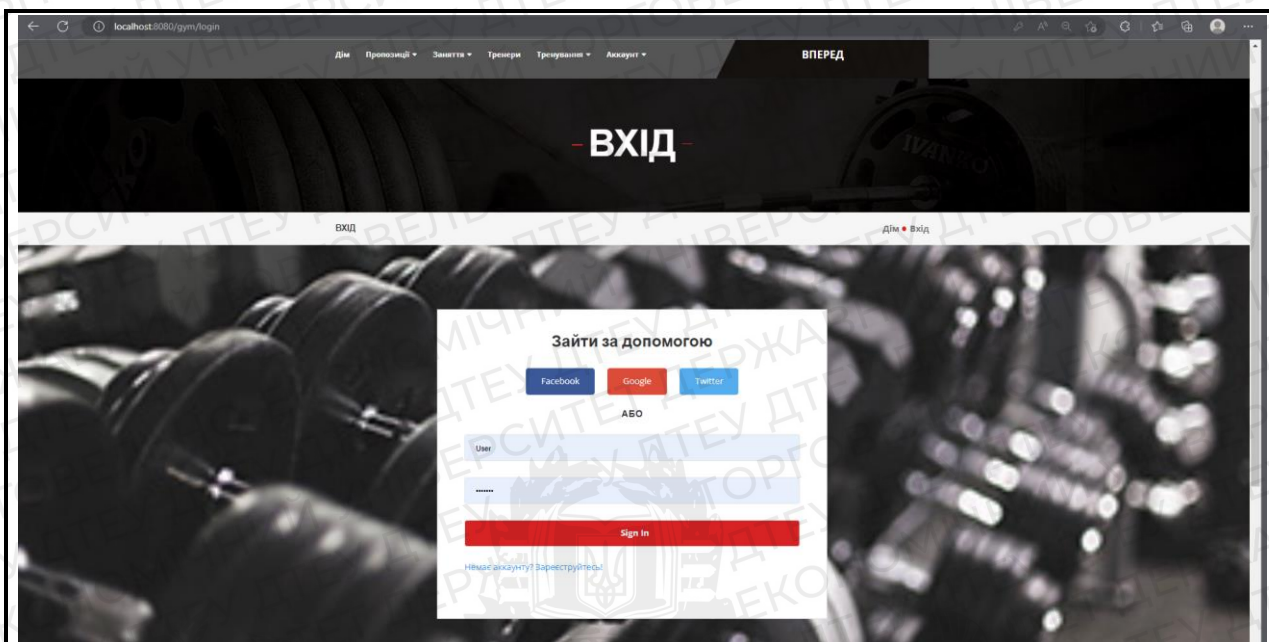


Рис. 3.14 – Вікно для авторизації користувача

Джерело: побудовано автором

Після авторизації користувачу стає доступний особистий кабінет, де можна подивитись записи на тренування, придбати план тренувань або створити свій власний план тренувань.

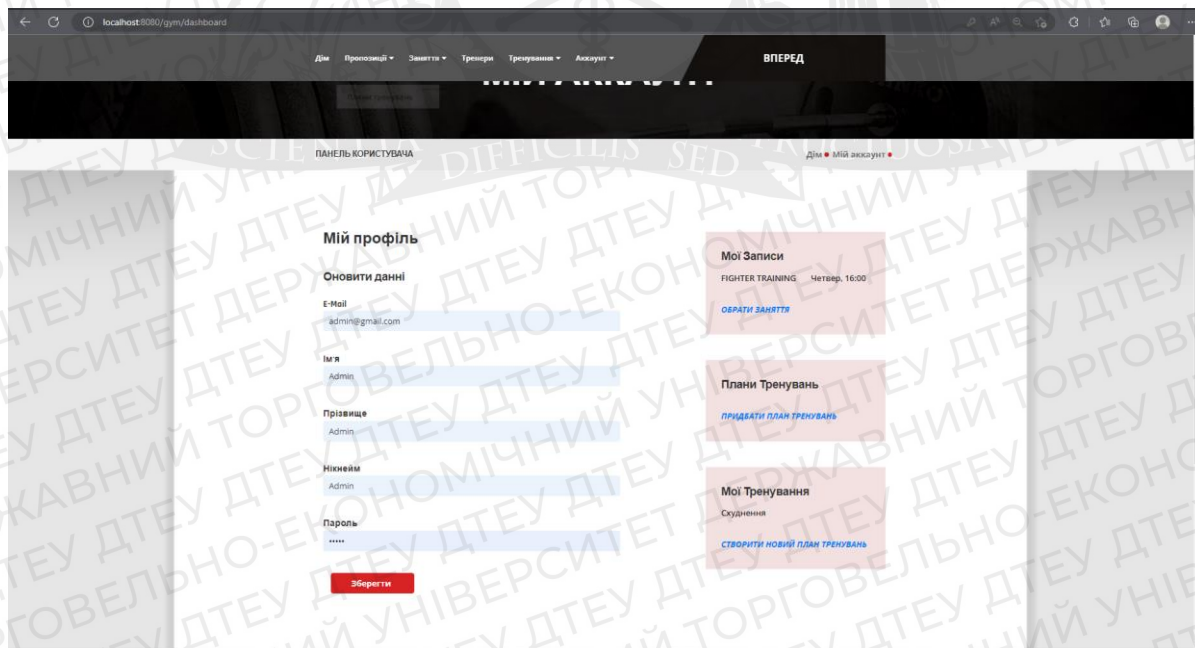


Рис. 3.15 – Вікно з особистим кабінетом користувача

Джерело: побудовано автором

					ДТЕУ 121 07-26.БР	Аркуш
						58
Зм.	Аркуш	№ докум	Підпис	Дата		

На сайті також присутній розклад проведення занять. Користувач може з легкістю записатись на заняття, натиснувши на відповідну кнопку в зручний для **НЬОГО** день.

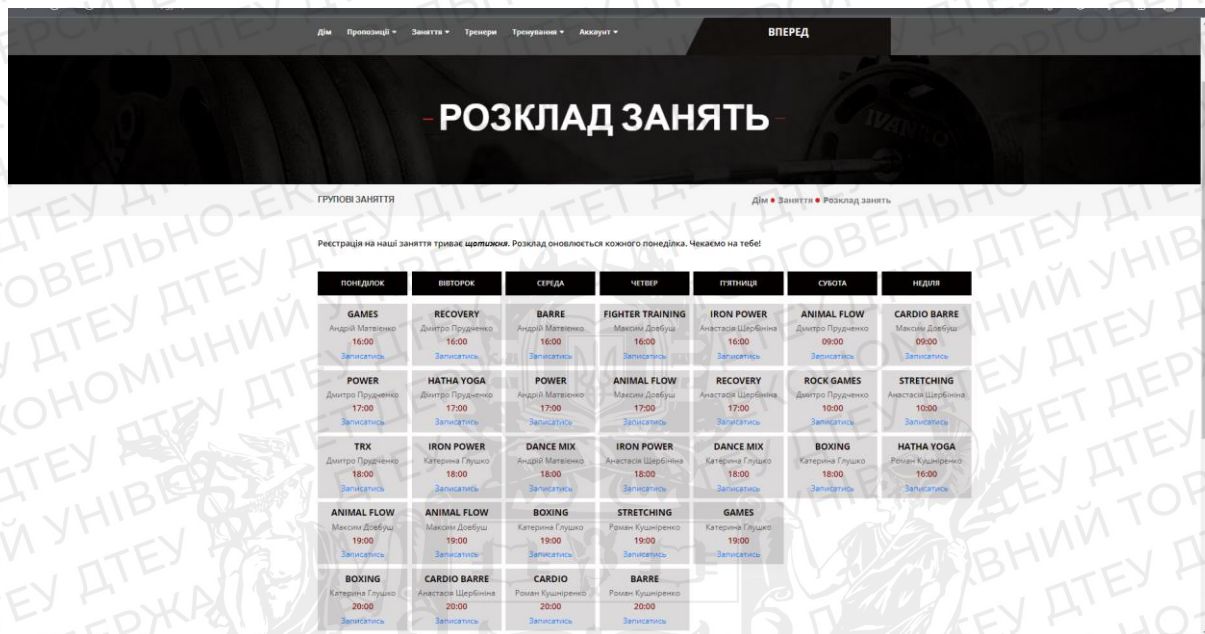


Рис. 3.16 – Вікно з розкладом занять

Джерело: побудовано автором

Після натискання на кнопку записатись в обраний день, користувач буде перенаправлений на сторінку з записом на тренування.

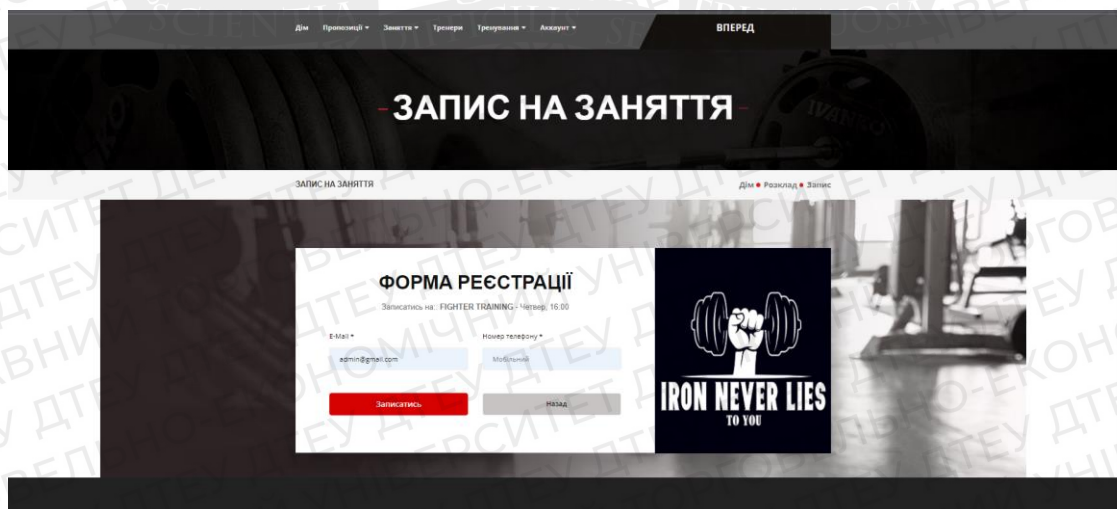


Рис. 3.17 – Форма запису на тренування

Джерело: побудовано автором

						Аркуш
						59
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-26.БР	

3.7. Висновки до розділу 3

У даному розділі було проведено розробку інформаційної системи для мережі спортивних клубів з використанням мови програмування Java, фреймворка Spring Boot та JavaScript. Реляційна база даних MySQL була обрана для зберігання та керування даними системи.

Було розроблено клієнтську частину системи, яка включає веб-інтерфейс для користувачів. За допомогою JavaScript та інших технологій було реалізовано функціональність, яка дозволяє клієнтам спортивних клубів взаємодіяти з системою, реєструватися на тренування, отримувати інформацію про розклад та інші послуги.

Також було розроблено серверну частину системи, яка включає логіку обробки запитів, взаємодію з базою даних та забезпечення безпеки і аутентифікації. Використання фреймворка Spring Boot спростило розробку серверної частини та забезпечило ефективність та масштабованість системи.

Під час демонстрації роботи розробленої системи було показано, що вона успішно функціонує та забезпечує необхідну функціональність для спортивних клубів. Користувачі можуть здійснювати реєстрацію, переглядати розклад тренувань, здійснювати оплату та отримувати інші послуги через веб-інтерфейс системи.

Висновки зазначають, що розроблена інформаційна система відповідає поставленим вимогам та дозволяє покращити управління та надання послуг власникам спортивних клубів. Використання Java, Spring Boot та JavaScript дозволило реалізувати потрібний функціонал і забезпечити ефективну роботу системи. Вибір реляційної бази даних MySQL є відповідним для потреб зберігання та обробки даних спортивних клубів.

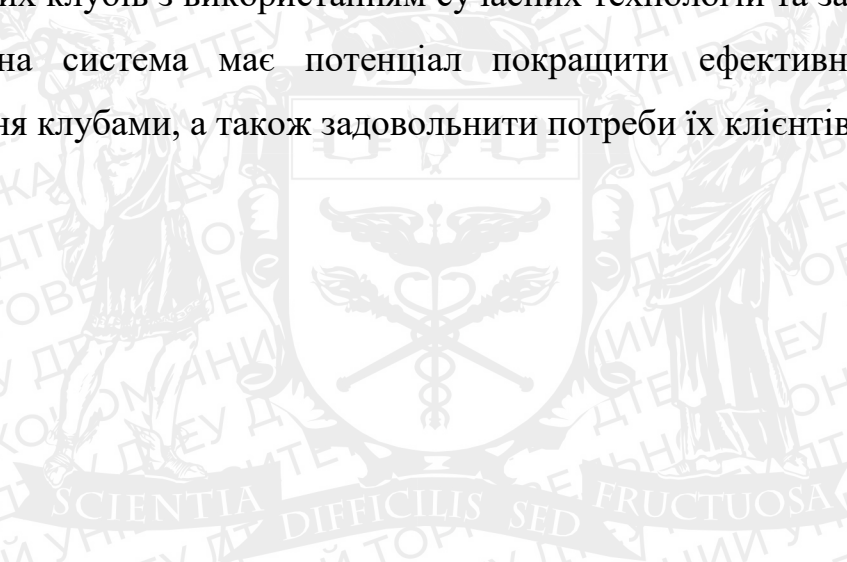
Розробка клієнтської та серверної частини системи демонструє професійність та навички у розробці програмного забезпечення. Клієнтська

					ДТЕУ 121 07-26.БР	Аркуш
						60
Зм.	Аркуш	№ докум	Підпис	Дата		

частина забезпечує зручний інтерфейс для користувачів, тоді як серверна частина виконує важливі функції обробки даних, забезпечення безпеки та взаємодії з базою даних.

Під час демонстрації роботи системи було підтверджено її функціональність та відповідність поставленим вимогам. Розроблена система може ефективно підтримувати роботу спортивних клубів, спрощуючи адміністративні процеси та поліпшуючи задоволення клієнтів.

У підсумку, розділ "Розробка інформаційної системи" підтверджує успішну реалізацію проекту та створення інформаційної системи для спортивних клубів з використанням сучасних технологій та засобів розробки. Розроблена система має потенціал покращити ефективність та якість управління клубами, а також задовольнити потреби їх клієнтів.



					ДТЕУ 121 07-26.БР	Аркуш
						61
Зм.	Аркуш	№ докум	Підпис	Дата		

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

У підсумку випускної роботи варто відзначити, що інтеграція автоматизації тренажерного залу потребує початку з найважливішої частини його роботи, яка надасть значний ефект. Під час розробки веб-сайту необхідно забезпечити можливість працювати з новими та старими технологіями, щоб не втручатися в роботу спортивного клубу.

Аналіз існуючих програмних рішень для роботи з клієнтами спортивних клубів показав, що вони не відповідають вимогам замовника, тому було розроблено інформаційну систему для роботи з відвідувачами спортивного залу, яка враховує потреби користувачів:

- Механізм ідентифікації користувача в системі.
- Створення та зберігання цифрових програм тренувань.
- Додавання, видалення та перегляд програм тренувань.
- Надання доступу до програм тренувань.
- Перевірка коректності введеної інформації.

У процесі розробки веб-сайту були вибрані такі технології: HTML для мови розмітки гіпертексту, MySQL для системи управління базами даних, JavaScript для клієнтської частини та Java в поєднанні зі Spring Boot та Hibernate для програмно-апаратної частини. Візуальне оформлення сайту здійснюється за допомогою CSS, а основна логіка реалізована за допомогою JavaScript.

Нова інформаційна система для організації роботи з відвідувачами спортивного залу має великий потенціал для поліпшення ефективності та зручності управління клієнтами. Шляхом використання сучасних технологій, таких як React та Node.js, система забезпечує швидку та плавну взаємодію з

Зм.	Аркуш	№ докум.	Підпис	Дата				
Зав. каф.		Криворучко О.В.		28.04.23	ДТЕУ 121 07-26.БР			
Керівник		Власенко Л.О.		28.04.23				
Гарант		Рзаєва С.Л.		28.04.23	Програмний додаток мережі спортивних клубів	Стадія	Аркуш	Аркушів
Розробив		Чеботаєв Д.О.		28.04.23		ВП	62	65
					Висновки та пропозиції	Факультет інформаційних технологій 4 курс, 7 група		

користувачами, а також можливість працювати на різних пристроях та платформах завдяки адаптивному дизайну.

Одним із ключових аспектів розробки було створення механізму ідентифікації користувача, що дозволяє кожному відвідувачу зареєструватися в системі та отримати персоналізований доступ до програм тренувань. Цей механізм гарантує безпеку та конфіденційність даних користувачів, а також забезпечує зручну авторизацію та аутентифікацію.

Розробка цифрового формату для програм тренувань дозволяє відвідувачам створювати власні тренувальні плани, зберігати їх у системі та легко редагувати або видаляти за необхідністю. Крім того, інформаційна система надає можливість обміну програмами тренувань між користувачами, що сприяє обміну досвідом та співпраці між тренувальними групами.

Одним із важливих аспектів було також забезпечення перевірки правильності введення інформації користувачами. Система надає можливість перевіряти та валідувати дані, що вводяться у програми тренувань, запобігаючи можливим помилкам або некоректним даним.

					ДТЕУ 121 07-26.БР	Аркуш
						63
Зм.	Аркуш	№ докум	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Spring Framework. [Електронний ресурс]. Режим доступу: <https://spring.io/>. (дата звернення 21.01.2023)
2. Hibernate. [Електронний ресурс]. Режим доступу: <https://hibernate.org/>. (дата звернення 17.02.2023)
3. Oracle Java Documentation. [Електронний ресурс]. Режим доступу: <https://docs.oracle.com/en/java/>. (дата звернення 17.02.2023)
4. MySQL Documentation. [Електронний ресурс]. Режим доступу: <https://dev.mysql.com/doc/>. (дата звернення 17.02.2023)
5. Walls, C. Spring in Action (October 2018). 520 p.
6. Bauer, C., King, G., Gregory, G. Java Persistence with Hibernate. (October 2015). 608 p.
7. Flanagan, D. JavaScript: The Definitive Guide (May 2018). 220 p.
8. Duckett, J. HTML and CSS: Design and Build Websites (November 8, 2011). 490 p.
9. Freeman, E., Robson, E. Head First HTML and CSS: A Learner's Guide to Creating Standards-Based Web Pages (October 2, 2012). 762 p.
10. Schutta, J. Building RESTful Web Services with Java EE 8 (July 31, 2018). 116 p.
11. Gosling, J., Joy, B., Steele, G., Bracha, G. The Java Language Specification 1 (September 2018). 772 p.
12. Schildt, H. Java: The Complete Reference (November 23, 2021). 1280 p.

<i>ДТЕУ 121 07-26.БР</i>																	
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>													
Зав. каф.		Криворучко О.В.		23.12.22	Програмний додаток мережі спортивних клубів												
Керівник		Власенко Л.О.		23.12.22													
Гарант		Рзаєва С.Л.		23.12.22	Список використаних джерел												
Розробив		Чеботасв Д.О.		23.12.22													
					<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20%; text-align: center;"><i>Стадія</i></td> <td style="width: 20%; text-align: center;"><i>Аркуш</i></td> <td style="width: 20%; text-align: center;"><i>Аркушів</i></td> </tr> <tr> <td style="text-align: center;">СВД</td> <td style="text-align: center;">64</td> <td style="text-align: center;">65</td> </tr> <tr> <td colspan="3" style="text-align: center;"><i>Факультет інформаційних технологій</i></td> </tr> <tr> <td colspan="3" style="text-align: center;"><i>4 курс, 7 група</i></td> </tr> </table>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>	СВД	64	65	<i>Факультет інформаційних технологій</i>			<i>4 курс, 7 група</i>		
<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>															
СВД	64	65															
<i>Факультет інформаційних технологій</i>																	
<i>4 курс, 7 група</i>																	

13. Duckett, J. JavaScript and JQuery: Interactive Front-End Web Development (June 30, 2014). 640 p.
14. Freeman, E., Robson, E. Head First JavaScript Programming: A Brain-Friendly Guide (May 6, 2014). 700 p.
15. FitPro [Електронний ресурс] – Режим доступу до ресурсу: <https://www.fitpro.com/>. (дата звернення 19.03.2023)
16. Clubmaster [Електронний ресурс] – Режим доступу до ресурсу: <https://clubmaster.com/>. (дата звернення 19.03.2023)
17. w3docs [Електронний ресурс] – Режим доступу до ресурсу: <https://www.w3docs.com/>. (дата звернення 19.03.2023)
18. Litslink [Електронний ресурс] – Режим доступу до ресурсу: <https://litslink.com/>. (дата звернення 19.03.2023)
19. VseOsvita [Електронний ресурс] – Режим доступу до ресурсу: vseosvita.ua. (дата звернення 19.03.2023)
20. MDN Web Docs. [Електронний ресурс]. Режим доступу: <https://developer.mozilla.org/>. (дата звернення 19.03.2023)
21. Stack Overflow. [Електронний ресурс]. Режим доступу: <https://stackoverflow.com/>. (дата звернення 28.03.2023)
22. DZone. [Електронний ресурс]. Режим доступу: <https://dzone.com/>. (дата звернення 28.03.2023)
23. JavaWorld. [Електронний ресурс]. Режим доступу: <https://www.javaworld.com/>. (дата звернення 29.03.2023)
24. Tutorialspoint. [Електронний ресурс]. Режим доступу: <https://www.tutorialspoint.com/>. (дата звернення 30.03.2023)

					ДТЕУ 121 07-26.БР	Аркуш 65
Зм.	Аркуш	№ докум	Підпис	Дата		

ДОДАТКИ

ДОДАТОК А

Код головної сторінки

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form" %>
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Мій спортклуб</title>
  <link href="<c:url
value="https://fonts.googleapis.com/css?family=Rajdhani:300,400,500,600,700&
isplay=swap"/>" rel="stylesheet">
  <link href="<c:url
value="https://fonts.googleapis.com/css?family=Open+Sans:400,400i,600,600i,70
0,700i&display=swap"/>"
rel="stylesheet">
  <link rel="stylesheet" href="<c:url
value="http://cdn.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"/>">
  <link rel="stylesheet" href="<c:url value="/static/css/all.min.css"/>">
  <link rel="stylesheet" href="<c:url value="/static/css/animate.css"/>">
  <link rel="stylesheet" href="<c:url value="/static/css/lightcase.css"/>">
  <link rel="stylesheet" href="<c:url value="/static/css/flaticon.css"/>">
  <link rel="stylesheet" href="<c:url
value="/static/css/swiper.min.css"/>">
  <link rel="stylesheet" href="<c:url value="/static/css/style.css"/>">
  <link rel="stylesheet" href="<c:url value="/static/css/services.css"/>">
  <link rel="stylesheet" href="<c:url
value="http://www.w3ii.com/lib/w3.css"/>">
</head>
<body>
<!-- Header -->
<%@ include file="./header.jsp" %>
<!-- /.header -->
<a href="#" class="scrollToTop"><i class="fas fa-angle-up"></i></a>
<!-- Banner Section Starts Here -->
<section class="banner-section bg_img" data-
background="./static/images/home_barbell.jpg">
  <div class="container">
    <div class="banner-content-container">
      <div class="banner-content">
        <h4 class="sub-title">
          <span class="shape-wrapper"><span
class="shape"></span>Найкраще місце для твоїх тренувань<span
class="shape"></span></span></h4>
        <h5 class="title"> з нами стаєш краще!</h5>
      </div>
    </div>
  </div>
</section>
<div class="building-outer">
  <div class="container">
```



```

<div class="building-list">
  <div class="row">
    <div class="col-sm-4 col-xs-12">
      <div class="building-box">
        <figure><img src=<c:out
value="/static/images/home/t.PNG"/>?" alt=""></figure>
        <h4>ТРЕНУЙСЯ З НАМИ</h4>
        <p>Запрошуємо до групових занять з інструктором</p><a
href="<c:out value="/gym/schedule"/>" class="btn">Детали</a></div>
      </div>
    <div class="col-sm-4 col-xs-12">
      <div class="building-box">
        <figure><img src=<c:out
value="/static/images/home/tr.PNG"/>?" alt=""></figure>
        <h4>ТРЕНУЙСЯ З ІНСТРУКТОРОМ</h4>
        <p>Отримай переваги від індивідуальних планів
тренувань від інструкторів </p><a href="<c:out value="/gym/plans"/>"
class="btn">Детали</a></div>
      </div>
    <div class="col-sm-4 col-xs-12">
      <div class="building-box">
        <figure><img src=<c:out
value="/static/images/home/i.PNG"/>?" alt=""></figure>
        <h4>ТРЕНУЙСЯ ІНДИВІДУАЛЬНО</h4>
        <p>Створи свій власний план тренувань</p><a
href="<c:out value="/gym/workout/create"/>" class="btn">Детали</a></div>
      </div>
    </div>
  </div>
</div>
</body>
<!-- Footer Section -->
<%@ include file="./footer.jsp" %>
</html>

```

Код сторінки реєстрації

```

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form" %>
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Мій спортклуб</title>
  <link href="<c:url
value="https://fonts.googleapis.com/css?family=Rajdhani:300,400,500,600,700&d
isplay=swap"/>"
    rel="stylesheet">
  <link href="<c:url
value="https://fonts.googleapis.com/css?family=Open+Sans:400,400i,600,600i,70
0,700i&display=swap"/>"
    rel="stylesheet">
  <link rel="stylesheet" href="<c:url
value="http://cdn.bootstrampb.com/bootstrap/4.3.1/css/bootstrap.min.css"/>">
  <link rel="stylesheet" href="<c:url value="/static/css/all.min.css"/>">
  <link rel="stylesheet" href="<c:url value="/static/css/animate.css"/>">
  <link rel="stylesheet" href="<c:url value="/static/css/lightcase.css"/>">
  <link rel="stylesheet" href="<c:url value="/static/css/flaticon.css"/>">
  <link rel="stylesheet" href="<c:url
value="/static/css/swiper.min.css"/>">
  <link rel="stylesheet" href="<c:url value="/static/css/style.css"/>">
  <link rel="stylesheet" href="<c:url value="/static/css/services.css"/>">
  <link rel="stylesheet" href="<c:url value="/static/css/class-
register.css"/>">
</head>
<body>
<!-- Header -->
<%@ include file="./header.jsp" %>
<!-- /.header -->
<a href="#" class="scrollToTop"><i class="fas fa-angle-up"></i></a>
<!-- Page Header-->
<section class="page-header bg_img" data-
background="/static/images/banner/05.jpg">
  <div class="container">
    <h3 class="title">
      <span class="shape-wrapper">
        <span class="shape"></span>Запис на заняття<span
class="shape"></span></span></h3>
  </div>
</section>

<!-- Breadcrumb Section -->
<div class="breadcrumb-section">
  <div class="container">
    <div class="breadcrumb-wrapper">
      <div class="breadcrumb-title"><h6 class="title">Запис на
заняття</h6></div>
      <ul class="breadcrumb">
        <li><a href="<c:url value="/home"/>">Дім</a></li>
        <li><a href="<c:url value="/gym/schedule"/>">Розклад</a></li>

```



```

    </div>
  </div>
</div>

<!-- Class register -->
<section class="register-section classes-page spad">
  <div class="container">
    <div class="classes-page-text">
      <div class="row">
        <div class="col-lg-8">
          <div class="register-text">
            <div class="section-title">
              <h2>Форма реєстрації</h2>
              <p>Записатись на: <b>${training.name}</b> -
                ${training.day}, ${training.hour}</p>
            </div>
            <form:form action="/gym/schedule/${training.id}"
              method="post" modelAttribute="training"
              class="register-form">
              <div class="row">
                <p hidden><form:input path="id"/></p>
                <p hidden><form:input path="day"/></p>
                <p hidden><form:input path="hour"/></p>
                <p hidden><form:input path="name"/></p>
                <p hidden><form:input path="nameOfTrainer"/>
                <div class="col-lg-6">
                  <label for="email">E-Mail *</label>
                  <input path="email" type="email"
                    id="email" name="email" value ="${email}" placeholder="${email}"/>
                  <form:errors path="email"
                    cssClass="error"/>--%>
                </div>
                <div class="col-lg-6">
                  <label for="phone">Номер телефону
                    *</label>
                  <input type="tel" id="phone" name="phone"
                    placeholder="Мобільний">
                </div>
                <div class="col-lg-6">
                  <input type="submit" class="register-btn"
                    value="Записатись"/>
                </div>
                <div class="col-lg-6">
                  <input type="button" class="register-btn-
                    2" onclick="location.href='/gym/schedule';"
                    value="Назад"/>
                </div>
              </div>
            </form:form>
          </div>
        </div>
        <div class="col-lg-4">
          <div class="register-pic">
            " alt="">
          </div>
        </div>
      </div>
    </div>
  </div>
</div>

```

```
</section>  
</body>  
<!-- Footer Section -->  
<%@ include file="./footer.jsp" %>  
</html>
```



Серверний код сторінки реєстрації

```
package diploma.company.web;

import diploma.company.model.User;
import diploma.company.service.UserService;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestParam;

import javax.validation.Valid;

@Controller
public class LoginController {

    private final UserService userService;

    public LoginController(UserService userService) {
        this.userService = userService;
    }

    @GetMapping("/gym/register")
    public String showRegisterForm(Model model) {
        User user = new User();
        model.addAttribute("user", user);
        return "registerForm";
    }

    @PostMapping("/gym/register")
    public String registerMember(@Valid @ModelAttribute User user,
        BindingResult result) throws LoginException {
        if (result.hasErrors()) {
            return "registerForm";
        }
        if (userService.findUserByUsername(user.getUsername()) != null) {
            throw new LoginException();
        }
        if (user.getSubscribed() == null) {
            user.setSubscribed(null);
        }
        else if (user.getSubscribed().equals("yes")) {
            user.setSubscribed("yes");
        }
        userService.addUser(user);
        return "redirect:/gym/login";
    }

    @GetMapping("/gym/login")
    public String showLoginForm(Model model, @RequestParam(required = false)
        String error) {
        User user = new User();
        if (error != null) {
            model.addAttribute("error", "\n" + "Incorrect login and/or
password");
        }
        model.addAttribute("user", user);
        return "loginForm";
    }
}
```

Серверний клас User

```
package diploma.company.model;

import lombok.AllArgsConstructor;
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;
import org.jetbrains.annotations.Nullable;

import javax.persistence.*;
import javax.validation.constraints.Email;
import javax.validation.constraints.NotBlank;
import javax.validation.constraints.NotNull;
import javax.validation.constraints.Size;
import java.util.ArrayList;
import java.util.List;
import java.util.Set;

@Table(name = "users")
@Entity
@Getter
@Setter
@NoArgsConstructor
@AllArgsConstructor
public class User {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @NotBlank
    @Size(min = 3, max = 20)
    private String firstName;

    @NotBlank
    @Size(min = 3, max = 20)
    private String lastName;

    @NotBlank
    @Email(message = "Invalid email address format")
    private String email;

    @NotBlank
    @Size(min = 3, max = 20)
    @Column(unique = true)
    private String username;

    @NotBlank
    @Size(min=8, message = "The password must contain at least 8 characters")
    private String password;

    @Nullable
    private String subscribed;

    @NotNull
    private int enabled;

    @ManyToMany(cascade = CascadeType.ALL, fetch = FetchType.EAGER)
    @JoinTable(name = "user_role",
```



```
joinColumns = @JoinColumn(name = "user_id"),
inverseJoinColumns = @JoinColumn(name = "role_id")

private Set<Role> roles;

@OneToMany(mappedBy = "user")
private List<ClassRegister> classes = new ArrayList<>();

@Nullable
@OneToMany(mappedBy="user")
private List<PlanOrder> planOrders = new ArrayList<>();

@OneToMany(mappedBy="user")
private List<Workout> workouts = new ArrayList<>();

@Override
public String toString() {
    return "User{" +
        "id=" + id +
        ", firstName='" + firstName + '\'' +
        ", lastName='" + lastName + '\'' +
        ", email='" + email + '\'' +
        ", username='" + username + '\'' +
        ", password='" + password + '\'' +
        ", subscribed='" + subscribed + '\'' +
        ", enabled=" + enabled +
        ", roles=" + roles +
        ", classes=" + classes +
        ", workouts=" + workouts +
        '}';
}
```