

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

ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Кафедра автоматизованих систем обробки інформації та управління

До захисту допущено:

В.о. завідувача кафедри

(підпис) Олександр ПАВЛОВ
(вл.ім'я, прізвище)

“ ____ ” _____ 2021 р.

Дипломний проєкт
на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Інформаційні управляючі
системи та технології»
спеціальності 126 «Інформаційні системи та технології»**

**на тему: «Система інформаційної підтримки діяльності
спортивної секції»**

Виконав:

студент IV курсу, групи ІС-72

Шаповал Валентин Вадимович

(прізвище, ім'я, по батькові)

(підпис)

Керівник

доц., к.т.н. Жураковська Оксана Сергіївна

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

(підпис)

**Консультант з
графічної
документації**

доц., к.т.н. Новінський Валерій Петрович

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

(підпис)

Рецензент

доц. каф. ТК ФІОТ, к.т.н., доц. Лісовиченко

Олег Іванович

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

(підпис)

Засвідчую, що у цьому дипломному проєкті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____
(підпис)

Київ – 2021 року

Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”

Факультет (інститут) _____ інформатики та обчислювальної техніки
(повна назва)

Кафедра _____ *автоматизованих систем обробки інформації та управління*
(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційні управляючі системи та технології»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

(підпис) Олександр ПАВЛОВ
(вл.ім'я, прізвище)

“ ____ ” _____ 2021 р.

ЗАВДАННЯ
на дипломний проєкт студенту

Шаповалу Валентину Вадимовичу
(прізвище, ім'я, по батькові)

1. Тема проєкту «Система інформаційної підтримки діяльності
спортивної секції»

керівник проєкту _____ Жураковська Оксана Сергіївна, к.т.н.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “11” травня 2021 р. № 1139-с

2. Термін подання студентом проєкту “04”червня 2021 року

3. Вихідні дані до проєкту

Технічне завдання

4. Зміст пояснювальної записки

1. Загальні положення: основні визначення та терміни, опис предметного
середовища, огляд ринку програмних продуктів, постановка задачі

2. Інформаційне забезпечення: вхідні дані, вихідні дані, опис структури бази даних

3. Математичне забезпечення: змістовна та математична постановки задачі,
обґрунтування та опис методу розв'язання

4. Програмне та технічне забезпечення: засоби розробки, вимоги до
технічного забезпечення, архітектура програмного забезпечення, побудова звітів

5. Технологічний розділ: керівництво користувача, методика випробувань
програмного продукту

5. Перелік графічного матеріалу

1. *Схема структурна варіантів використань*

2. *Схема структурна станів системи*

3. *Схема структурна бази даних*

4. *Схема структурна класів програмного забезпечення*

5. *Креслення вигляду екранних форм*

6. Консультанти розділів проєкту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання «7» квітня 2021 року

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1.	<i>Вивчення рекомендованої літератури</i>	<i>13.04.2020</i>	
2.	<i>Аналіз існуючих методів розв'язання задачі</i>	<i>14.04.2020</i>	
3.	<i>Постановка та формалізація задачі</i>	<i>15.04.2020</i>	
4.	<i>Розробка інформаційного забезпечення</i>	<i>16.05.2020</i>	
5.	<i>Алгоритмізація задачі</i>	<i>18.04.2020</i>	
6.	<i>Обґрунтування використовуваних технічних засобів</i>	<i>19.04.2020</i>	
7.	<i>Розробка програмного забезпечення</i>	<i>04.05.2020</i>	
8.	<i>Налагодження програми</i>	<i>07.05.2020</i>	
9.	<i>Виконання графічних документів</i>	<i>09.05.2020</i>	
10.	<i>Оформлення пояснювальної записки</i>	<i>12.05.2020</i>	
11.	<i>Подання ДП на попередній захист</i>	<i>14.05.2021</i>	
12.	<i>Подання ДП на основний захист</i>	<i>04.06.2021</i>	
13.	<i>Подання ДП рецензенту</i>	<i>07.06.2021</i>	

Студент

Валентин ШАПОВАЛ

Керівник

Оксана ЖУРАКОВСЬКА

[illegible]

Пояснювальна записка до дипломного проєкту

на тему: «Система інформаційної підтримки діяльності
спортивної секції»

Київ – 2021 року

АНОТАЦІЯ

Структура та обсяг роботи. Пояснювальна записка дипломного проєкту складається з п'яти розділів, містить 20 рисунків, 3 таблиці, 1 додаток, 10 джерел.

Дипломний проєкт присвячений розробці системи інформаційної підтримки секції рукопашного бою. Метою розробки є підвищення ефективності роботи секції за рахунок її раціональної організації та забезпечення зручної взаємодії користувачів та адміністрації, враховуючи режим пандемії.

У розділі інформаційного забезпечення наведено опис вхідних, вихідних даних та структури бази даних.

Розділ математичного забезпечення присвячений вирішенню задачі організації групи учнів-спортсменів при підготовці до змагань, враховуючи їх перспективність та реально можливі ресурси тренера. Дано огляд методів розв'язання. Наведено розв'язання задачі, приклади вирішення.

У розділі програмне забезпечення наведено засоби розробки, основні компоненти архітектури: діаграму класів, послідовності, компонентів та розгортання.

У технологічному розділі подана інструкція користувача, здійснений опис тестування програмного продукту.

Перелік ключових слів: СПОРТИВНА СЕКЦІЯ, КЛІЄНТ-СЕРВЕРНА АРХІТЕКТУРА, ЗАДАЧІ КОМБІНАТОРНОЇ ОПТИМІЗАЦІЇ, ЗАДАЧА ПРО РЮКЗАК, ЕВРИСТИЧНІ МЕТОДИ, МЕТОД ГІЛОК ТА МЕЖ, МЕТОД ЛІНІЙНОГО ПРОГРАМУВАННЯ

					ДП 7223.00.000 ПЗ					
Зм.	Арк.	Прізвище	Підпис	Дата	Система інформаційної підтримки діяльності спортивної секції			Літ.	Лист	Листів
Розроб.		Шаповал В.В.								
Перевірив.		Жураковська О. С.							2	65
								КПІ ім. Ігоря Сікорського Каф. АСОІУ		
Н. кон.		Новінський В.П.								
Затв.		Павлов О.А.								

ABSTRACT

Structure and scope of work. The explanatory note of the diploma project consists of five sections, contains 20 figures, 2 tables, 2 appendices, 7 sources.

The diploma project is devoted to the development of the information support system of the hand-to-hand combat section. The purpose of the development is to increase the efficiency of the section due to its rational organization and to ensure convenient interaction between users and administration, taking into account the pandemic regime.

The information support section describes the input, output data and database structure.

The section of mathematical support is devoted to solving the problem of organizing a group of student-athletes in preparation for competitions, taking into account their prospects and the real possible resources of the coach. An overview of solution methods is given. The solution of the problem, examples of the solution are given.

The software section provides development tools, key components of the architecture: class diagram, sequence, components, and deployment.

In the technological section the user's instruction is given, the description of testing of a software product is carried out.

List of keywords: SPORT CLASSES, WEB APPLICATION, CLIENT-SERVER ARCHITECTURE, COMBINARY OPTIMIZATION PROBLEMS, BACKPACK PROBLEM, HEURISTIC METHODS, BRANCH AND BOUNDARY METHODS, LINEAR PROGRAMMING METHODS

ЗМІСТ

ВСТУП.....	6
1 ЗАГАЛЬНІ ПОЛОЖЕННЯ	7
1.1 ОПИС ПРЕДМЕТНОГО СЕРЕДОВИЩА	7
1.2 ОГЛЯД АНАЛОГІВ ІСНУЮЧИХ РІШЕНЬ.....	7
1.3 ОПИС ФУНКЦІОНАЛЬНОЇ МОДЕЛІ.....	9
1.5 ПОСТАНОВКА ЗАДАЧІ.....	11
1.5.1 Призначення розробки	11
1.5.2 Цілі та задачі розробки	11
Висновок до розділу	12
2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ	13
2.1 ВХІДНІ ДАНІ	13
2.2 ВИХІДНІ ДАНІ	13
2.3 ОПИС СТРУКТУРИ БАЗИ ДАНИХ.....	14
Висновок до розділу	15
3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ.....	16
3.1 ЗМІСТОВНА ПОСТАНОВКА ЗАДАЧІ.....	16
3.2 МАТЕМАТИЧНА ПОСТАНОВКА ЗАДАЧІ	16
3.3 ОБҐРУНТУВАННЯ МЕТОДУ РОЗВ’ЯЗАННЯ	17
3.4 ОПИС МЕТОДІВ РОЗВ’ЯЗАННЯ	20
Висновок до розділу	24
4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ.....	25
4.1 ЗАСОБИ РОЗРОБКИ.....	25
4.2 ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ	27
4.2.1 Загальні вимоги	27
4.3 АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	28
4.3.1 Діаграма класів.....	28
4.3.2 Діаграма послідовності	29
4.3.3 Діаграма компонентів	31
4.3.4 Діаграма розгортання.....	31
4.3.5 Специфікація функцій.....	32

Висновок до розділу	34
5 ТЕХНОЛОГІЧНИЙ РОЗДІЛ	35
5.1 Керівництво користувача	35
5.2 Випробування програмного продукту	40
5.2.1 Мета випробувань	40
5.2.2 Загальні положення	40
5.2.3 Результати випробувань	41
Висновок до розділу	41
ЗАГАЛЬНІ ВИСНОВКИ	43
ПЕРЕЛІК ПОСИЛАНЬ	44
ДОДАТОК А	45

ВСТУП

Забезпечення власної безпеки і розвиток та підтримка фізичних навичок завжди залишається актуальною. Існує багато секцій з різними напрямленнями бойових мистецтв, розподілені за віковими, гендерними та іншими критеріями. Ці розподілення не є обов'язковими, та при правильній організації знання можна донести до кожного, хто їх прагне (будь то спортивний сегмент, напружені на перемогу у змаганнях або оздоровчий). Якщо у вас є навички вижити у екстремальній ситуації: швидке прийняття рішень, стійкість до неочікуваних джерел небезпеки та вміння володіти своїм тілом у всіх його станах (хворим, втомленим, травмованим і т. і.) дає зброю, яку не можна побачити та не можливо відібрати, доки людина жива.

У сучасному світі навчальна діяльність у всіх сферах має високий рівень автоматизації. Тому саме полегшення та підвищення ефективності роботи спортивної секції і є предметом розробки даного проекту. Існують різні фактори проведення занять: групові заняття, індивідуальні (у невеликих групах та персональні), семінари тощо. Необхідно розробити інформаційну систему із якомога більш зручним інтерфейсом та функціями для підтримки діяльності секції, зокрема організації розкладу занять, перегляду навчально-методичних матеріалів та проведенню онлайн занять.

Практичне значення одержаних результатів. Розроблено систему для інформаційної підтримки діяльності спортивної секції.

					ДП 7223.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Опис предметного середовища

Робота спортивної секції полягає у проведенні занять у спортивному залі, перегляд навчально-методичного матеріалу, залучення та знайомство з новими учнями шляхом їх систематизації (за відсутності інформаційної системи – принаймні шляхом паперових анкет). Формування розкладу занять – вимагає постійної комунікації між учнем та тренером для знаходження найбільш зручного місця, часу та умов для обох сторін.

У спортивній секції, яка є об'єктом автоматизації заняття проходять: як регулярна секція на 20-40 учнів тричі на тиждень у визначений час, індивідуальні заняття в невеликих групах та персональні заняття. Також один два рази на рік може проводитись семінар на декілька днів. Проведення онлайн занять на момент початку розробки системи інформаційної підтримки не передбачені.

Оплата занять здійснюється по факту їх проведення.

Отримання основної більшості методичного матеріалу, розгорнутий зворотній зв'язок відбувається після заняття і, зазвичай, у паперовому вигляді.

1.2 Огляд аналогів існуючих рішень

Розглянемо існуючі сайти для зберігання та демонстрації медалей:

1) Веб-застосування Mobitfitness [7]

Переваги:

- облікова система для управління;
- мобільний додаток для комунікації з клієнтом;
- онлайн запис на заняття;
- онлайн оплата.

Недоліки:

- не працює у режимі оффлайн;

					ДП 7223.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

- дорога у розробці;
- не передбачені заняття онлайн.

2) Веб-сайт Nikko [8]

Переваги:

- невелика система лояльності у вигляді безкоштовного першого заняття;
- зворотній зв'язок у вигляді бота для комунікації з клієнтом;
- фото-, відеогалерея.

Недоліки:

- інтерфейс не дуже зрозумілий для використання;
- не передбачені заняття онлайн;
- не працює у режимі оффлайн;
- немає можливості працювати з розкладом занять.

3) Веб-сайт IGymnasium [9];

Переваги:

- велике розмаїття спортивних напрямлень;
- є фотогалерея та блог новин.

Недоліки:

- не передбачені заняття онлайн;
- незручний, застарілий інтерфейс;
- відсутність навчально-методичного матеріалу;
- відсутня онлайн оплата;
- не працює у режимі оффлайн.

В результаті аналізу існуючих рішень, можна зробити висновок, що розробка системи підтримки діяльності спортивної секції залишається актуальною, враховуючи усі недоліки аналогів та передбачаючи їх усунення.

Система повинна:

- мати зручний інтуїтивно зрозумілий інтерфейс;
- зручну роботу із розкладом занять, як для тренера так і для учнів;

					ДП 7223.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

- зручний перегляд, редагування та коментування контенту, новин спортивної секції та навчально-методиного матеріалу;
- мати сучасні методи клієнт-серверної взаємодії та роботи у режимі оффлайн;
- ведення особистих досягнень учня;
- мати можливість ведення онлайн занять.

1.3 Опис функціональної моделі

У системі наявні три актори: учень, адміністратор та тренер. Дійові особи та варіанти використання, які вони утилізують в межах розробленого програмного продукту наведені у таблиці 1.1:

Таблиця 1.1 – Опис варіантів використання

Актор	Варіант використання	Опис дії варіанта використання
Учень	Авторизація Реєстрація	Користувач має авторизуватися у системі для перегляді персональної інформації та більшості інших активностей. (актори тренер та адміністратор також мають цей варіант використання)
	Відновлення паролю	(актори тренер та адміністратор також мають цей варіант використання)

Продовження таблиці 1.1

					Перегляд умов навчання	
					Перегляд розкладу	
					Подання заявки на індивідуальне заняття	
					Відмітити заняття заплановане для відвідування	
					Оплата заняття	Із використанням усіх сучасних платіжних систем
					Проходження онлайн заняття	Реалізовані: відеозв'язок, чат та аудіозв'язок, увесь функціонал системи Zoom
					Перегляд навчально-методичного матеріалу	
					Перегляд особистих досягнень	
Тренер					Проведення онлайн занять	
					Ведення учнів	Додання нових, бан існуючих та редагування особистих

Змн.	Арк.	№ докум.	Підпис	Дата

Кінець таблиці 1.1

	Перегляд статистичної інформації про учнів	
	Формування розкладу	
	Призначення індивідуальних занять	Підтвердження обраних учнем індивідуальних занять
Адміністратор	Ведення тренерів	Додання, бан та редагування тренерів
	Редагування контенту	Редагування фото відео та бібліотеки
	Модерація коментарів	

1.4 Постановка задачі

1.4.1 Призначення розробки

Призначенням розробки є підтримка діяльності спортивної секції.

1.4.2 Цілі та задачі розробки

Цілі розробки:

Метою розробки є підвищення ефективності роботи секції за рахунок її раціональної організації та забезпечення зручної взаємодії користувачів та адміністрації.

1.4.3 Задачі розробки

Для забезпечення мети розробки необхідно реалізувати такі задачі:

- розробка UI/UX інтерфейсу;
- розробка бази даних;

- реалізація можливості користувачу реєстрації та авторизації, та відновлення паролю;
- створення та наповнення галереї, відеотеки та бібліотеки навчально-методичним контентом, додання можливості коментування;
- ведення тренерів, учнів;
- складання розкладу занять секції;
- покращення позицій у пошукових системах;
- реалізація проведення занять в онлайн режимі;
- розробка програмного забезпечення.

Висновок до розділу

У даному розділі було наведено опис предметного середовища спортивної секції, цілі та головні задачі розробки, опис функціональної моделі.

Було розглянуто аналоги існуючих рішень, виявлено їх недоліки та переваги. Показана актуальність розробки системи.

У розділі було надано опис дійових осіб та їх головні призначення. За допомогою структурної схеми використань можна побачити функції, які будуть виконуватися дійовими особами.

					ДП 7223.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Вхідні дані

Вхідні дані для задачі підтримки діяльності спортивної секції:

Якщо користувач проходить авторизацію у якості учня, вхідними даними для системи мають бути:

- особиста інформація про користувача, у разі проходження реєстрації (email, ПІБ, номер телефону, пароль);
- час та місце проведення заняття, вибір методу оплати;

Якщо користувач проходить авторизацію у якості тренера вхідними даними для системи можуть бути:

- запит на конкретного учня, для перегляду його даних, статусу оплати заняття і його досягнення;
- список учнів-кандидатів на участь у змаганнях.

2.2 Вихідні дані

Вихідними даними системи для клієнта мають бути:

- інформація про учнів (у вигляді його рівень досягнень, обумовлений його регулярністю та тривалістю відвідувань занять, також доповнений тренером);
- інтерфейс для проведення онлайн уроку, за умов успішно підтвердженого його статусу оплати;
- чек по оплаті, за необхідності відправлений на електронну пошту;
- навчально-методичні матеріали (відео, галерея, бібліотека).

Вихідними даними системи для тренера мають бути:

- графік прибутку школи та відвідувань;
- список досягнень, статусу оплати та статистика відвідування кожного

					ДП 7223.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

конкретного учня;

- список учнів, що будуть приймати участь у змаганнях.

Для адміністратора вихідними даними системи є відредагований вид контенту, списку коментарів та список тренерів.

2.3 Опис структури бази даних

У системі буде використовуватися noSQL база даних MongoDB, яка розміщена на хмарному сервісі Cloud DB. Для побудови бази даних була спроектована ER-діаграма (рис. 2.1). Запити буде виконуватися на сервері, а клієнтське відображення контенту або частину з веденням учнів будуть реалізовані на технології GraphQL, яка дозволить не навантажувати мережу зайвими об'єктами даних.

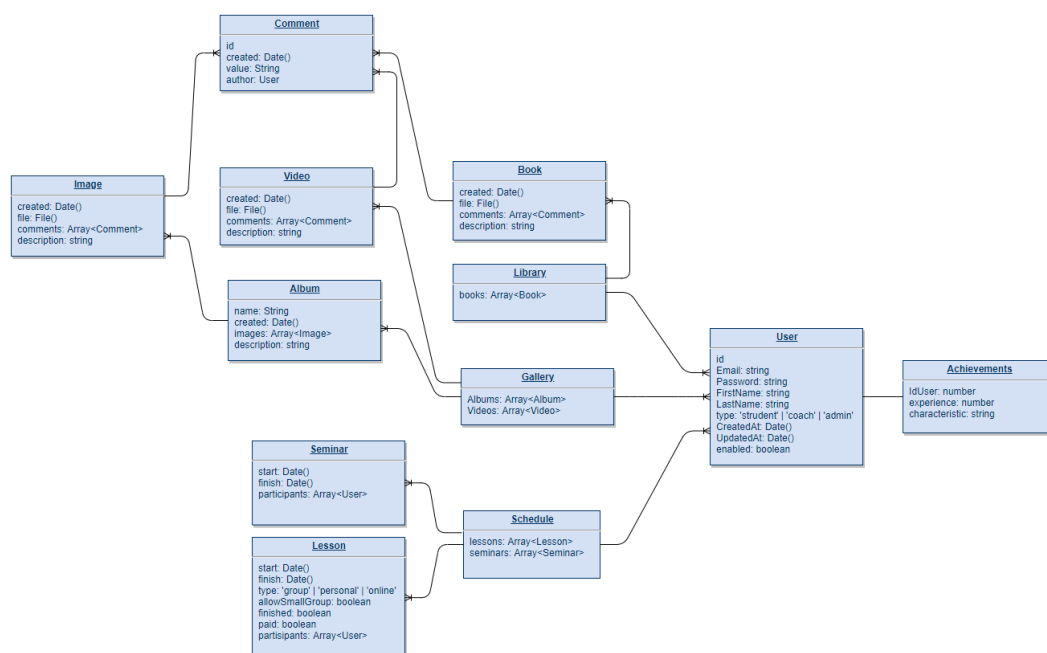


Рисунок 2.1 – ER діаграма бази даних

На наведеній діаграмі актори учень, тренер та адміністратор представлені у сутності User яка пов'язана із сутностями контенту, розкладу занять та системи особистих досягнень. Контент представлений у вигляді галереї з альбомів з фотографіями, відео та бібліотеки з книжками і буклетами.

Висновок до розділу

У розділі були представлені структури вхідних та вихідних даних, з якими працює побудована система. Наведено схему poSQL бази даних із полями та їх типами, також описані методи взаємодії з ними.

					ДП 7223.00.000 ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Змістовна постановка задачі

Є множина учнів, кожний з яких має власний рівень вправності та темпи розвитку здібностей, визначені тренером та досвід. Ресурси тренера обмежені: учням з меншим досвідом необхідно приділити більше уваги, кількість більш досвідчених мають бути обмежена для більш ефективного розвитку за рахунок створення індивідуального графіку, навантаження, системи харчування і т. і. Задача полягає у досягненні максимальної продуктивності заняття і максимізувати перспективність групи. Дана задача відноситься до NP-складних.

На стадії підготовки тренер обирає найкращих учнів для проведення семінару з підготовки до спортивних змагань. Обрана група обмежена ресурсами часу тренера, який він може приділити кожному учню. Серед різних критеріїв необхідного ресурсу, який виділяється на учня існують: досвід спортивної кар'єри, досвід співпраці з цим тренером, швидкість здобуття навичок учнем, фізичні параметри (у тому числі травми) і таке інше.

Задача полягає у підборі такого складу групи, щоб досягти максимальної ефективності групи при участі в змаганнях і при цьому врахувати обмежені ресурси. Для побудови математичної моделі можемо звести критерії ресурсу необхідного на учня до одного значення, яке визначається однаковим чином для кожного учня за заданою тренером бізнес логікою, винісши таким чином нюанси предметної області на нижчий рівень абстракції.

3.2 Математична постановка задачі

Є множина із n учнів, перспективність кожного учня j , $j = 1, 2, \dots, n$ при участі в змаганнях виражається ваговим коефіцієнтом $p_j > 0$. Кожен учень для тренування потребує часу та зусиль тренера по його підготовці, що може бути

					ДП 7223.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

виражена ваговим коефіцієнтом $w_j > 0$. Сукупний ресурс тренера по роботі з учнями не може перевищувати $U > 0$. Таким чином обмеження на роботу тренера з учнями може бути сформульоване у вигляді

$$\sum_{j=1}^n w_j > U,$$

$$\forall w_j < U, j = 1, 2, \dots, n$$

Введемо такі булеві змінні $x_j \in \{0, 1\}, j = 1, 2, \dots, n$.

При $x_j = 1$ учень приймає участь у тренуванні, при $x_j = 0$ не приймає.

Необхідно сформувати групу таким чином, щоб максимізувати сумарна ефективність групи за виконанням умов:

$$f(x) = \sum_{j=1}^n p_j x_j \rightarrow \max,$$

$$\sum_{j=1}^n w_j x_j \leq U,$$

$$x_j \in \{0, 1\}, j = 1, 2, \dots, n$$

3.3 Обґрунтування методу розв'язання

Задача, що розглядається є задачею про одновимірний рюкзак (knapsack problem), є NP-складною і може бути розв'язана методами: повного перебору, евристичним методом, методом гілок та меж, метаевристичними та методами динамічного програмування [3].

Метод повного перебору

Розмірність цієї задачі рівна n , звідки число допустимих розв'язків складатиме 2^n . Отже метод повного перебору не підходить для даної предметної області, оскільки задача може бути вирішена лише при невеликих

розмірностях. Тому, зазвичай, для її вирішення застосовують інші методи, що скорочують перебір.

Евристичні алгоритми

Таким, наприклад, може бути евристичний (жадібний) алгоритм. Його робота полягає у сортуванні учнів за спаданням певної евристичної функції, наприклад, питомою вартістю (відношенням перспективності до вагового коефіцієнту зусиль, необхідних для цього учня). Після цього виконується покрокове додавання учня до складу команди. Складність цього алгоритму становитиме $O(n \cdot \log(n))$ операцій [1].

Огляд метаевристичних методів

У результаті еволюції евристичних методів виникли так звані метаевристичні, серед них: метод мурашиних колоній (ant colony optimization) та генетичні алгоритми (genetic algorithms).

У випадку з генетичними алгоритмами маємо наступні недоліки: у випадку, коли треба знайти точний глобальний оптимум – велика кількість мутацій не приводить до покращення результату та коли конфігурація є складною (кодування рішення).

У випадку мурашиного алгоритму маємо наступні недоліки: ускладнений теоретичний аналіз (розподіл вірогідностей змінюється при ітераціях). Сходимість гарантується, але час сходимості не визначений. Потребує використання додаткових методів (наприклад, локальний пошук). Велика залежність від параметрів налаштування, котрі обираються виходячи з експериментів. [4]

Методу динамічного програмування

Для початку наведемо визначення алгоритму пошуку оптимальної стратегії запропоноване Річардом Беллманом: яким би не був початковий стан і перше рішення, наступні рішення оптимізують його [3].

					ДП 7223.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

Перевагами методу динамічного програмування є те, що він не накладає ніяких спеціальних вимог на вигляд і форму представлення функцій, що становлять критерій. Відшукування глобального екстремуму багатовимірної функції є дуже складною проблемою. Динамічне програмування знижує розмірність вирішуваних завдань в n раз (по числу змінних), що значно полегшує знаходження глобального екстремуму. Накладення специфічних умов на змінні не «обважнює» рішення задачі методом. Трудомісткість обчислень і обсяг пам'яті для зберігання результатів на одному кроці не залежить від числа кроків. Тому збільшення числа кроків призводить тільки до пропорційного збільшення часу рішення задачі, тоді як для інших методів зі збільшенням розмірності задачі трудомісткість її рішення зростає набагато швидше [11].

Саме цей метод буде застосований у програмній реалізації проекту. Вибір зроблений на основі того, що метод дійсно знаходить оптимальний розв'язок, робить це за достатньо малий час і відносно простий у реалізації.

Метод гілок та меж

Метод знаходить оптимальне рішення, в основі своєї роботи має три принципи: розгалуження, верхню межу та нижню межу. Розгалуження – розбиття вихідної задачі на підзадачі і будуючи таким чином дерево пошуку. Це рекурсивний метод, який дозволяє вирішувати задачу з уже присутнім частковим рішенням або бажаними вхідними даними. Верхня межа використовується для оцінки перспективності підзадачі, тобто ймовірність наявності оптимального рішення у цій гілці. Нижня межа використовується для відсікання неоптимальних гілок.

Головний недолік алгоритму методу гілок і меж полягає в необхідності повністю вирішувати завдання лінійного програмування, асоційовані з кожної з вершин багатогранника допустимих рішень. Для задач великої розмірності

це вимагає значних і, до певної міри, невинуватених з практичної точки зору витрат часу.

Метод гілок та меж було обрано як основний, оскільки він дозволяє знайти оптимальне рішення, при цьому істотно скорочує діапазон пошуку. Для задач великої розмірності було також реалізовано евристичний метод.

3.4 Опис методів розв'язання

Першим способом вирішення задачі буде реалізація евристичного (жадібного) алгоритму. Нехай ми надаємо перевагу учням з максимальною питомою вартістю часового ресурсу тренера $\lambda_j = \frac{p_j}{w_j}$. Тоді порахувавши для кожного предмета λ_j , впорядкуємо предмети за спаданням питомої вартості ($\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$). Далі просто будемо додавати у порядку спадання у відповідь із усіх існуючих аж доки це можливо. Даний алгоритм має складність $O(m \cdot n^2)$.

Для розглянення вирішення задачі предметної області методом гілок та меж оберемо алгоритм Данцига. Спочатку, як і у евристичному алгоритмі, сортуємо учнів за спаданням λ_i . Галуження відбувається доданням або виключенням учня до розв'язку. Далі додаємо предмети, доки не вміститься один і розраховуємо для нього верхню межу за формулою:

$$UB = Z + \sum_{k=1}^{j-1} p_k x_k$$

Складність такого алгоритму становить $O(x^n)$, тобто є експоненційною.

Тепер перейдемо до розгляду застосування методу динамічного програмування. Для його коректної роботи має виконуватися умова p_i, w_i та $U \in \mathbb{Z}^+$. По-друге, алгоритм є покроковим, на кожному кроці j приймається рішення, чи буде учень брати участь у занятті. Рішення складається для кожного стану s , який дає характеристику максимально

					ДП 7223.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

можливих умов заняття для учнів $i = 1, 2, \dots, j$. Таким чином, при включенні учнів на наступних кроках загальною вагою вимог до заняття $\sum_{i=j+1}^n x_i w_i$, то для учнів $i = 1, 2, \dots, j$ в нас залишається вільним $s = U - \sum_{i=j+1}^n x_i w_i$ од. «вимог». Отже, враховуємо майбутні стани системи. Функція вигідності для цієї задачі буде така:

$$g_j(s) = \max_{x_j \in \{0,1\}} (p_j x_j + g_{j-1}(s - w_j x_j)), s \geq w_j x_j$$

Якщо $s \leq w_j$, то $x_j = 0$.

Приклад розв'язання задачі

$$\begin{aligned} f(x) &= 3x_1 + 7x_2 + 2x_3 + 6x_4 \rightarrow \max \\ 3x_1 + 8x_2 + 4x_3 + 5x_4 &\leq 11 \end{aligned}$$

Маємо чотири учні, тоді:

$$\text{Початкова умова UB}\{1\} = 3 + 7 + 2 \frac{11-3-8}{4} = 10.5$$

Переходимо до гілки 2, де $x_1 = 1$, $\text{UB}\{2\} = 3 + 7 + 2 \frac{11-3-8}{4} = 10.5$, та гілки 3, де $x_1 = 0$, $\text{UB}\{2\} = 7 + 2 \frac{11-8}{4} = 8.75$. Поки що підемо по гілці 2, у гілці 4 при $x_2 = 1$ маємо $\text{UB}\{4\} = 3 + 7 + 2 \frac{11-3-8}{4} = 10.5$, і $\text{UB}\{5\} = 3 + 2 + 6 \frac{11-3-2}{5} = 12.2$ для $x_2 = 0$ відповідно.

Для розгалуження обираємо гілку {5}, але в ній існує тільки розв'язок (1,0,1,0) із значенням цільової функції 5.

Повернемось до гілки {4} для неї єдиним можливим рішенням є (1, 1, 0, 0) із значенням цільової функції 11.

Повернемось, нарешті, до гілки {3} для якої існує розгалуження:

гілка {6} $x_2 = 1$, єдине рішення – (0, 1, 0, 0), ЦФ = 7

гілка {7} $x_2 = 0$, єдине рішення – (0, 0, 1, 1), ЦФ = 8

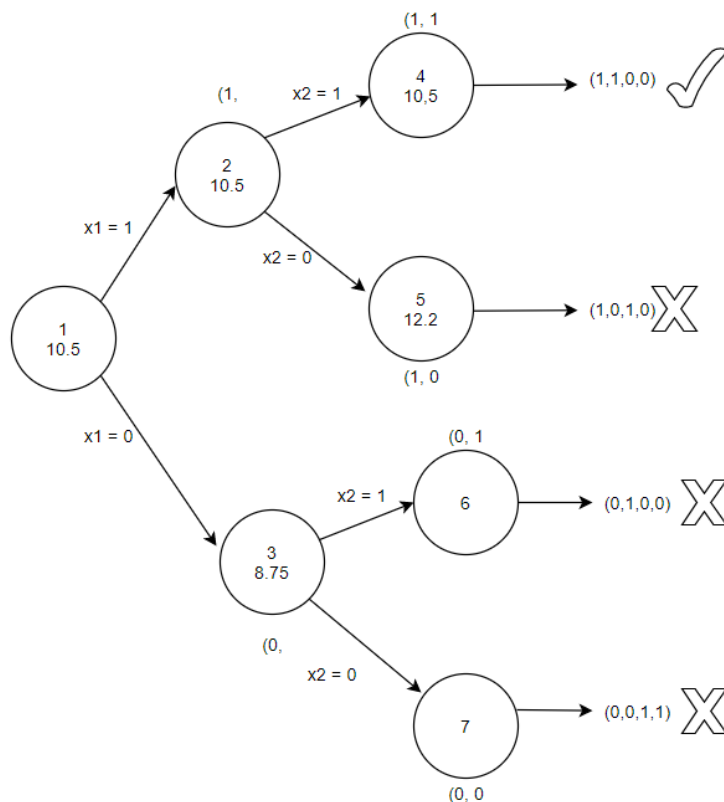


Рисунок 3.1 – Графічне зображення дерева галуження

Спробуємо розв'язати задачу з чотирма учнями (1) методом динамічного програмування.

Таблиця 3.1 - Кроки розв'язку

s	$g_1(s)$	$x(s)$	$g_2(s)$	$x(s)$	$g_3(s)$	$x(s)$	$g_4(s)$	$x(s)$
0	0	(0, , ,)	0	(0, 0, ,)	0	(0, 0, 0,)	0	(0, 0, 0, 0)
1	0	(0, , ,)	0	(0, 0, ,)	0	(0, 0, 0,)	0	(0, 0, 0, 0)
2	0	(0, , ,)	0	(0, 0, ,)	0	(0, 0, 0,)	0	(0, 0, 0, 0)
3	3	(1, , ,)	3	(1, 0, ,)	3	(1, 0, 0,)	3	(1, 0, 0, 0)
4	3	(1, , ,)	3	(1, 0, ,)	3	(1, 0, 0,)	3	(1, 0, 0, 0)
5	3	(1, , ,)	3	(1, 0, ,)	3	(1, 0, 0,)	6	(0, 0, 0, 1)
6	3	(1, , ,)	3	(0, 0, ,)	3	(1, 0, 0,)	6	(0, 0, 0, 1)
7	3	(1, , ,)	3	(1, 0, ,)	7	(1, 0, 1,)	7	(1, 0, 1, 0)
8	3	(1, , ,)	7	(0, 1, ,)	7	(0, 1, 0,)	9	(1, 0, 1, 0)
9	3	(1, , ,)	7	(0, 1, ,)	7	(0, 1, 0,)	9	(1, 0, 0, 1)
10	3	(1, , ,)	7	(0, 1, ,)	7	(0, 1, 0,)	9	(1, 0, 0, 1)
11	3	(1, , ,)	10	(1, 1, ,)	10	(1, 1, 0,)	10	(1, 1, 0, 0)

Алгоритм швидкий і не потребує сортування даних. Складність його легко обчислити, і вона складає $O(nU)$, де n – кількість кроків (кількість змінних), а U , як пам'ятаємо – максимальне покриття вимог учнів тренером.

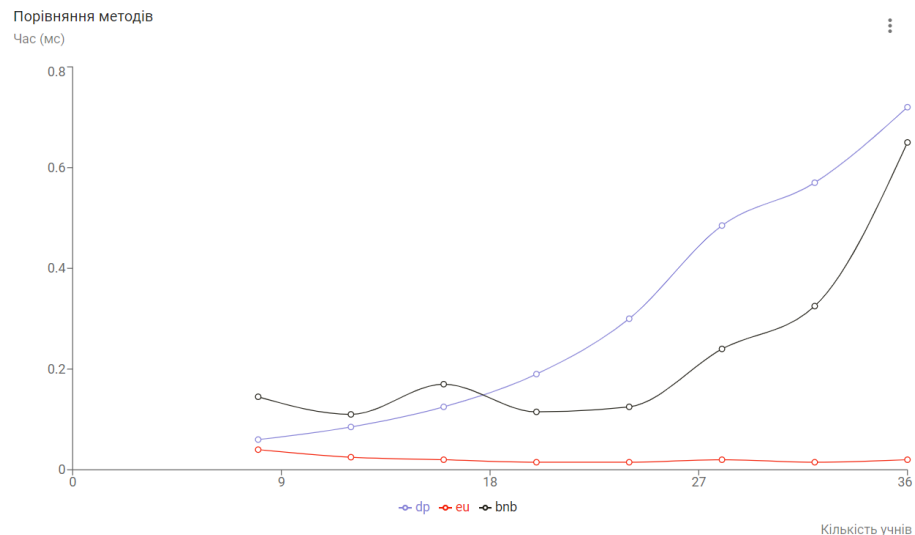


Рисунок 3.2 – Час виконання евристичного методу (червоний колір), методу динамічного програмування (синій колір) та методу гілок та меж (чорний колір)

Проведемо дослідження за цільовою функцією методів гілок та меж та евристичного. Оцінимо Δ -відхилення від оптимального значення для евристичного методу.

$$\Delta = \frac{f(\text{опт}) - f(E)}{f(\text{опт})}$$

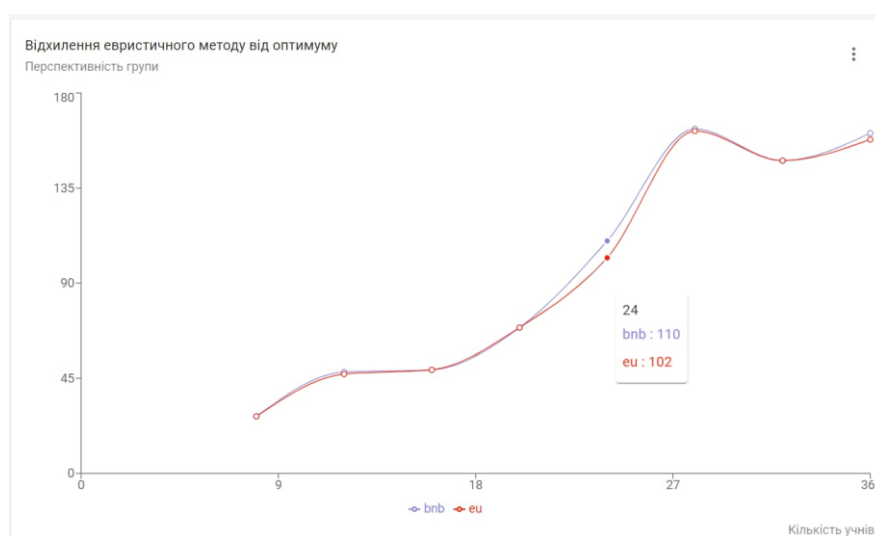


Рисунок 3.3 – Порівняння результатів евристичного алгоритму та методу гілок та меж

Змн.	Арк.	№ докум.	Підпис	Дата

При різних n (кількості учів) Δ складає 8.43% в середньому.

Провівши випробування після програмної реалізації видно, що алгоритм методу гілок та меж дає найкращий часовий результат в разі необхідності отримання точного результату.

Висновок до розділу

У цьому розділі сформовано змістовну та математичну постановки задачі формування групи для участі у тренуванні. Задача сформульована як задача про рюкзак. Наведено опис методів її розв'язання. Були обрані такі методи: метод динамічного програмування, евристичний метод та метод гілок і меж. Наведено приклади розв'язання задачі. Здійснено експериментальні дослідження ефективності застосування вказаних методів до вирішення задачі.

					ДП 7223.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		24

4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

4.1 Засоби розробки

Для розробки програмного забезпечення із клієнт-серверною архітектурою було обрано такі засоби:

- мова програмування: TypeScript – типізоване розширення над мовою JavaScript, збиральник проекту webpack виконує компіляцію з TS у JS, який уже буде виконуватися на браузерному двигуні v8. Чудово підходить для створення Front-end та Back-end частин. Типізація дозволяє уникнути додавання примісей (додавання нових полів у об'єкт) та передавання параметрів неправильного типу – відсіює таким чином велику частину помилок виконання ще на етапі компіляції;

- лінтер програмного коду: ES Lint – написання якісного програмного коду, який має відповідати сучасним вимогам та у строки, які визначаються новітніми темпами розробки програмного забезпечення неможливе без використання так званого лінера та prettier – автоматичного форматування коду. У поєднанні вони дозволяють не лише відповідати примітивним стандартам мови (спосіб розставлення дужок, перенесення коду при перевищенні максимально допустимої довжини рядка і т. і.), а й відстежують, особливо у поєднанні з TypeScript правильність описаних типів та взаємодію програмних компонентів. Відбувається це прямо у редакторі за допомогою підсвічення таких як warning та error, а форматування просто при кожному збереженні вихідного файлу;

- система контролю версій Git, та сервіс для зберігання та роботи з версіями програмного коду Github;

- MVC-фреймворк для написання фронт-енду – React.js [5]. Мінімалістичний і дуже потужний інструмент у якому реалізований так

званий віртуальний DOM і при зміні React автоматично порівнює і вносить зміни до реального DOM;

- Для роботи зі стейтом застосування була використана бібліотека `redux` та миддлвейр для неї – `redux saga`. Вони виконують роль Controller у MVC і дозволяють працювати з імутабельним сховищем;

- Для реалізації SSR (Server Side Rendering) із фреймворком React була використана бібліотека `Next.js` [6]. Річ у тім, що Single Page Application не видає цілісну статичну сторінку при першому запиті, і це дуже погано впливає на роботу пошукових роботів і позицію програмного продукту у видачі пошукових систем. Next підготує статику на сервері Node.js, і поверне її клієнту/пошуковому роботу;

- Для створення PWA (Progressive Web App) була використана бібліотеки `nuxt/pwa` та `workbox` від гугл. PWA дозволяє веб-застосування бути встановленим на персональний комп'ютер або смартфон користувачам із будь-якою операційною системою.

- Для реалізації запитів до бекенду використовується бібліотека GraphQL

- Back-end частина була написана з використанням фреймворку для Node.js, `Nest.js` [7] – дуже вдалого продукту, який правильно працює із шарами абстракції застосування.

- Платформою для розгортання програмного забезпечення була обрана `heroku`, завдяки перевіреним досвідом ефективності та зручності інтерфейсу серед усіх безкоштовних аналогів.

- Базою даних була обрана `noSQL` база `MongoDB` і платформа-сховище для неї `mongo-cloud`.

- Редактор коду – Visual Studio Code, сучасний та дуже кастомізований редактор із відкритим вихідним кодом.

4.2 Вимоги до технічного забезпечення

4.2.1 Загальні вимоги

Система має виконувати наступні функції:

- перегляд навчально-методичного матеріалу;
- перегляд розкладу занять;
- оплата заняття онлайн;
- проведення онлайн занять;
- перегляд та редагування системи досягнень учнів;
- забезпечити формування розкладу, створення шоу, створення груп з найвищим коефіцієнтом перспективності.

Дипломний проект повинен відповідати також наступним нефункціональним вимогам:

- кроссбраузерність;
- робота оффлайн у режимі PWA;
- локалізація інтерфейсу як мінімум 3-ома мовами: українською, російською та англійською;
- інтерфейс зрозумілий для користувача;

- навантаження на сервер та мережеве підключення клієнта зменшене за рахунок технології GraphQL та архітектурою «Товстий клієнт – тонкий сервер»;
- дипломний проект можна легко адаптувати під більшість сучасних спортивних секцій.

4.3 Архітектура програмного забезпечення

4.3.1 Діаграма класів

На рис. 4.1 наведена діаграма класів програмного забезпечення.

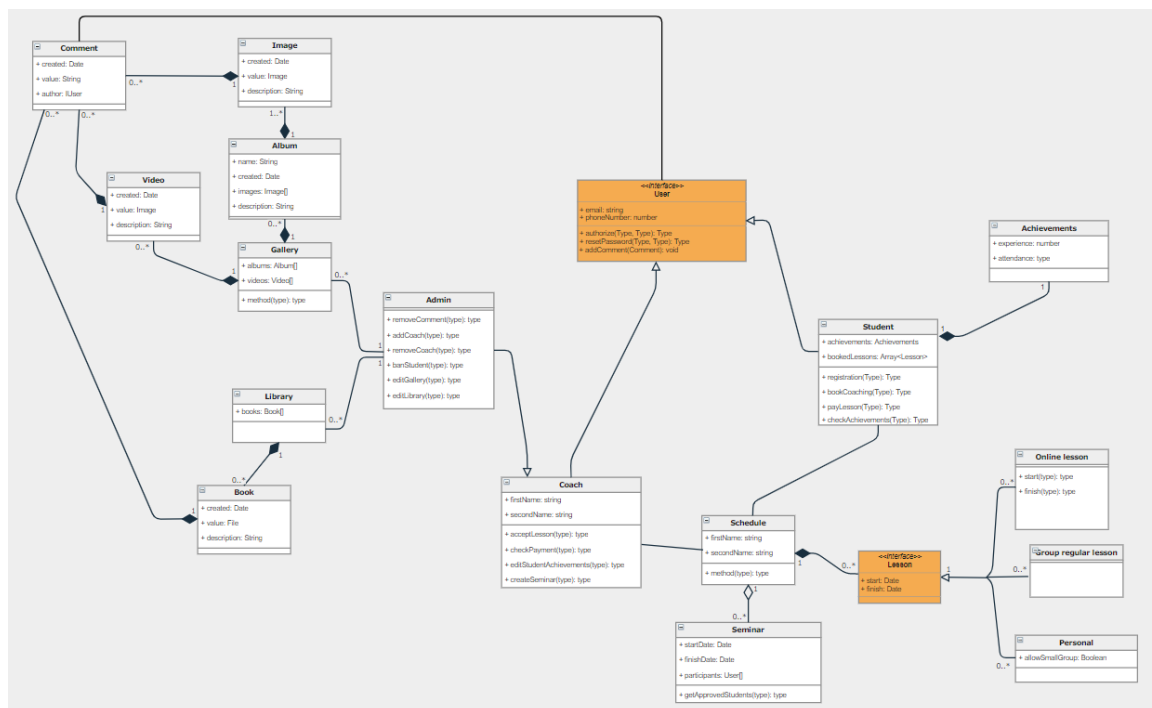


Рисунок 4.1 – Схема структурна класів

Користувача User від інтерфейсу котрого наслідуються класи учень (Student) та тренер (Coach), від тренера наслідується адміністратор(Admin).

User має спільні поля для авторизації і реєстрації, а саме email та password, firstName, lastName. Student має поля achievements, яке зберігає інстанс класу Achievements (досягнення учня) та поле bookedLessons із

заброньованими заняттями для оплати занять, реєстрації та перегляду досягнень.

Тренер має методи для перевірки оплати та редагування досягнень учня. Адміністратор має методи для редагування навчально методичного матеріалу та ведення тренерів.

Тренер та учень пов'язані із класом Schedule, який використовує інтерфейс уроку (Lesson), реалізований класами онлайн уроку (Online lesson), групове заняття у секції (Group regular lesson) та персональні заняття (Personal Lesson). Також до класу розкладу агрегований клас семінарського заняття (Seminar). Усі заняття мають поля дати початку і кінця заняття. А їх інстанси матимуть поля учасників заняття. Персональне заняття дозволяє проводити заняття у маленьких групах (поле allowSmallGroups) по 2-4 учня.

4.3.2 Діаграма послідовності

На рисунку 4.2 наведена діаграма послідовності.

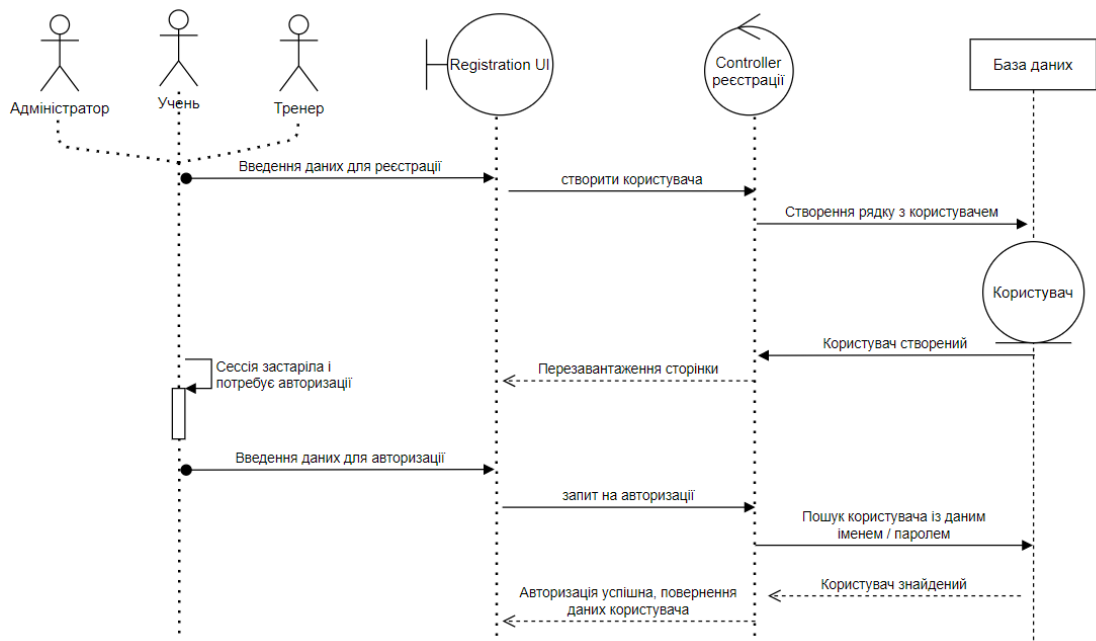


Рисунок 4.2 – Схема структурна послідовності для процесів реєстрації та авторизації

Для роботи із системою та доступу функцій коментування контенту, оплати і відвідування онлайн занять, користувач має авторизуватись або пройти процес реєстрації.

Тренер та адміністратор також мають авторизуватись у системі для виконання своїх функцій. Вони реєстрацію не проходять.

Далі наведено діаграму послідовності для опису процесу онлайн уроку:

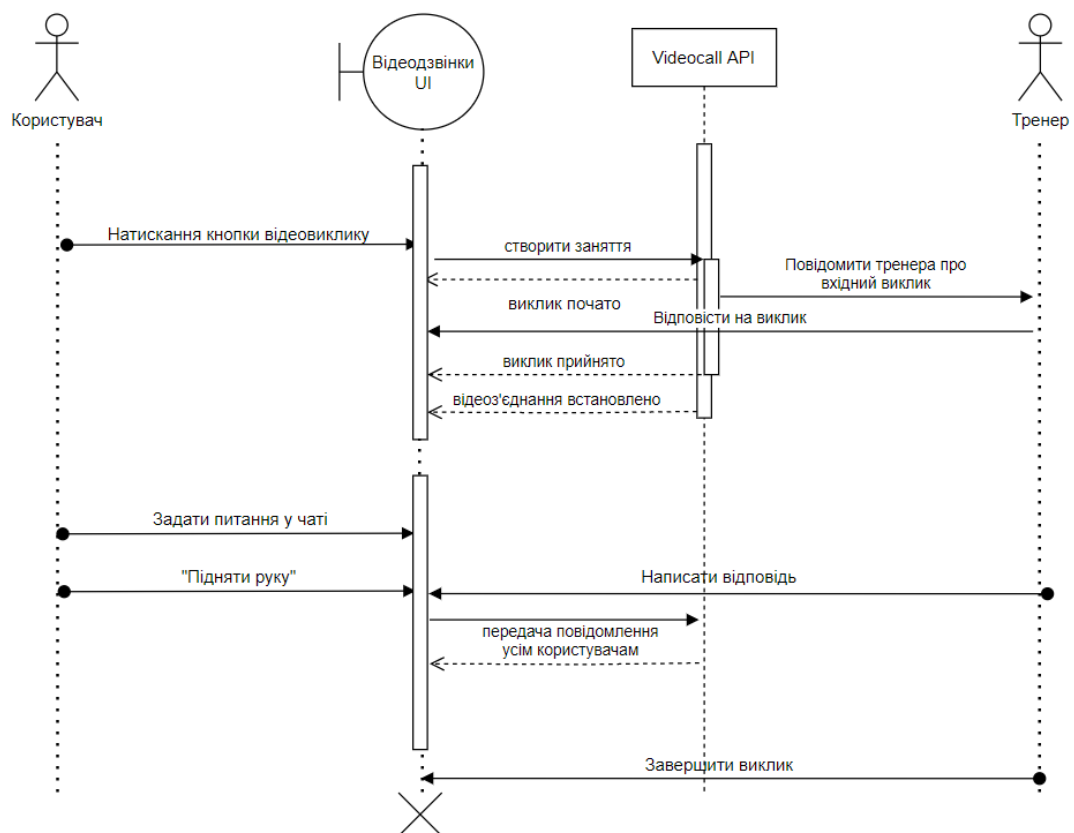


Рисунок 4.3 – Схема структурна послідовності для процесу онлайн уроку

Для проведення онлайн уроків у системі реалізований функціонал відеодзвінків, завдяки чому під час пандемії та локдауну можна залишатися відповідальним громадянином не припиняючи процес підготовки спортсменів.

Функціонал дозволяє здійснювати відеозвінки по протоколу UDP, а також у системі присутній чат і візуальні повідомлення вигляду «учень підняв руку».

4.3.3 Діаграма компонентів

На рисунку 4.4 наведена діаграма компонентів.

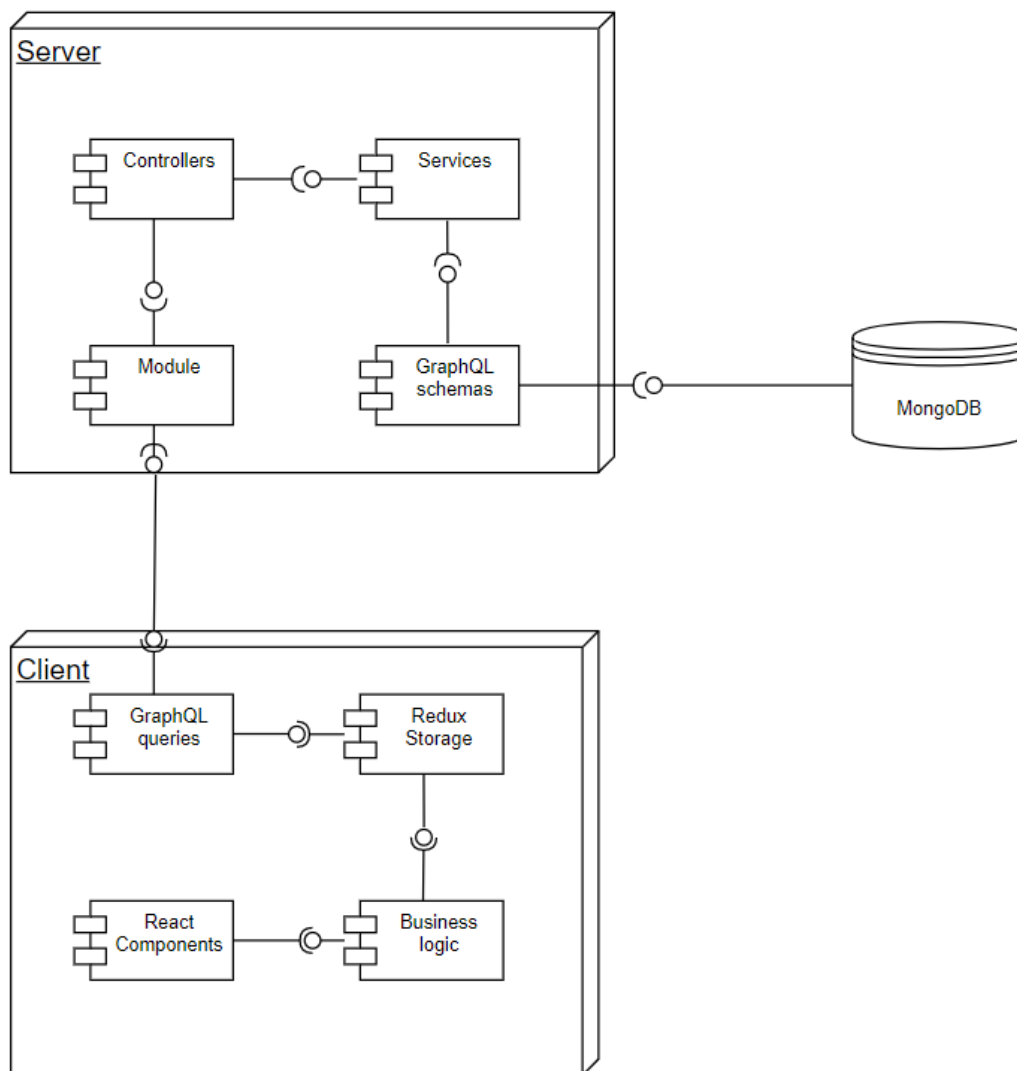


Рисунок 4.4 – Схема структурна компонентів клієнт-серверного застосування

4.3.4 Діаграма розгортання

На діаграмі розгортання (рис. 4.5) показано, як саме працює технологія PWA (Progressive Web App) на практиці. Певні, обрані командою розробки

запити кешуються та, під час відсутнього інтернет з'єднання або перешкодах у його роботі, дані відображаються коректно. Це, у поєднанні із GraphQL запитами суттєво знижує навантаження на трафік.

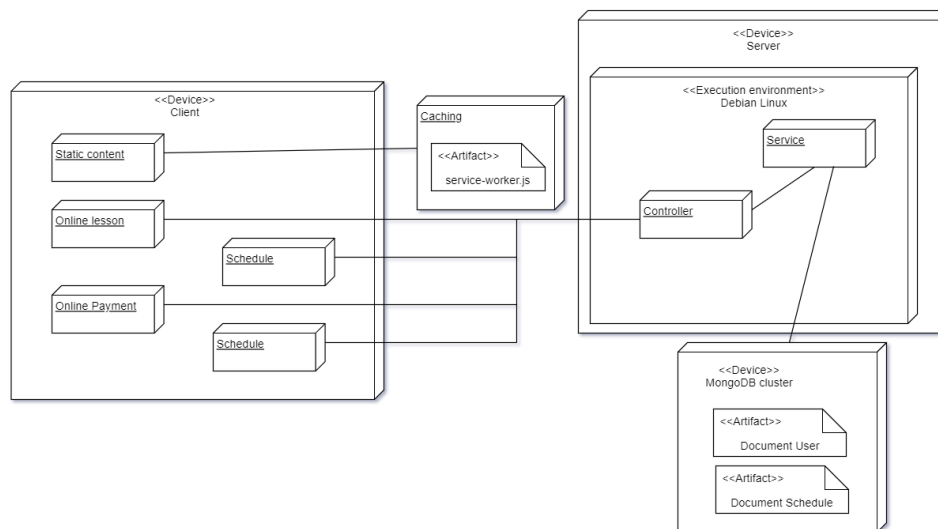


Рис. 4.5 – Схема структурна розгортання клієнт-серверного застосування

4.3.5 Специфікація функцій

Програмний продукт розроблено за змішаною парадигмою. Front-end частина написана із використанням функціональних React-компонентів, API ж побудовано на технології Nest із використанням багатошарової архітектури: сервісів, модулів та контролерів.

Детальний опис методів, використаних при реалізації бізнес-логіки та математичного забезпечення веб-застосування наведений у таблиці 4.1.

Таблиця 4.1 – Специфікація функцій

Назва функції	Детальний опис її роботи
branchesAndBoundes()	Функція реалізовує алгоритм пошуку складу найперспективнішої групи методом гілок та меж. На вхід приймає кількість кандидатів, масиви з вартістю їх навчання та їхньою перспективністю, а також – максимальний ресурс тренера. На вихід повертає час виконання, масив задіяних учнів та їхню сумарну перспективність
euristic()	Функція реалізовує алгоритм пошуку складу найперспективнішої групи евристичним методом. Вхідні і вихідні дані співпадають із функцією вище.
dynamicProgramming()	Функція реалізовує алгоритм пошуку складу найперспективнішої групи методом динамічного програмування. Вхідні і вихідні дані співпадають із функцією вище.
getUB()	Порахувати верхню межу (для методу гілок та меж)
sortWithIndeces()	Службова функція, яка сортує масив, запам'ятовуючи вихідні індекси та примішуючи їх до масиву-відповіді.
generateTestingInputs()	Метод для генерації тестових даних для випробувань алгоритмів.
useLocationBlocker()	Перешкоджає роутеру додавати у об'єкт history двічі
addImageToGallery()	Додає завантажену фотографію до галереї

Кінець таблиці 4.1

payWithStripe()	Поповнення рахунку для оплати заняття
max()	Знаходить найбільше з двох чисел
SmoothScrollTop()	Плавню прокрулює сторінку до гори при першому рендері
unixToDateString()	Трансформація часової мітки у читабельний для людини вигляд
fetchPosts()	Завантажити з сервера наявні пости
authorize()	Залогінення користувачем
recoverPassword()	Надсилає на сервер дані сесії для відновлення втраченого паролю
addImageToGallery()	Додає завантажену фотографію до галереї

Висновок до розділу

У даному розділі наведено опис архітектури, засобів розробки та основних методів програмного забезпечення. Було описано технології, на основі яких була побудована система, а також – причини їх використання. Розроблено діаграми, наведено опис функцій і сформульовано вимоги до технічного забезпечення. Було описано принцип за яким проект буде розгорнутий у глобальній мережі.

5 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

5.1 Керівництво користувача

Після завантаження веб-сайт користувач бачить головне меню домашньої сторінки застосування. Знизу висвітяться попап (на смартфоні) або в адресній строці у хромі з'явиться кнопка з пропозицією завантажити застосування на пристрій, у разі відсутності інтернет з'єднання, користувач побачить інформаційне повідомлення.

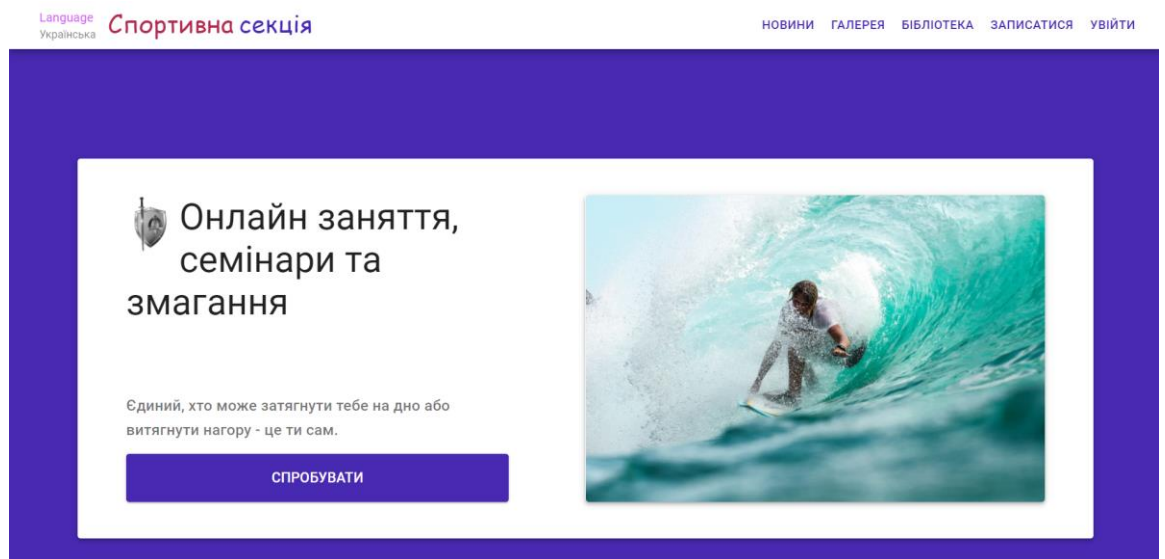


Рисунок 5.1 – Головне меню

Головне меню на мобільній версії можна відкрити через кнопку-бургер. На десктопі воно знаходиться у шапці сайту. З головного меню можна змінити мову інтерфейсу, а також потрапити на:

- блогу з новинами;

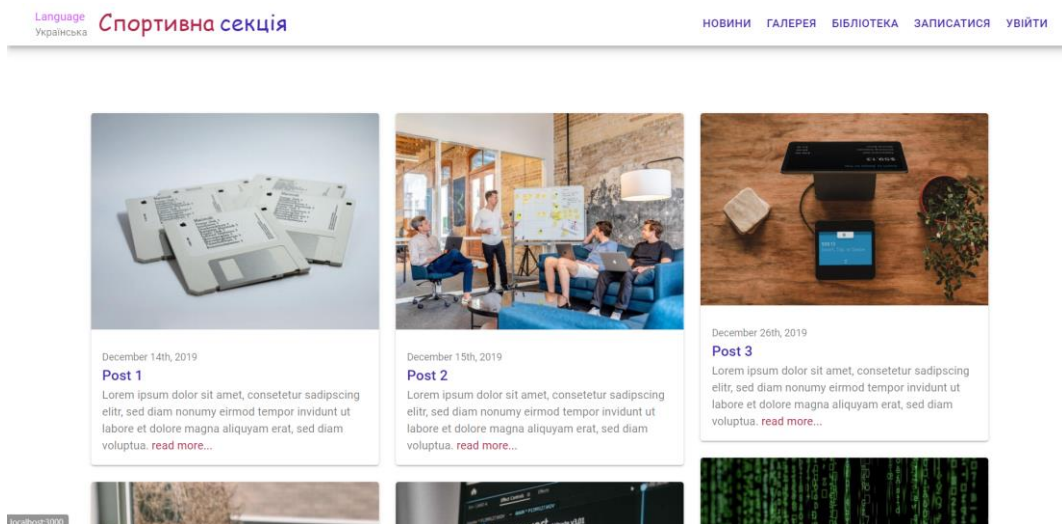


Рисунок 5.2 – Блог новин

- відео/фотогалереї;
- бібліотеки;
- розкладу занять (лише авторизованим користувачам);
- особистий кабінет (лише авторизованим користувачам);
- відкрити реєстрацію;

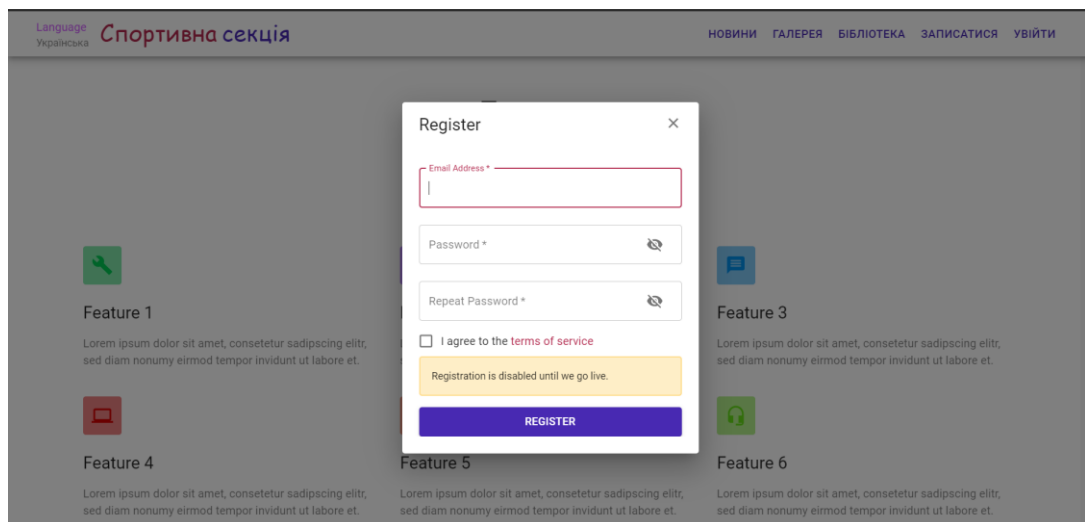


Рисунок 5.3 – Попап реєстрації

- відкрити попап логіну;

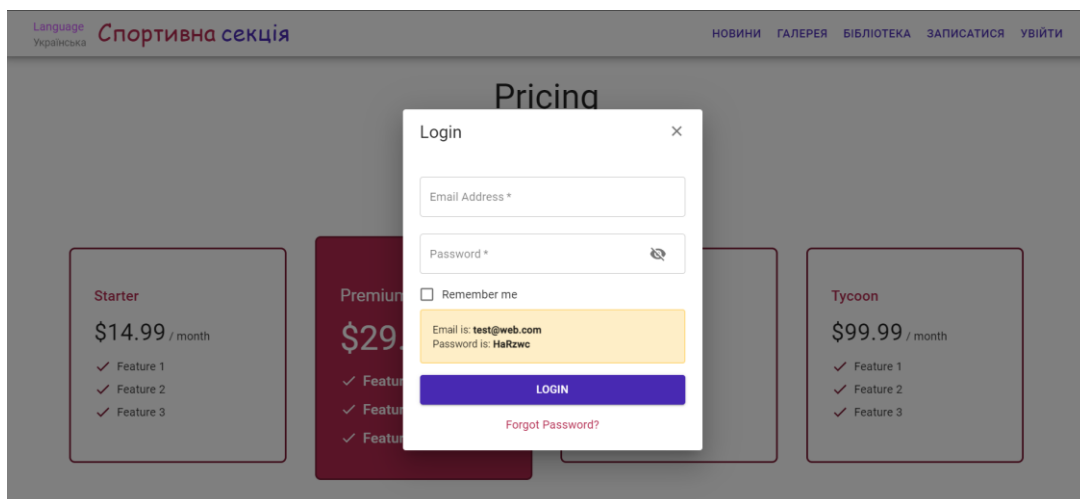


Рисунок 5.4 – Попап логіну

На сторінці блогу можна відкрити кожний пост для детального перегляду, звідки переходити на наступні, перемикаючи їх власноруч.



Рисунок 5.5 – Запис у блозі (розгорнутий)

Адміністратор має можливість редагувати записи у блозі за допомогою сторінки панелі інструментів. Звісно, тільки після проходження авторизації. Наявні інструменти дозволяють додавати зображення, редагувати його інструментом «кроп» та додавати опис до фото. Весь функціонал адмін панелі адаптований під будь-який розмір та орієнтацію екрану більшості пристроїв.

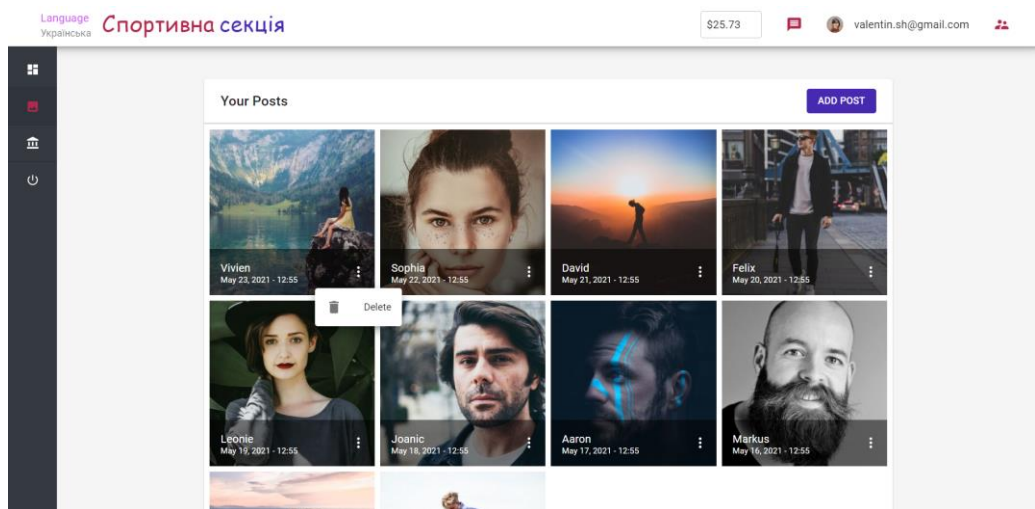


Рисунок 5.6 – Видалення фото із галереї

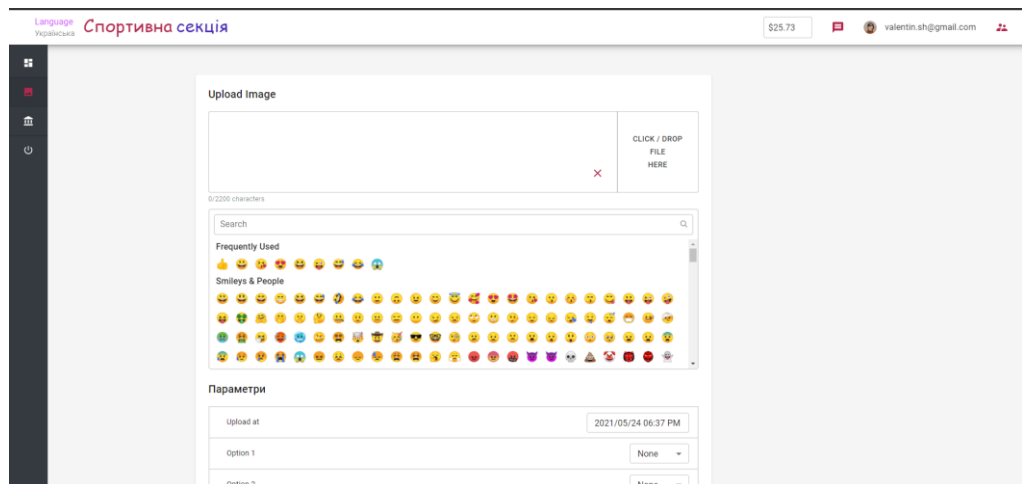


Рисунок 5.7 – Додання фото до галереї

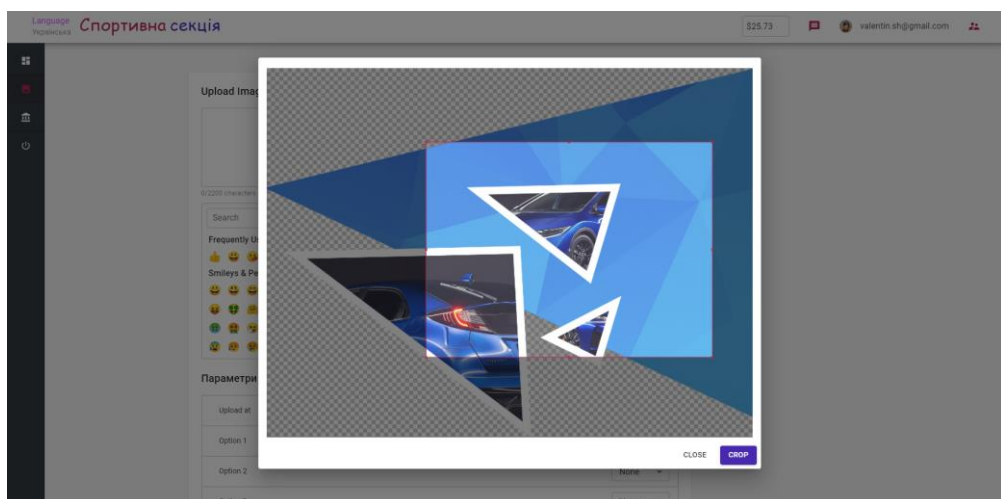


Рисунок 5.8 - «Кроп» меню зображення

Змн.	Арк.	№ докум.	Підпис	Дата

Language Українська **Спортивна секція** \$25.73 valentin.sh@gmail.com

Status Premium Account	ADD BALANCE		
Action	Balance change	Date	Paid until
Business subscription	-\$49.99	24.4.2021	24.5.2021
Premium subscription	-\$29.99	24.3.2021	24.4.2021
Starter subscription	-\$14.99	22.2.2021	24.3.2021
Added funds	+\$50.00	22.1.2021	22.2.2021
Premium subscription	-\$29.99	23.12.2020	22.1.2021
Business subscription	-\$49.99	22.11.2020	23.12.2020
Starter subscription	-\$14.99	23.10.2020	22.11.2020
Business subscription	-\$49.99	22.9.2020	23.10.2020
Added funds	+\$50.00	23.8.2020	22.9.2020
Added funds	+\$50.00	23.7.2020	23.8.2020

Рисунок 5.9 – Сторінка перевірки статусу оплати занять

Спортивна секція \$25.73 valentin.sh@gmail.com

Status Premium Account	ADD BALANCE		
Action	Balance change	Date	Paid until
Business subscription	-\$49.99	24.4.2021	24.5.2021
Premium subscription	-\$29.99	24.3.2021	24.4.2021
Starter subscription	-\$14.99	22.2.2021	24.3.2021
Added funds	+\$50.00	22.1.2021	22.2.2021
Premium subscription	-\$29.99	23.12.2020	22.1.2021
Business subscription	-\$49.99	22.11.2020	23.12.2020
Starter subscription	-\$14.99	23.10.2020	22.11.2020
Business subscription	-\$49.99	22.9.2020	23.10.2020
Added funds	+\$50.00	23.8.2020	22.9.2020

Add Balance

CREDIT CARD **SEPA DIRECT DEBIT**

Your Name * Amount *

Credit Card *

You can check this integration using the credit card number **4242 4242 4242 4242 04 / 24 24 242 42424**

PAY WITH STRIPE

Рисунок 5.10 – Попап оплати за допомогою Stripe

У разі переходу на неіснуючу сторінку (наприклад, при помилковому введенні URL сторінки вручну), користувач потрапляє на сторінку 404. З неї можна легко повернутися на головну натиснувши або на інформаційне повідомлення, або на логотип спортивної секції.

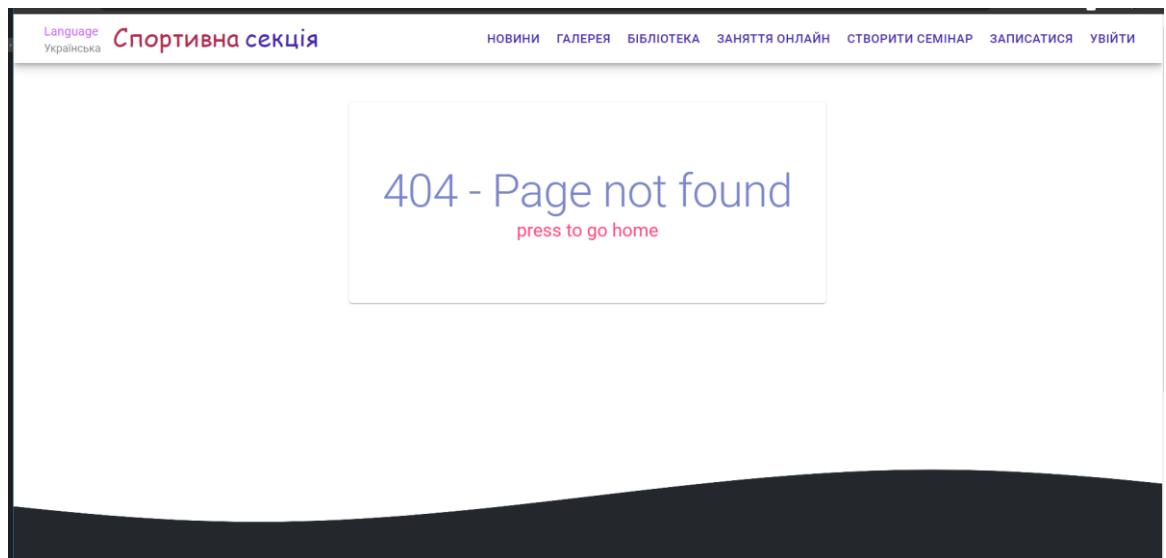


Рисунок 5.12 – Сторінка не знайдена

5.2 Випробування програмного продукту

Для перевірки роботи програмного забезпечення використовується широкий спектр засобів виявлення некоректної роботи програми та покращення рівня її відмовостійкості. У цьому веб-застосуванні було застосовано як автоматизоване тестування програмних модулів (фреймворк Jest) та вручну протестовані функціональні вимоги системи.

5.2.1 Мета випробувань

Метою випробувань являється перевірка відповідності функцій комплексу задач системи інформаційної підтримки діяльності спортивної секції вимогам технічного завдання.

Фреймворк Jest – засіб для написання unit-тестів, для перевірки коректності роботи програмних модулів при їх написанні та при їх подальшій модифікації.

5.2.2 Загальні положення

Випробування проводяться на основі наступних документів:

					ДП 7223.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		40

- ГОСТ 34.603–92. Інформаційна технологія. Види випробувань автоматизованих систем;
- ГОСТ РД 50-34.698-90. Автоматизовані системи вимог до змісту документів.

5.2.3 Результати випробувань

У результаті проведених випробувань протестовано наступні функції системи:

- реєстрація нового користувача;
- авторизація користувача;
- відновлення паролю;
- перегляд новин блогу;
- створення та публікація нових новин;
- робота форми оплати заняття та перегляд статусу оплати;
- додання та видалення фотографій із галереї;
- створення складу групи учнів для підготовки до змагань;
- робота онлайн уроку.

Під час кожного комміту та створення pull-реквесту у гілку master, відбувалася перевірка коректності коду за допомогою лінера та прогону юніт-тестів. Це дозволило виявити значну кількість проглядів та помилок, і суттєво підвищило рівень якості вихідного коду.

Тестування алгоритмів пошуку найперспективнішого складу учнів для змагань подано у третьому розділі.

Висновок до розділу

У даному розділі наведено керівництво користувача із прикладами зображень десктопної версії застосунку. Описаний спосіб та результати випробувань, згідно яким розроблений програмний продукт відповідає усім

					ДП 7223.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		41

необхідним вимогам. Скриншоти мобільної версії можна побачити у графічних матеріалах проекту.

					ДП 7223.00.000 ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗАГАЛЬНІ ВИСНОВКИ

В першому розділі обґрунтовано актуальність розробки, сформульовано мету і задачі розробки. Сформульовані функціональні вимоги.

Розроблено систему для організації діяльності спортивних секцій різних розмірів та направленості, що є гнучкою та масштабованою. Цю гнучкість отримано завдяки продуманій архітектурі на застосування новітніх технологій розробки веб-застосунків. Описано вхідні та вихідні дані та структурну схему бази даних.

Випробування системи показали її ефективність: як-то вимоги до SEO-оптимізації (SSR технологія), так і сучасний інтерфейс користувача, а також – розвинена функціональність системи (актуальність якої очевидна, напр. «онлайн заняття»). Використання технології GraphQL істотно зменшило навантаження на мережу, а застосований фреймворк React і бібліотека MaterialUI – час створення сучасного адаптивного кроссбраузерного інтерфейсу.

Розроблено алгоритмічне та програмне забезпечення, що дає тренерам змогу створювати команди для змагань, які матимуть найвищу перспективність та досягатимуть високих спортивних результатів. Наведено змістовну та математичну постановку задачі формування команди та методи її вирішення. Наведено результати та експериментальні дослідження ефективності застосованих методів.

Сформульовані вимоги до технічного забезпечення, описані засоби розробки. Розроблено керівництво користувача. Проведені випробування програмного забезпечення.

Розроблена система може використовуватись для підтримки процесів роботи спортивних секцій.

					ДП 7223.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		43

ПЕРЕЛІК ПОСИЛАНЬ

1. Peter Brucker “Scheduling algorithms” – Springer – 2006 – ст. 56-59.
2. Кононов Александр Вениаминович - Актуальные задачи теории расписаний: вычислительная сложность и приближенные алгоритмы – 2014 – ст. 63-65.
3. Лазарев Александр Алексеевич, Гафаров Евгений Рашидович “Теория расписаний задачи и алгоритмы” – Підручник з теорії розкладів. – с. 53-55.
4. Michael L. “Pinedo Scheduling Theory, Algorithms, and Systems” – Springer – 2009 – ст. 385-388.
5. React documentation [Електронний ресурс] — Режим доступу <https://reactjs.org/docs/getting-started.html>
6. Nextjs documentation [Електронний ресурс] — Режим доступу <https://nextjs.org/docs/getting-started>
7. Nestjs documentation [Електронний ресурс] — Режим доступу <https://docs.nestjs.com/>
8. Mobitfitness [Електронний ресурс] — Режим доступу <https://new.mobifitness.ru/>
9. Niko [Електронний ресурс] — Режим доступу <https://nikko.ua/>
10. IGymnasium [Електронний ресурс] — Режим доступу http://lgymnasium.kiev.ua/fiz_studiya.html
11. Основы динамического программирования. Достоинства и недостатки метода [Електронний ресурс] — Режим доступу http://refpin.ru/ref_yfsjgebewbew.html

Додаток А

Тексти програмного коду**Система інформаційної підтримки діяльності спортивної секції**

(Найменування програми (документа))

DVD-R

(Вид носія даних)

21 арк, 121 Кб

(Обсяг програми (документа) , арк., Кб)

Київ – 2021 року

					ДП 7223.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		45

_app.tsx

```

import React, { useEffect } from 'react'
import App from 'next/app'
import AOS from 'aos'
import 'aos/dist/aos.css'

import '../styles/globals.sass'
import { appWithTranslation } from 'i18n'

const MyApp = ({ Component, pageProps }: any) => {
  useEffect(() => {
    AOS.init({
      once: true,
    })
  }, [])

  return <Component {...pageProps} />
}

MyApp.getInitialProps = async (appContext: any) => ({
  ...(await App.getInitialProps(appContext)),
})

```

```
export default appWithTranslation(MyApp)
```

index.tsx

```

import React, { useCallback, useState } from 'react'
import { NextPage } from 'next'
import { MuiThemeProvider, CssBaseline } from '@material-ui/core'
import theme from '../common/theme'
import GlobalStyles from '../common/GlobalStyles'
import smoothScrollTop from 'common/functions/smoothScrollTop'
import { Nullable } from '@helpers/commonInterfaces/interfaces'
import Home from 'wrappers/LoggedOut/components/home/Home'

```

					ДП 7223.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		46

```

import LoggedOutWrapper from
'wrappers/LoggedOut/components/LoggedOutWrapper'

import { PageHead } from 'common/components/PageHead'

const IndexPage: NextPage = () => {
  const [selectedTab, setSelectedTab] =
useState<Nullable<string>>(null)

  const selectHome = useCallback(() => {
    smoothScrollTop()
    document.title = 'Workout | Kyiv'
    setSelectedTab('Home')
  }, [setSelectedTab])

  return (
    <>
      <PageHead />
      <MuiThemeProvider theme={theme}>
        <CssBaseline />
        <GlobalStyles />
        <LoggedOutWrapper {...{ selectedTab }} {...{ setSelectedTab }}>
          { /*@ts-ignore*/ }
          <Home {...{ selectHome }} />
        </LoggedOutWrapper>
      </MuiThemeProvider>
    </>
  )
}

export default IndexPage

```

[create-team.tsx](#)

```

import React, { useCallback, useState } from 'react'
import { NextPage } from 'next'
import { MuiThemeProvider, CssBaseline } from '@material-ui/core'
import theme from '../..common/theme'

```

					ДП 7223.00.000 ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

```
import GlobalStyles from '../common/GlobalStyles'

import LoggedInWrapper from
'wrappers/LoggedIn/components/LoggedInWrapper'

import smoothScrollTop from 'common/functions/smoothScrollTop'

import { PageHead } from 'common/components/PageHead'

import { Nullable } from '@helpers/commonInterfaces/interfaces'

import SeminarDashboard from
'wrappers/LoggedIn/components/dashboard/SeminarDashboard'

import { useTranslation } from 'i18n'

const IndexPage: NextPage = () => {
  const [selectedTab, setSelectedTab] =
useState<Nullable<string>>(null)

  const [isAddBalanceDialogOpen, setIsAddBalanceDialogOpen] =
useState(false)

  const [KnapsackChart, setKnapsackChart] = useState<any>(null)

  const [hasFetchedCardChart, setHasFetchedCardChart] = useState(false)

  const { t } = useTranslation(['common'])

  const selectCreateTeam = useCallback(() => {
    smoothScrollTop()

    document.title = 'Workout - Dashboard'

    setSelectedTab(t('Create team'))

    if (!hasFetchedCardChart) {
      setHasFetchedCardChart(true)

      import('../common/components/KnapsackChart').then((Component)
=> {
        setKnapsackChart(Component.default)
      })
    }

    }, [setSelectedTab, setKnapsackChart, hasFetchedCardChart,
setHasFetchedCardChart])

  return (
```

<PageHead />

<MuiThemeProvider theme={theme}>

<CssBaseline />

<GlobalStyles />

<LoggedInWrapper

{...{ selectedTab }}

{...{ setSelectedTab }}

{...{ isAddBalanceDialogOpen }}

{...{ setIsAddBalanceDialogOpen }}

>

<SeminarDashboard {...{ selectCreateTeam }} />

</LoggedInWrapper>

</MuiThemeProvider>

</>

)

}

export default IndexPage

SeminarDashboard.tsx

import { Container, Grid, Theme, Typography, withStyles } from
'@material-ui/core'

import KnapsackChart from 'common/components/KnapsackChart'

import { branchesAndBoundes } from 'common/functions/knapsack/bnb'

import { dynamicProgramming } from
'common/functions/knapsack/dynamicProgramming'

import { heuristic } from 'common/functions/knapsack/heuristic'

import { Input } from 'common/functions/knapsack/interfaces'

import { defaultTheme } from 'common/theme'

import { useTranslation } from 'i18n'

import React, { FC, useEffect, useState } from 'react'

const styles = (theme) => ({

```
root: {

  paddingLeft: 0,
  paddingRight: 0,
  '@media (max-width: 350px)': {
    paddingLeft: 0,
    paddingRight: 0,
  },
},
}))

type SeminarDashboardProps = {
  theme?: Theme
  classes: any
  selectCreateTeam: () => void
}

const SeminarDashboard: FC<SeminarDashboardProps> = ({ theme, classes,
selectCreateTeam }) => {

  const [styles, setStyles] = useState(defaultTheme)
  const { t } = useTranslation(['dashboard'])

  useEffect(selectCreateTeam, [selectCreateTeam])

  useEffect(() => {
    if (theme) {
      setStyles(theme)
    }
  }, [theme])

  type testingData = Array<{ pr: number; dp?: number; bnb?: number;
eu?: number }>
```

type delOpti = Array<{ pr: number; bnb: number; eu: number }>

Арк.

ДП 7223.00.000 ПЗ

50

```

const generateTestingInputs = () => {

  const testingData = []

  const genRandArr = (size: number) => {
    const res = []
    for (let i = 0; i < size; i += 1) {
      res.push(Math.floor(Math.random() * 10) + Math.abs(i -
Math.floor(Math.random() * 10)))
    }
    return res
  }

  for (let i = 8; i <= 36; i += 4) {
    testingData.push({
      n: i,
      cost: genRandArr(i),
      value: genRandArr(i),
      resource:
        Math.floor(Math.random() * 10) + 2 * Math.abs(i -
Math.floor(Math.random() * 10)) + i,
    })
  }

  return testingData
}

useEffect(() => {

  // const input: Input = { n: 4, cost: [3, 8, 4, 5], value: [3, 7,
2, 6], resource: 11 }

```

```

const inputs = generateTestingInputs()

console.log('inputs', inputs)

const dpOut: testingData = [],
      euOut: testingData = [],
      bnbOut: testingData = [],
      optOut: deltaOpti = []

inputs.forEach((input) => {
  const dp = dynamicProgramming(input)
  dpOut.push({ pr: input.n, dp: dp.time })

  const eu = euristic(input)
  euOut.push({ pr: input.n, eu: eu.time })

  const bnb = branchesAndBoundes(input)
  bnbOut.push({ pr: input.n, bnb: bnb.time })

  optOut.push({ pr: input.n, eu: eu.z, bnb: dp.z })
})

console.log('dp', dpOut)
console.log('eu', euOut)
console.log('bnb', bnbOut)

setDp(dpOut)
setEu(euOut)
setBnb(bnbOut)

setData(dpOut.map((item, idx) => ({ ...item, eu: euOut[idx].eu,
bnb: bnbOut[idx].bnb })))

setOpti(optOut)
}, [])

const [dp, setDp] = useState<testingData>([])
const [eu, setEu] = useState<testingData>([])

```

```

const [bnb, setBnb] = useState<testingData>([])

const [opti, setOpti] = useState<deltaOpti>([])


const [data, setData] = useState<testingData>([])


useEffect(() => {
  console.log('dat', data)
  console.log('opti', opti)
}, [data, opti])


const euLeg = [{ value: 'Euristics', id: 'time', type: 'line' }]
const dpLeg = [{ value: 'Dynamic Programming', id: 'time', type:
'line' }]
const bnbLeg = [{ value: 'Branches and Bounds', id: 'time', type:
'line' }]


return (
  <Container className={classes.root}>
    <Typography variant="subtitle1" gutterBottom>
      {t('Create seminar')}
    </Typography>
    <Grid container spacing={3}>
      <Grid item xs={12} md={12}>
        <KnapsackChart
          data={eu}
          legend={euLeg}
          color={styles.palette.primary.light}
          height="500px"
          title={t('Euristic method')}

```

			>	
Змн.	Арк.	№ докум.	Підпис	Дата

```
</Grid>

<Grid item xs={12} md={12}>

  <KnapsackChart

    data={bnb}

    legend={bnbLeg}

    color={styles.palette.secondary.light}

    height="500px"

    title={t('B&B method')}

  />

</Grid>

<Grid item xs={12} md={12}>

  <KnapsackChart

    data={dp}

    legend={dpLeg}

    color={styles.palette.primary.light}

    height="500px"

    title={t('DP method')}

  />

</Grid>

<Grid item xs={12} md={12}>

  <KnapsackChart

    data={data}

    legend={Array.prototype.concat(euLeg, bnbLeg, dpLeg)}

    additional={[[eu, dp]]}

    color={styles.palette.primary.light}

    height="500px"

    title={t('Methods comparison')}

  />
```

</Grid>

ДП 7223.00.000 ПЗ

Арк.

54

```
      <Grid item xs={12} md={12}>

        <KnapsackChart

          data={opti}

          legend={Array.prototype.concat(euLeg, bnbLeg, dpLeg)}

          yAxisName={t('Group prospects')}

          additional={[[opti.map((item) => ({ bnb: item.bnb, pr:
item.pr }))] ]}

          color={styles.palette.primary.light}

          height="500px"

          title={t('Euristic optimum delta')}

        />

      </Grid>

    </Grid>

  </Container>

)

}
```

```
export default withStyles(styles, { withTheme: true
})(SeminarDashboard)
```

[Knapsack.tsx](#)

```
import React, { useState, useCallbck, useEffect } from 'react'

import {

  AreaChart,

  Area,

  XAxis,

  Tooltip,

  ResponsiveContainer,

  YAxis,
```

LineChart,

```
Line,  
Legend,  
Label,  
} from 'recharts'  
  
import format from 'date-fns/format'  
  
import {  
  Card,  
  CardContent,  
  Typography,  
  IconButton,  
  Menu,  
  MenuItem,  
  withStyles,  
  Box,  
} from '@material-ui/core'  
  
import MoreVertIcon from '@material-ui/icons/MoreVert'  
  
import { useTranslation } from 'i18n'  
  
const styles = (theme) => ({  
  cardContentInner: {  
    marginTop: theme.spacing(-4),  
  },  
  xAxis: {  
    position: 'relative',  
    textAlign: 'right',  
  },  
})
```

// function labelFormatter(label) {

ДП 7223.00.000 ПЗ

Арк.

56

```
// return format(new Date(label * 1000), 'MMMM d, p yyyy')  
// }
```

```
function calculateMin(data, yKey, factor) {  
  let max = Number.POSITIVE_INFINITY  
  data.forEach((element) => {  
    if (max > element[yKey]) {  
      max = element[yKey]  
    }  
  })  
  return Math.round(max - max * factor)  
}
```

```
const itemHeight = 216
```

```
const options = ['1 Week', '1 Month', '6 Months']
```

```
interface KnapsackChartProps {  
  color: string  
  data: any[]  
  additional?: any[]  
  yAxisName?: string  
  title: string  
  classes: any  
  theme: any  
  height: string  
  legend: any  
}
```

```
const KnapsackChart: React.FC<KnapsackChartProps> = (props) => {
```

Арк.

ДП 7223.00.000 ПЗ

57

```
const { color, data, title, classes, theme, height, legend,
additional, yAxisName } = props

const { t } = useTranslation(['dashboard'])

const [anchorEl, setAnchorEl] = useState(null)

const [selectedOption, setSelectedOption] = useState('1 Month')

const handleClick = useCallback(
  (event) => {
    setAnchorEl(event.currentTarget)
  },
  [setAnchorEl]
)

const formatter = useCallback(
  (value) => {
    return [value, title]
  },
  [title]
)

const getSubtitle = useCallback(() => {
  switch (selectedOption) {
    case '1 Week':
      return 'Last week'
    case '1 Month':
      return yAxisName ?? 'Time (ms)'
    case '6 Months':
      return 'Last 6 months'
    default:
```

```

        throw new Error('No branch selected in switch-
statement')
    }
}, [selectedOption])

// const processData = useCallback(() => {
//     let seconds
//     switch (selectedOption) {
//         case '1 Week':
//             seconds = 60 * 60 * 24 * 7
//             break
//         case '1 Month':
//             seconds = 60 * 60 * 24 * 31
//             break
//         case '6 Months':
//             seconds = 60 * 60 * 24 * 31 * 6
//             break
//         default:
//             throw new Error('No branch selected in switch-statement')
//     }
//     const minSeconds = new Date() / 1000 - seconds
//     const arr = []
//     for (let i = 0; i < data.length; i += 1) {
//         if (minSeconds < data[i].timestamp) {
//             arr.unshift(data[i])
//         }
//     }
//     return arr
// }, [data, selectedOption])

```

```
setAnchorEl(null)
}, [setAnchorEl])

const selectOption = useCallback(
  (selectedOption) => {
    setSelectedOption(selectedOption)
    handleClose()
  },
  [setSelectedOption, handleClose]
)

const isOpen = Boolean(anchorEl)

const getTime = (): 'dp' | 'bnb' | 'eu' => {
  if (data[0]?.dp) return 'dp'
  if (data[0]?.bnb) return 'bnb'
  if (data[0]?.eu) return 'eu'
  return 'dp'
}

const [state, setState] = useState({
  opacity: {
    dp: 1,
    eu: 1,
    bnb: 1,
  },
})

useEffect(() => {
```

if (data.keys.length > 2) {

```
      setState({

        opacity: {
          dp: 1,
          eu: 1,
          bnb: 1,
        },
      })
    }
  }, [])

const handleMouseEnter = (o) => {
  const { dataKey } = o
  const { opacity } = state

  setState({
    opacity: { ...opacity, [dataKey]: 0.5 },
  })
}

const handleMouseLeave = (o) => {
  const { dataKey } = o
  const { opacity } = state

  setState({
    opacity: { ...opacity, [dataKey]: 1 },
  })
}
```

		return (		
Змн.	Арк.	№ докум.	Підпис	Дата

ДП 7223.00.000 ПЗ

Арк.

61

```
<Card>

  <Box pt={2} px={2} pb={4}>

    <Box display="flex" justifyContent="space-between">

      <div>

        <Typography variant="subtitle1">{title}</Typography>

        <Typography variant="body2" color="textSecondary">

          {t(getSubtitle())}

        </Typography>

      </div>

      <div>

        <IconButton

          aria-label="More"

          aria-owns={isOpen ? 'long-menu' : undefined}

          aria-haspopup="true"

          onClick={handleClick}

        >

          <MoreVertIcon />

        </IconButton>

        <Menu

          id="long-menu"

          anchorEl={anchorEl}

          open={isOpen}

          onClose={handleClose}

          PaperProps={{

            style: {

              maxHeight: itemHeight,

              width: 200,

            },

          }}

        </Menu>

      </div>

    </Box>

  </Card>
```

```
    }}
    disableScrollLock
  >
  {options.map((option) => (
    <MenuItem
      key={option}
      selected={option === selectedOption}
      onClick={() => {
        selectOption(option)
      }}
      name={option}
    >
      {option}
    </MenuItem>
  ))}
</Menu>
</div>
</Box>
</Box>
<CardContent>
  <Box className={classes.cardContentInner} height={height}>
    <ResponsiveContainer width="100%" height="100%">
      <LineChart data={data}>
        <Line
          type="monotone"
          dataKey={getTime()}
          strokeOpacity={state.opacity[getTime()]}
          stroke="#8884d8"
```

			/>	
Змн.	Арк.	№ докум.	Підпис	Дата

```
<Label value="Pages of my website" offset={0}
position="insideBottom" />

{additional &&
  additional.map((_, idx) => (
    <Line
      key={idx}
      type="monotone"
      dataKey={idx === 0 ? 'eu' : 'bnb'}
      strokeOpacity={state.opacity[idx === 0 ? 'eu' :
'bnb']]
      stroke={`#${idx === 0 ? 'f' : '2'}422${idx}a`}
    />
  ))}
<XAxis dataKey="pr" type="number" />
<YAxis dataKey={getTime()} />
<Legend onMouseEnter={handleMouseEnter}
onMouseLeave={handleMouseLeave} />
<Tooltip
  /* formatter={formatter} */
  cursor={false}
  contentStyle={{
    border: 'none',
    padding: theme.spacing(1),
    borderRadius: theme.shape.borderRadius,
    boxShadow: theme.shadows[1],
  }}
  labelStyle={theme.typography.body1}
  itemStyle={{
    fontSize: theme.typography.body1.fontSize,
    letterSpacing: theme.typography.body1.letterSpacing,
    fontFamily: theme.typography.body1.fontFamily,
```

```

                                lineHeight:
theme.typography.body1.lineHeight,

                                fontWeight: theme.typography.body1.fontWeight,

                                }}

                                />

                                </LineChart>

                                </ResponsiveContainer>

                                <Typography variant="body2" className={classes.xAxis}
color="textSecondary">

                                {t('Number of students')}

                                </Typography>

                                </Box>

                                </CardContent>

                                </Card>

                                )

                                }

                                export default withStyles(styles, { withTheme: true })(KnapsackChart)
```

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО”
Кафедра автоматизованих систем обробки інформації та управління

УЗГОДЖЕНО

Керівник проєкту

Оксана ЖУРАКОВСЬКА

(підпис)

(вл. ім'я, прізвище)

“5” квітня 2021 р.

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

Олександр ПАВЛОВ

(підпис)

(вл. ім'я, прізвище)

“6” квітня 2021 р.

Система інформаційної підтримки діяльності спортивної секції

ТЕХНІЧНЕ ЗАВДАННЯ

Шифр *ДП 7223.01.000 ТЗ*

на 11 сторінках

Київ – 2021 року

ЗМІСТ

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ.....	1
1.1 Повне найменування системи та її умовне позначення.....	2
1.2 Найменування організації-замовника та організацій-учасників робіт.....	3
1.3 Перелік документів, на підставі яких створюється система	3
1.4 Планові терміни початку і закінчення роботи зі створення системи.....	3
2 ПРИЗНАЧЕННЯ І ЦІЛІ СТВОРЕННЯ СИСТЕМИ	4
2.1 Призначення системи.....	4
2.2 Цілі створення системи.....	4
2.2 Задачі при реалізації системи.....	4
3 ХАРАКТЕРИСТИКА ОБ'ЄКТА АВТОМАТИЗАЦІЇ.....	5
4 ВИМОГИ ДО СИСТЕМИ.....	6
4.1 Вимоги до системи в цілому.....	6
4.2 Вимоги до функціональних характеристик.....	6
4.3 Вимоги до видів забезпечення	6
5 СТАДІЇ ТА ЕТАПИ РОЗРОБКИ.....	7
6 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ.....	8

					ДП 7223.01.000 ТЗ			
Зм.	Арк.	Прізвище	Підпис	Дата				
Розроб.		Шаповал В. В.			Система інформаційної підтримки діяльності спортивної секції	Літ.	Лист	Листів
Перевірив.		Жураковська О. С.					2	11
						КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС72		
Н. кон.		Новінський В. П.						
Затв.		Павлов О.А.						

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Повне найменування системи та її умовне позначення

Повна назва системи: Система інформаційної підтримки діяльності спортивної секції.

Умовне позначення: Sport school.

1.2 Найменування організації-замовника та організацій-учасників робіт

Замовником проекту є кафедра Автоматизованих систем обробки інформації та управління НТУУ «КПІ ім. Ігоря Сікорського». Представником замовника є Жураковська Оксана Сергіївна.

Розробником системи є студент факультету інформатики та обчислювальної техніки НТУУ «КПІ ім. Ігоря Сікорського»: студент групи ІС72 Шаповал Валентин Вадимович.

1.3 Перелік документів, на підставі яких створюється система

При розробці системи і створенні проектно-експлуатаційної документації Виконавець повинен керуватися вимогами наступних нормативних документів:

- ДСТУ 19.201-78. Технічне завдання. Вимоги до змісту і оформлення;
- ДСТУ 34.601-90. Комплекс стандартів на автоматизовані системи. Автоматизовані системи. Стадії створення;
- ДСТУ 34.201-89. Інформаційні технології. Комплекс стандартів на автоматизовані системи. Види, комплексність і позначення документів при створенні автоматизованих систем.

1.4 Планові терміни початку і закінчення роботи зі створення системи

Плановий термін початку роботи над створенням системи – 31 безерня 2021 року. Плановий термін по закінченню роботи над створенням системи

					ДП 7223.01.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

ДП 7223.01.000 ПЗ

– не пізніше 4 червня 2021 року.

					ДП 7223.01.000 ПЗ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

2 ПРИЗНАЧЕННЯ І МЕТА СТВОРЕННЯ СИСТЕМИ

2.1 Призначення системи

Призначенням розробки є підтримка діяльності спортивної секції

2.2 Цілі створення системи

Метою розробки є підвищення ефективності роботи секції за рахунок її раціональної організації та забезпечення зручної взаємодії користувачів та адміністрації, враховуючи режим пандемії.

Цілями розробки системи виступають:

- створення порталу для перегляду новин та навчально-методичного матеріалу секції;
- організація розкладу занять із ціллю максимізації перспективності групи;
- забезпечення функціоналу для проведення онлайн занять;
- функціональність онлайн оплати занять;
- перегляд та редагування системи досягнень учнів.

2.3 Задачі при реалізації системи

Наступні задачі постають для досягнення вказаних вище цілей:

- створення унікального UI/UX дизайну;
- верстання Front-end частини сайту та програмування базової бізнес логіки;
- написання серверної частини застосування, налаштування бази даних;
- заповнити контент сайту базовим достатнім датасетом навчально-методичного матеріалу (відео, галерея, бібліотека, блог новин);
- реалізувати функціонал для оплати занять онлайн;
- реалізувати функціонал для проведення занять онлайн;
- реалізувати алгоритм вирішення «задачі з рюкзаком»;

					ДП 7223.01.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		5

- проаналізувати створений датасет за розробленими алгоритмами, за проведенням аналізом визначити лінгвістичні параметри досліджуваних перекладачів;

- аналіз та тлумачення висновків по технологічним та алгоритмічним рішенням, використаним при реалізації та тестуванні програмного продукту.

					ДП 7223.01.000 ПЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

3 ХАРАКТЕРИСТИКА ОБ'ЄКТА АВТОМАТИЗАЦІЇ

Розгорнути систему можна на локальному сервері або на хостинговому сервері у глобальній мережі. Взаємодія із системою передбачена на будь-якому пристрої із браузером. Передбачений також мінімальний функціонал при роботі у режимі оффлайн.

Результатом роботи алгоритму для організації розкладу є створений список учнів, здатний досягти найвищих спортивних показників.

Об'єктом автоматизації є організація занять спортивної секції.

					ДП 7223.01.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Вимоги до функціональних характеристик

Система має виконувати наступні функції:

- перегляд навчально-методичного матеріалу;
- перегляд розкладу занять;
- оплата заняття онлайн;
- проведення онлайн занять;
- перегляд та редагування системи досягнень учнів;
- забезпечити формування розкладу, створення шоу, створення груп з найвищим коефіцієнтом перспективності.

4.2 Вимоги до надійності

У разі відсутності інтернет з'єднання веб застосунок автоматично переходить у оффлайн режим, дані при цьому підтягуються з кешу. Також користувача повідомляють про проблеми з інтернет-з'єднанням, усі нестандартні сценарії роботи супроводжуються інформаційними повідомленнями. Для зв'язку з користувачем передбачена форма зворотного зв'язку.

Для даних користувачів має бути забезпечена повна конфіденційність.

4.3 Вимоги до складу і параметрів технічних засобів

Програмний продукт має бути представлений у вигляді клієнт-серверного веб-застосування, із клієнтською частиною із використанням SSR (server-side rendering) технології для забезпечення оптимізації парсингу сайті пошуковими роботами та PWA (Progressive Web Apps) для кращого користувацького досвіду при роботі зі застосунком.

Для коректної роботи серверної частини веб застосування необхідно використовувати веб-сервер, що може бути представлений локальною

					ДП 7223.01.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

обчислювальною машиною. Мінімальні системні вимоги для роботи серверу розгортання застосунку:

- процесор з тактовою частотою не нижче 2.2 ГГц;
- об'єм оперативної пам'яті (не менше 4 ГБ);
- жорсткий диск (не менше 32 ГБ);
- операційна система Linux Ubuntu 16.04 і вище;
- Node.js v 8+ npm v 6+.

Для запуску та коректної роботи клієнтської частини необхідні браузер та інтернет з'єднання.

					ДП 7223.01.000 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

5 СТАДІЇ І ЕТАПИ РОЗРОБКИ

5.1 Основні етапи виконання робіт з розробки системи ведення наукової роботи

№ п/п	Назва етапу роботи	Термін виконання етапу	Результат виконання
1	Підготовка технічного завдання на розробку програмного продукту	12.02.2021	
2	Створення архітектури системи	17.02.2020	
3	Технічне проектування – функціональність, модулі, задачі, цілі тощо	21.02.2020	
4	Узгодження з керівником інтерфейсу користувача	07.03.2020	
5	Розробка програми	14.04.2020	
6	Налагодження програми	22.04.2020	
7	Тестування програми	16.05.2020	
8	Здача готового програмного продукту замовнику	27.05.2020	

6 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ СИСТЕМИ

До системи, що побудована, мають бути додані:

- технічне завдання
- пояснювальна записка системи
- графічний матеріал креслень архітектури системи
- керівництво користувача

Методи рішення, зокрема, математичної частини мають бути порівняні та їх робота перевірена на реальних та синтетичних тестових даних.

					ДП 7223.01.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

Ім'я користувача:
Попенко Володимир Дмитрович

Дата перевірки:
31.05.2021 02:34:01 EEST

Дата звіту:
31.05.2021 23:15:28 EEST

ID перевірки:
1008093266

Тип перевірки:
Doc vs Internet + Library

ID користувача:
77149

Назва документа: Shapoval_bachelor_is72_v2

Кількість сторінок: 42 Кількість слів: 5564

Кількість символів: 47114 Розмір файлу: 5.47 MB ID файлу: 1008178200

11.3% Схожість

Найбільша схожість: 2.44% з джерелом з Бібліотеки (ID файлу: 1008167650)

6.38% Джерела з Інтернету

8

Сторінка 44

8.43% Джерела з Бібліотеки

30

Сторінка 45

0% Цитат

Не знайдено жодних цитат

Вилучення списку бібліографічних посилань вимкнене

0% Вилучень

Немає вилучених джерел

Модифікації

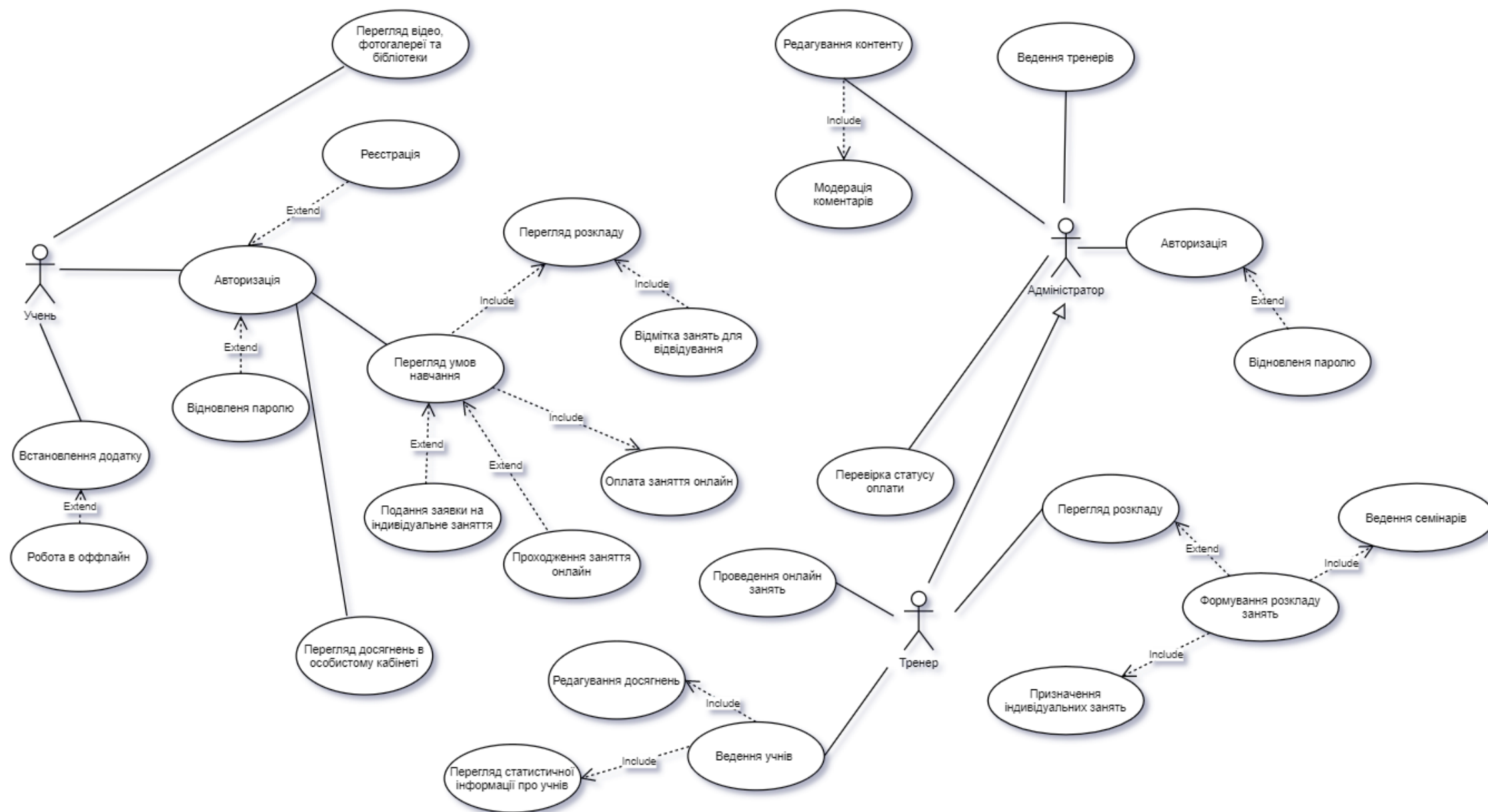
Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

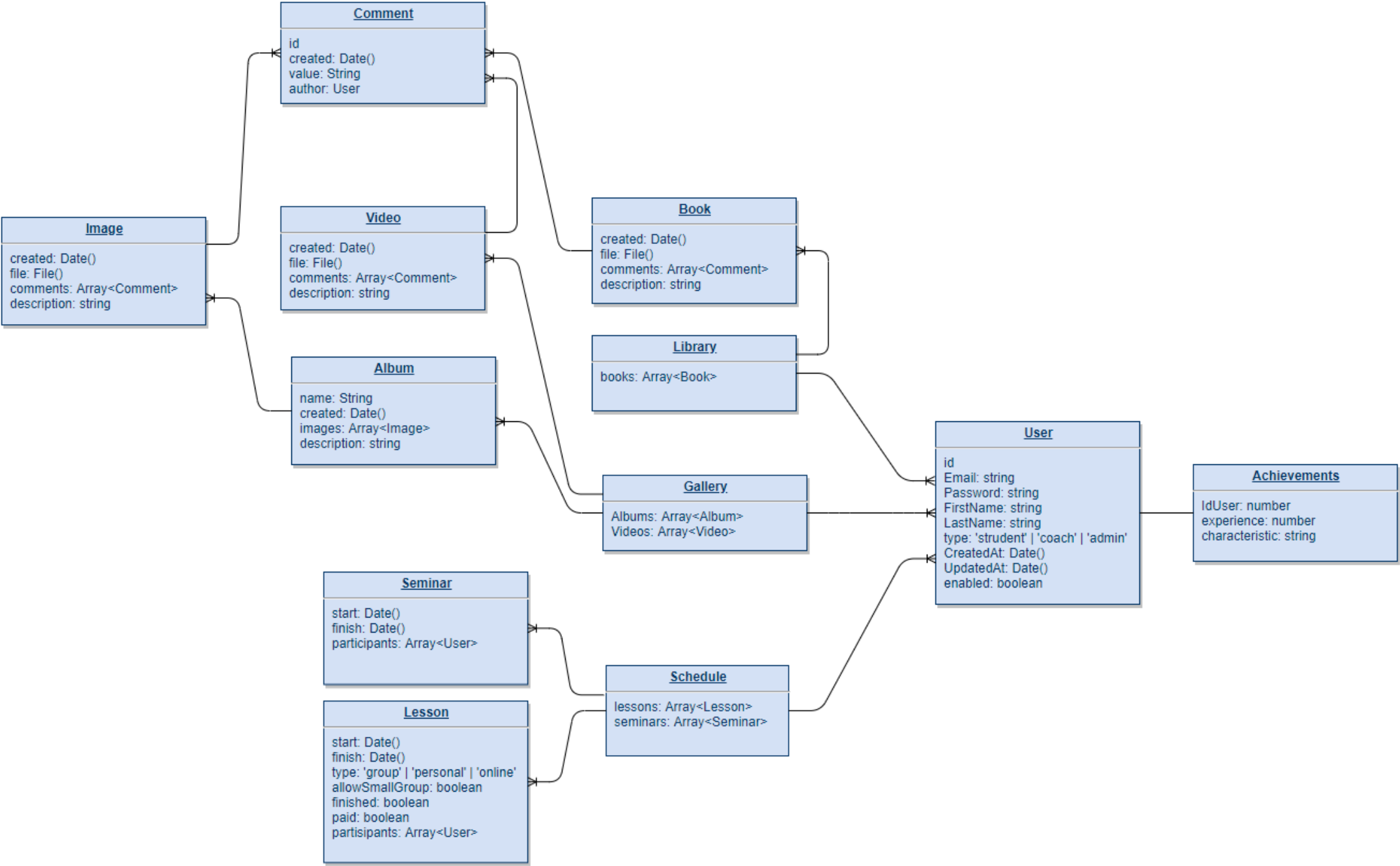
15

Графічний матеріал до дипломного проєкту

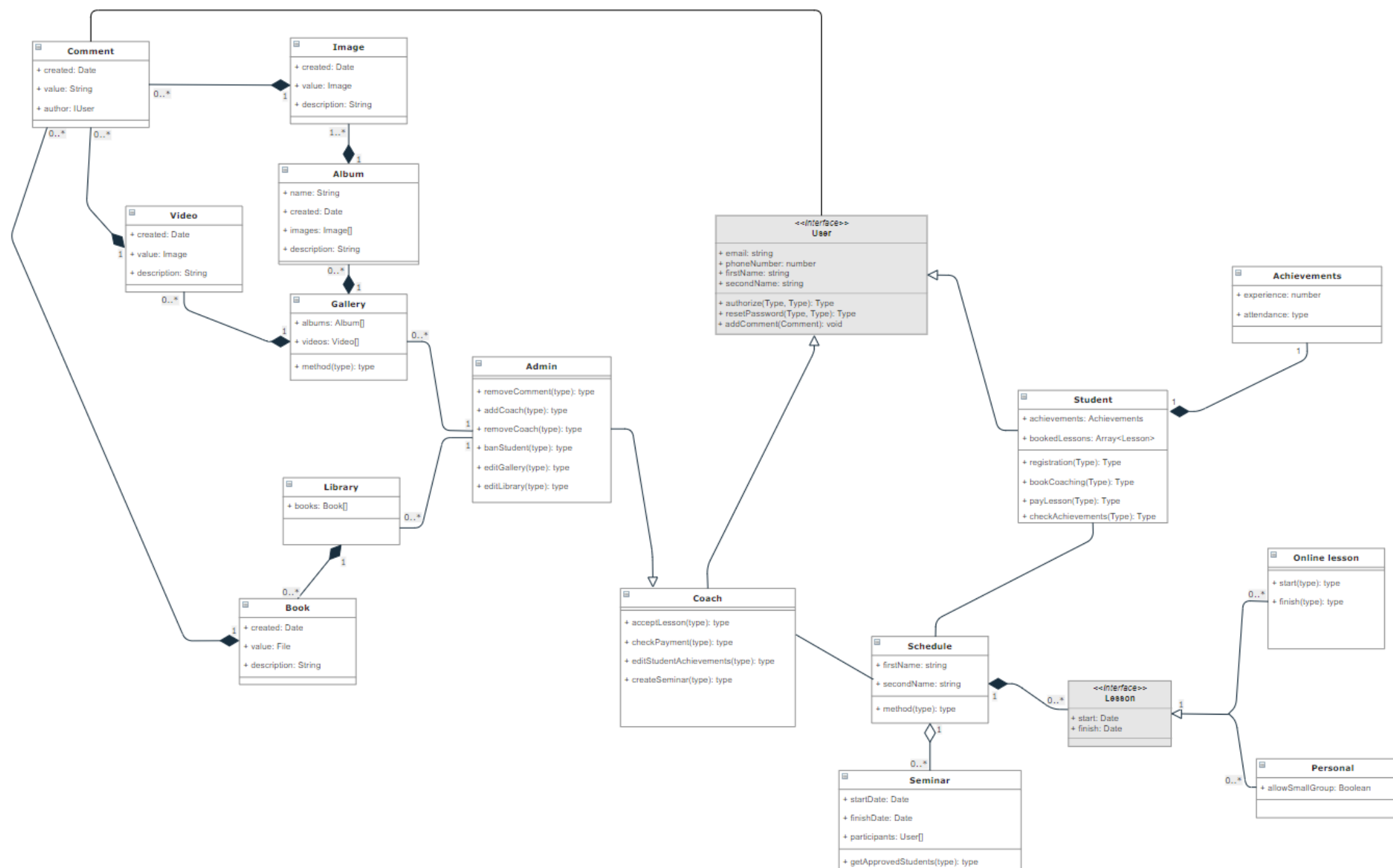
на тему: Система інформаційної підтримки діяльності спортивної
секції



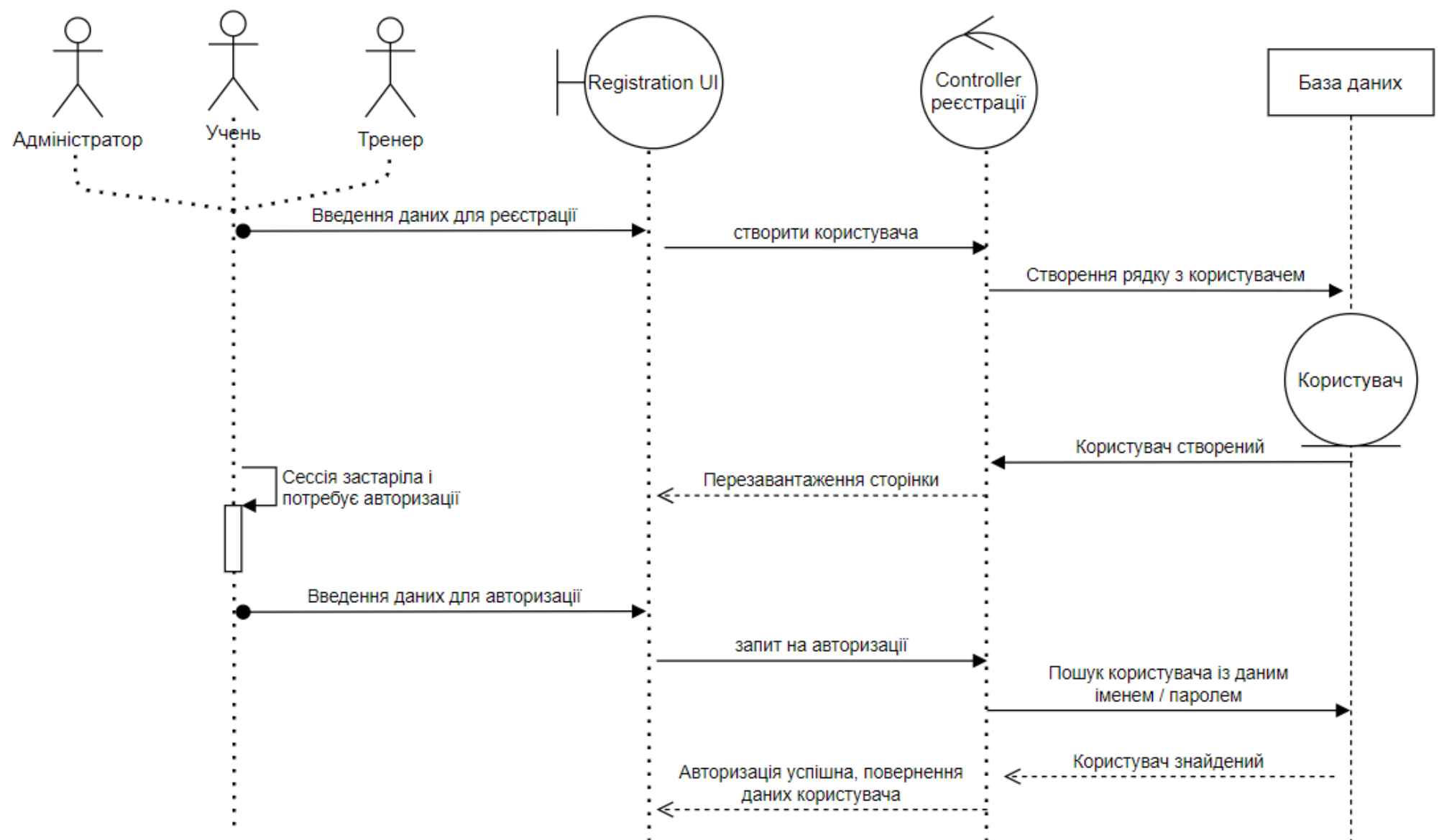
						ДП 7223.02.000 ССВ			
						Схема структурна варіантів використань	Літера	Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата					
Розробив		Шаповал В.В.							
Перевірив		Жураковська О.С.							
Консульт.									
Н. кон.		Новінський В.П.							
Затвердив		Жураковська О.С.							
						Система інформаційної підтримки діяльності спортивної секції			
						КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-72			



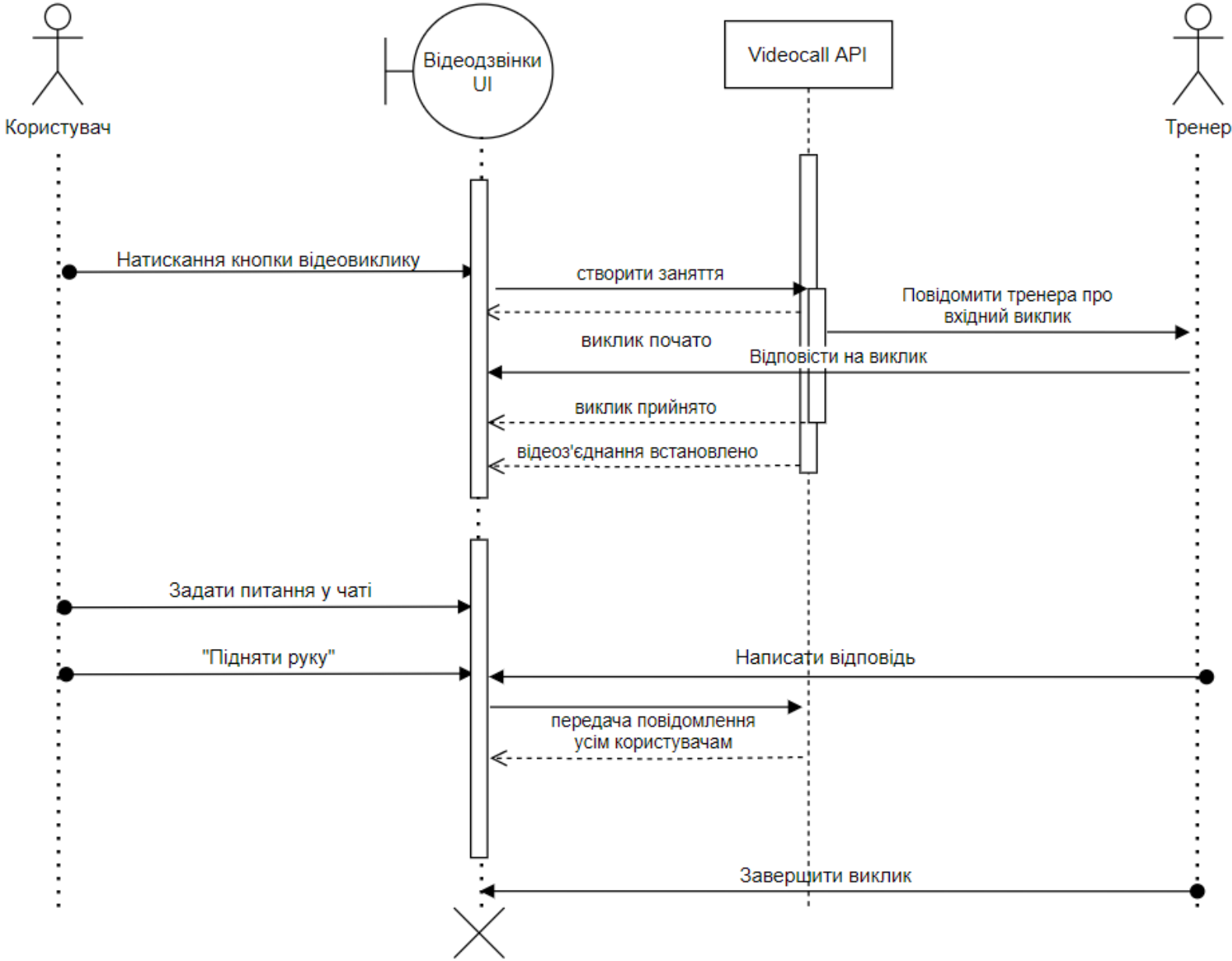
					ДП 7223.03.000 СБД				
					Схема структурна бази даних				
Зм	Арк.	№ документа	Підпис	Дата	Система інформаційної підтримки діяльності спортивної секції				
Розробив		Шаповал В.В.							
Перевішив		Жураковська О.С.							
Консульт.					КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-72				
Н. кон.		Новінський В.П.							
Затвердив		Жураковська О.С.							
					Літера				
					Маса				
					Масштаб				
					Аркуш 1				
					Аркушів 1				



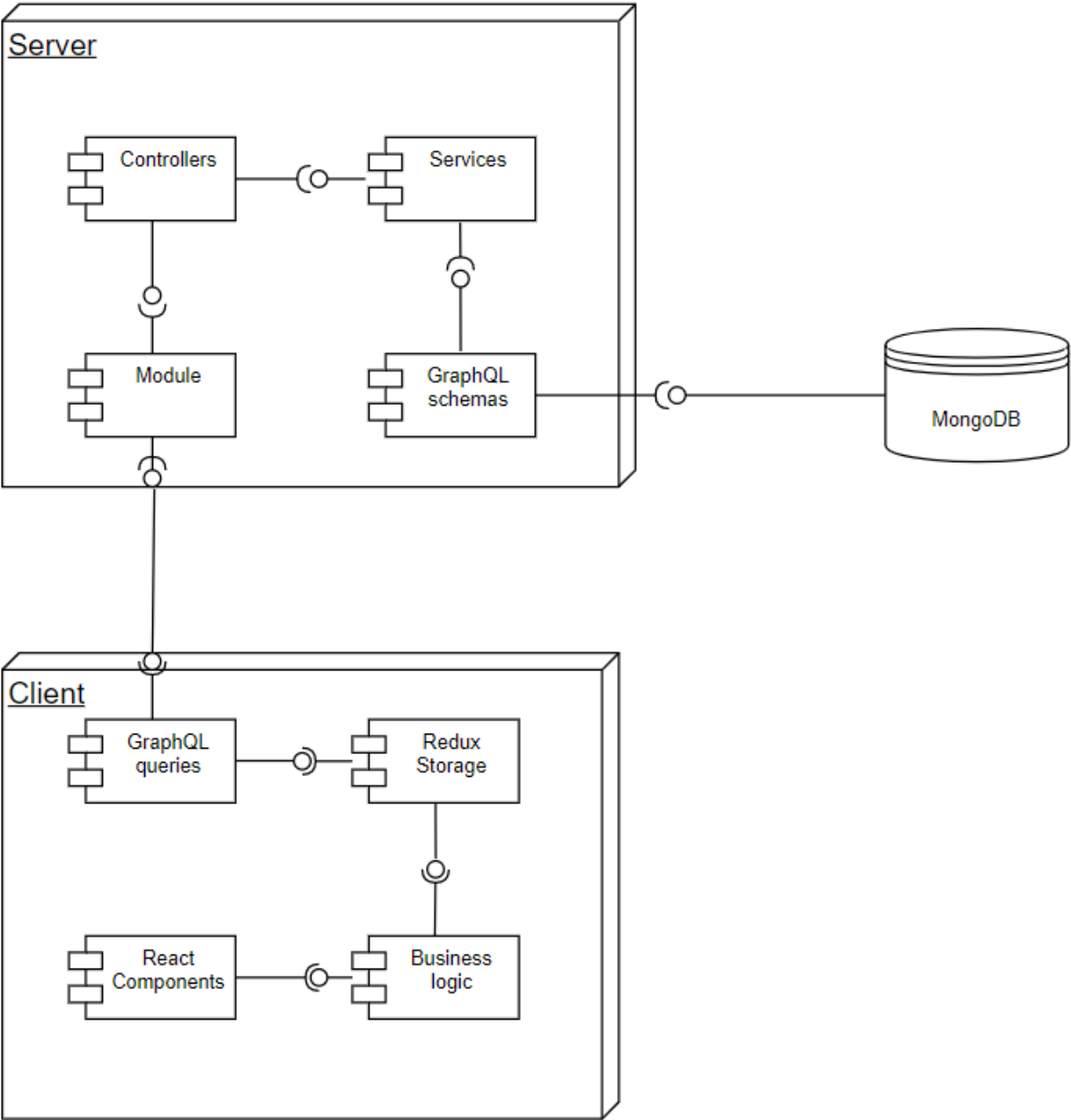
					ДП 7223.04.000 ССК			
					Схема структурна класів програмного забезпечення	Літера	Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата				
Розробив		Шаповал В.В.						
Перевірив		Жураковська О.С.			Система інформаційної підтримки діяльності спортивної секції	Аркуш 1		Аркушів 1
Консульт.						КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-72		
Н. кон.		Новінський В.П.						
Затвердив		Жураковська О.С.						



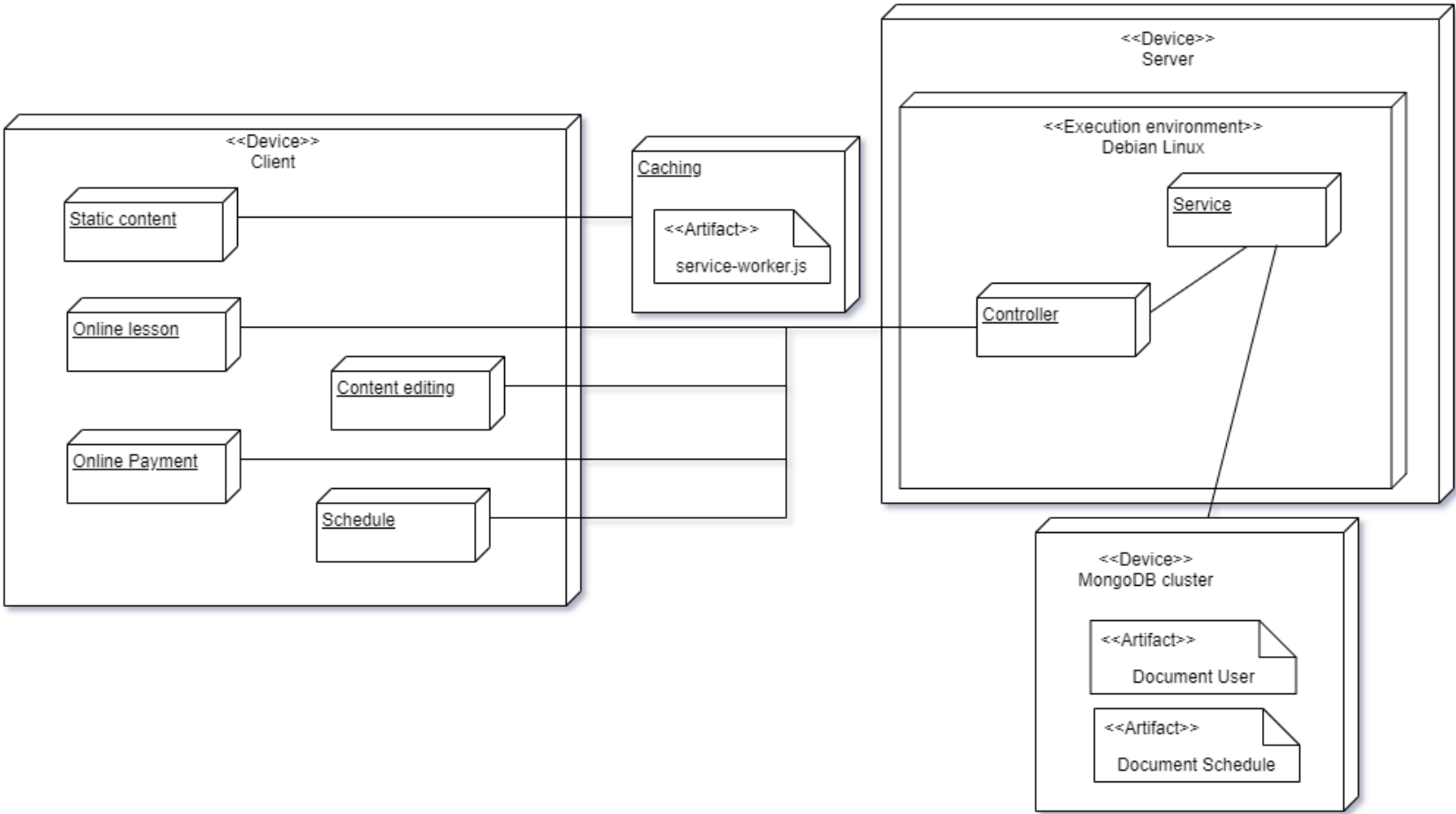
					ДП 7223.05.000 ССП			
					Схема структурна послідовності	Літера	Маса	Масштаб
Зм	Арк.	№ документа	Підпис	Дата				
Розробив		Шаповал В.В.				Аркуш 1		
Перевірив		Жураковська О.С.				Аркушів 2		
Консульт.					Система інформаційної підтримки діяльності спортивної секції	КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-72		
Н. кон.		Новінський В.П.						
Затвердив		Жураковська О.С.						



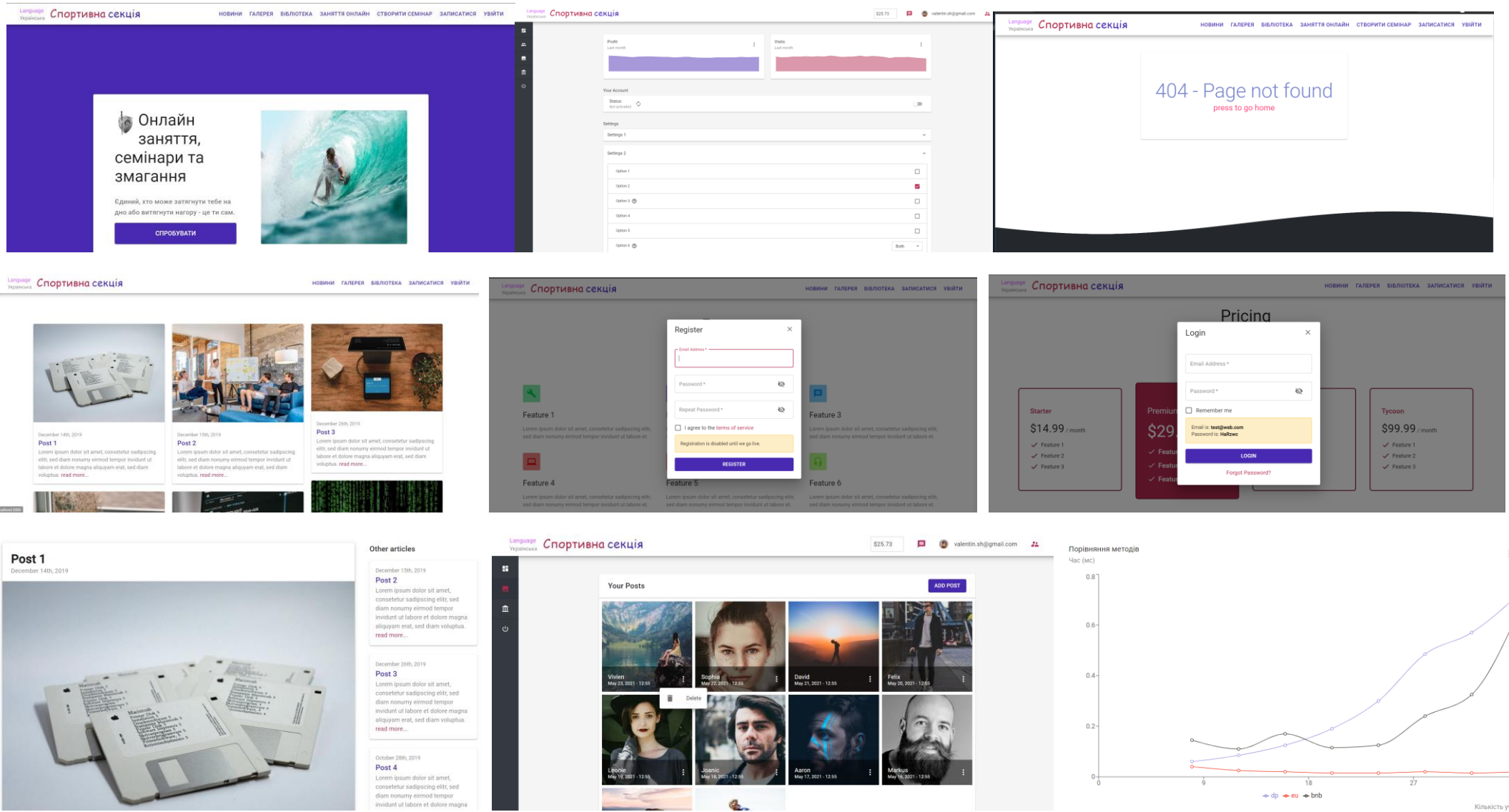
					ДП 7223.05.000 ССП			
					Схема структурна послідовності	Літера	Маса	Масштаб
Зм	Арк.	№ документа	Підпис	Дата				
Розробив		Шаповал В.В.						
Перевірів		Жураковська О.С.				Аркуш 2		Аркушів 2
Консульт.								
Н. кон.		Новінський В.П.			Система інформаційної підтримки діяльності спортивної секції	КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-72		
Затвердив		Жураковська О.С.						



					ДП 7223.06.000 СК			
					Схема структурна компонентів програмного забезпечення	Літера	Маса	Масштаб
Зм	Арк.	№ документа	Підпис	Дата				
Розробив		Шаповал В.В.						
Перевірів		Жураковська О.С.				Аркуш 1		Аркушів 1
Консульт.					Система інформаційної підтримки діяльності спортивної секції	КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-72		
Н. кон.		Новінський В.П.						
Затвердив		Жураковська О.С.						

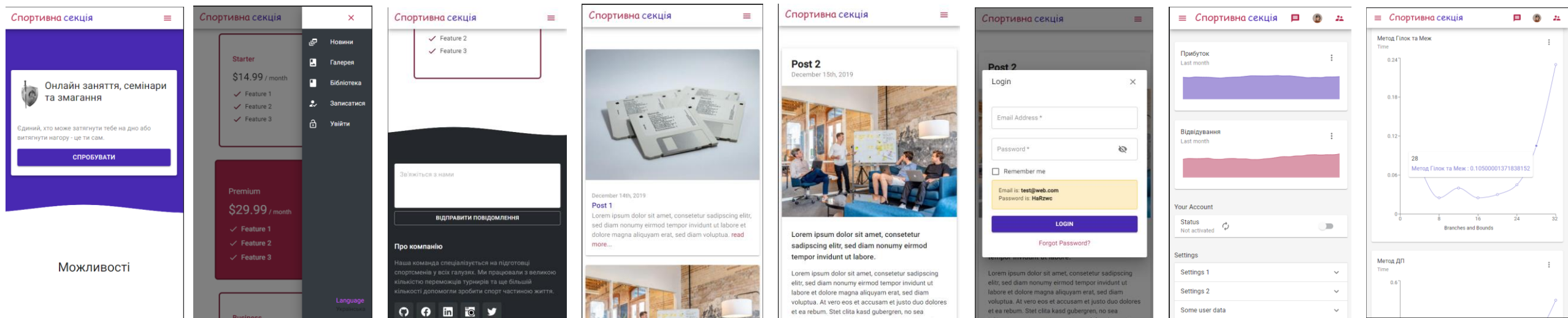


					ДП 7223.07.000 ССР			
					Схема структурна розгортання програмного забезпечення	Літера	Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата				
Розробив		Шаповал В.В.						
Перевірив		Жураковська О. С.				Аркуш 1	Аркушів 1	
Консульт.					Система інформаційної підтримки діяльності спортивної секції	КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-72		
Н. кон.		Новінський В.П.						
Затвердив		Жураковська О. С.						



					ДП 7223.08.000 ЕК						
					Креслення вигляду екранних форм	Літера		Маса		Масштаб	
Зм.	Арк.	№ документа	Підпис	Дата							
Розробив	Шаповал В.В.										
Перевірів	Жураковська О.С.					Аркуш 1		Аркушів 2			
Консульт.					Система інформаційної підтримки діяльності спортивної секції	КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-72					
Н. кон.	Новінський В.П.										
Затвердив	Жураковська О.С.										

ДП 7223.08.000 ЕК



					ДП 7223.08.000 ЕК									
					Креслення вигляду екранних форм	Літера			Маса			Масштаб		
Зм.	Арк.	№ документа	Підпис	Дата										
Розробив	Шаповал В.В.													
Перевірив	Жураковська О.С.					Аркуш 1			Аркушів 2					
Консульт.						Система інформаційної підтримки діяльності спортивної секції КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-72								
Н. кон.	Новінський В.П.													
Затвердив	Жураковська О.С.													