

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»
Факультет інформатики та обчислювальної техніки
(повна назва інституту/факультету)

Кафедра інформатики та програмної інженерії
(повна назва кафедри)

«До захисту допущено»

Завідувач кафедри

_____ Едуард ЖАРІКОВ
(підпис) (ім'я прізвище)

“ ” _____ 2023 р.

Дипломний проєкт

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Інженерія програмного забезпечення
комп'ютерних систем»

спеціальності «121 Інженерія програмного забезпечення»

на тему: Програмне забезпечення для організації навчання в дистанційному
режимі

Виконав студент IV курсу, групи ІТ-93
(шифр групи)

Григор'єв Антон Сергійович
(прізвище, ім'я, по батькові)

_____ (підпис)

Керівник асистент, Носов К.С.
(посада, науковий ступінь, вчене звання, прізвище та ініціали)

_____ (підпис)

Консультант з графічної документації ст. викл. Головченко М. М.
(посада, науковий ступінь, вчене звання, прізвище та ініціали)

_____ (підпис)

Рецензент ст. викл. Шемседінов Т.Г.
(посада, науковий ступінь, вчене звання, прізвище та ініціали)

_____ (підпис)

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших
авторів без відповідних посилань.

Студент _____
(підпис)

Київ –2023

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 121 Інженерія програмного забезпечення

Освітньо-професійна програма – Інженерія програмного забезпечення
комп'ютерних систем

ЗАТВЕРДЖУЮ

Завідувач кафедри

(підпис) Едуард ЖАРІКОВ
(ім'я прізвище)

“ ____ ” _____ 2023 р.

ЗАВДАННЯ
на дипломний проєкт студенту

Григор'єву Антону Сергійовичу
(прізвище, ім'я, по батькові)

Тема проєкту Програмне забезпечення для організації дистанційного навчання

керівник проєкту Носов Константин Сергійович, асистент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «31» травня 2023 р. №2101-с

2. Термін подання студентом проєкту « 17 » червня 2023 року

3. Вихідні дані до проєкту: технічне завдання

4. Зміст пояснювальної записки

1) Аналіз вимог до програмного забезпечення: загальні положення, змістовий опис та аналіз предметної області, аналіз успішних ІТ-проєктів, аналіз вимог до програмного забезпечення, постановка задачі.

2) Моделювання та конструювання програмного забезпечення: моделювання та аналіз програмного забезпечення, архітектура програмного забезпечення, аналіз безпеки даних.

3) Аналіз якості та тестування програмного забезпечення: аналіз якості ПЗ, опис процесів тестування.

4) Впровадження та супровід програмного забезпечення: розгортання програмного забезпечення, робота з програмним забезпеченням.

5. Перелік графічного матеріалу

1) Схема бази даних _____

2) Схема структурна класів програмного забезпечення _____

3) Схема структурна варіантів використання _____

6. Консультанти розділів проєкту

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

7. Дата видачі завдання «10» березня 2023 року _____

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Вивчення рекомендованої літератури	10.03.2023	
2	Аналіз існуючих методів розв'язання задачі	21.03.2023	
3	Постановка та формалізація задачі	30.03.2023	
4	Розробка інформаційного забезпечення	14.04.2023	
5	Алгоритмізація задачі	21.04.2023	
6	Обґрунтування вибору використаних технічних засобів	28.04.2023	
7	Розробка програмного забезпечення	15.05.2023	
8	Налагодження програми	20.05.2023	
9	Виконання графічних документів	25.05.2023	
10	Оформлення пояснювальної записки	30.05.2023	
11	Подання ДП на попередній захист	02.06.2023	
12	Подання ДП рецензенту	12.06.2023	
13	Подання ДП на основний захист	17.06.2023	

Студент

(підпис)

Антон ГРИГОР'ЄВ

(ініціали, прізвище)

Керівник

(підпис)

Константин НОСОВ

(ініціали, прізвище)

АНОТАЦІЯ

Пояснювальна записка дипломного проєкту складається з чотирьох розділів, містить 40 таблиць, 16 рисунків та 6 джерел – загалом 67 сторінки.

Дипломний проєкт присвячений програмному забезпеченню для організації навчання в дистанційному режимі.

Мета: розробка веб-додатка для організації навчання в дистанційному режимі.

Об'єкт дослідження: програмне забезпечення для створення, організації, і перегляду розкладів.

В першому розділі даної роботи було досліджено предметну область. Було проведено аналіз відомих технічних рішень та успішних аналогів.

В другому розділі описано моделювання та конструювання додатку. Була розроблена архітектура застосунку, описані програмні продукти, які були використані.

В третьому розділі було складено тестовий план, визначено об'єкти тестування, визначено підходи до тестування та описано основні задачі тестування.

В четвертому розділі наведено опис впровадження Web-додатку та представлено керівництво користувача.

КЛЮЧОВІ СЛОВА: ВЕБ-ДОДАТОК, ОСВІТА, JAVA, SPRING, MVC, HIBERNATE, MYSQL, ARCHITECTURE, БАЗА ДАНИХ, NETBEANS

ABSTRACT

The explanatory note of the diploma project consists of four sections, contains 40 tables, 16 figures and 6 sources – in total 78 pages.

The purpose of the diploma project is the development of software for organizing distance learning.

Objective: to develop a web application for organizing distance learning.

The research object is software for creating, organizing, and viewing schedules.

The first chapter of this work explores the subject area. An analysis of known technical solutions and successful analogs was conducted.

The second chapter describes the modeling and construction of the application. The application's architecture was developed, and the software products used were described.

The third chapter compiled a test plan, identified test objects, determined testing approaches, and described the main testing tasks.

The fourth chapter provides a description of the implementation of the web application and presents a user manual.

KEYWORDS: WEB APPLICATION, EDUCATION, JAVA, SPRING, MVC, HIBERNATE, MYSQL, ARCHITECTURE, DATABASE, NETBEANS.

Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

_____ Едуард ЖАРІКОВ

“ ____ ” _____ 2023 р.

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ОРГАНІЗАЦІЇ НАВЧАННЯ В
ДИСТАНЦІЙНОМУ РЕЖИМІ**

Технічне завдання

КПІ.IT-9308.045440.01.91

“ПОГОДЖЕНО”

Керівник проєкту:

Константин НОСОВ

Нормоконтроль:

Максим ГОЛОВЧЕНКО

Виконавець:

Антон ГРИГОР'ЄВ

Київ – 2023

ЗМІСТ

1	НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ.....	3
2	ПІДСТАВА ДЛЯ РОЗРОБКИ.....	4
3	ПРИЗНАЧЕННЯ РОЗРОБКИ.....	5
4	ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	6
4.1	Вимоги до функціональних характеристик	6
4.1.1	Функції користувачького інтерфейсу.....	6
4.1.2	Для користувача:.....	8
4.1.3	Для адміністратора системи (якщо він передбачений):	8
4.1.4	Додаткові вимоги:.....	9
4.2	Вимоги до надійності	9
4.3	Умови експлуатації.....	9
4.3.1	Вид обслуговування	9
4.3.2	Обслуговуючий персонал	9
4.4	Вимоги до складу і параметрів технічних засобів	9
4.5	Вимоги до інформаційної та програмної сумісності	10
4.5.1	Вимоги до вхідних даних.....	10
4.5.2	Вимоги до вихідних даних	10
4.5.3	Вимоги до мови розробки.....	10
4.5.4	Вимоги до середовища розробки	10
4.5.5	Вимоги до представленню вихідних кодів	10
4.6	Вимоги до маркування та пакування.....	10
4.7	Вимоги до транспортування та зберігання	10
4.8	Спеціальні вимоги	10
5	ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ	11
5.1	Попередній склад програмної документації	11
5.2	Спеціальні вимоги до програмної документації	11
6	СТАДІЇ І ЕТАПИ РОЗРОБКИ.....	12
7	ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ	13

1 НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ

Назва розробки: Програмне забезпечення для організації навчання в дистанційному режимі.

Галузь застосування: Освіта

Наведене технічне завдання поширюється на розробку програмного забезпечення для організації навчання в дистанційному режимі [045440], котра використовується для полегшення організації і відслідковування навчального розкладу та призначена для навчальних закладів різного рівня.

					КПІ.ІТ-9308.045440.01.91	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		3

2 ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки програмного забезпечення для організації навчання в дистанційному режимі є завдання на дипломне проектування, затверджене кафедрою інформатики та програмної інженерії Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

					КПІ.ІТ-9308.045440.01.91	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		4

3 ПРИЗНАЧЕННЯ РОЗРОБКИ

Розробка призначена для полегшення організації навчання в дистанційному режимі.

Метою розробки є забезпечення навчальних закладів зручним і простим веб-застосунком для створення і відслідковування навчального розкладу, а також іншої релевантної інформації.

					КПІ.ІТ-9308.045440.01.91	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		5

4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Вимоги до функціональних характеристик

Програмне забезпечення повинно забезпечувати виконання наступних основних функцій:

4.1.1 Користувацького інтерфейсу

— відображення розкладу з можливостями фільтрації (Рис. 4.1);

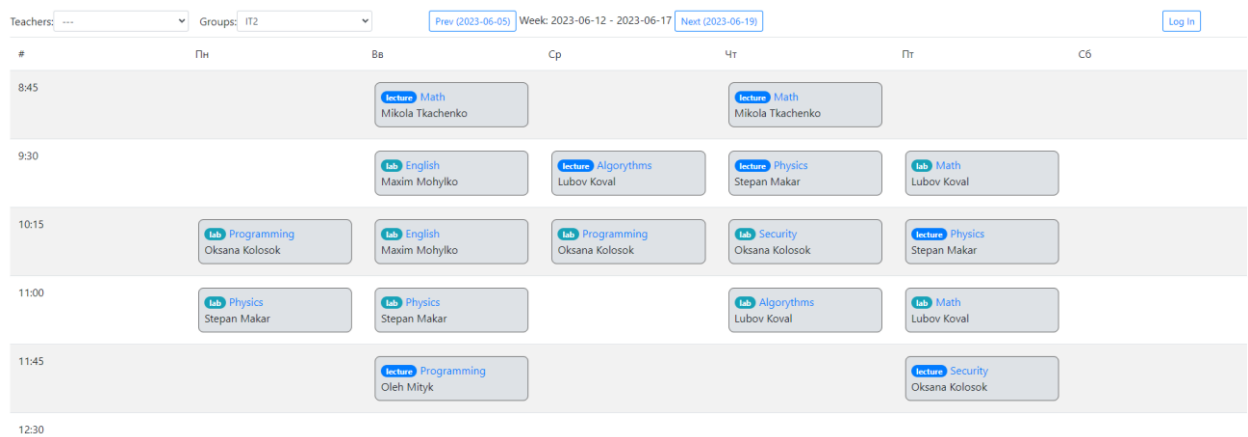


Рисунок 4.1 – Модель відображення розкладу

— відображення інформації про події (Рис. 4.2);

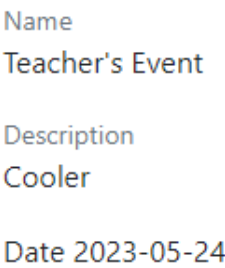


Рисунок 4.2 – Модель відображення інформації про події

— відображення інформації про курси (Рис. 4.3);

Back to Schedule

Name

Algorithms

Description

A great lesson

Рисунок 4.3 – Модель відображення інформації про курси

— авторизація (Рис. 4.4);

Авторизація

Email address

Password

☒ Remember me

[Forgot password?](#)

Sign in

Рисунок 4.4 – Модель інтерфейсу авторизації

— інтерфейс адміністратора (Рис. 4.5);

Back to main page

Save changes

Teachers: --- Groups: ---

Reset all

Assign all

Get current schedule

Download

Upload

Unassigned Lessons

+

#	Пн	Вт	Ср	Чт	Пт	Сб
8:45	+	lecture Math Mikola Tkachenko [IT1, IT2, IT3, IT4]	lecture Artificial Intelligence Mikola Tkachenko [IT3, IT4]	lecture Math Mikola Tkachenko [IT1, IT2, IT3, IT4]	+	+
9:30	lab Math Lubov Koval [IT1]	lab Math Mikola Tkachenko [IT4]	lecture Algorithms Lubov Koval [IT1, IT2, IT3, IT4]	lab Programming Oleh Mityk [IT4]	lab Programming Oleh Mityk [IT3]	+
	lab Math Mikola Tkachenko [IT4]	lab Physics Stepan Makar [IT1]	lecture Physics Stepan Makar [IT1, IT2]	lab Math Lubov Koval [IT2]	lab Math Lubov Koval [IT2]	
	lab English Maxim Mohylko [IT3]	lab English Maxim Mohylko [IT2]	lab Artificial Intelligence Mikola Tkachenko [IT3]	lab Physics Stepan Makar [IT1]		
10:15	lab Programming Oksana Kolosok [IT2]	lab Programming Oksana Kolosok [IT1]	lab Programming Oksana Kolosok [IT2]	lab English Maxim Mohylko [IT1]	lab Programming Oleh Mityk [IT4]	
	lab Math Lubov Koval [IT1]	lab English Maxim Mohylko [IT2]	lab Algorithms Lubov Koval [IT4]	lab Security Oksana Kolosok [IT2]	lab Math Lubov Koval [IT3]	

Рисунок 4.5 – Модель інтерфейсу адміністратора

4.1.2 Для користувача:

- Авторизація – користувач має мати можливість авторизуватися в власний аккаунт;
- перегляд розкладу певної групи і/або викладача – користувач повинен мати можливість переглядати розклад певної групи або викладача;
- перегляд інформації про курси і події – користувач має мати можливість переглядати різноманітну інформацію про курси і події;
- створення і редагування подій – користувач має мати можливість створювати власні події і редагувати їх;

4.1.3 Для адміністратора системи (якщо він передбачений):

- редагування розкладу – адміністратор має мати можливість редагувати розклад;
- автоматичне генерування розкладу – адміністратор має можливість запустити автоматичну генерацію розкладу;
- завантаженні розкладу на персональний комп'ютер – адміністратор має мати можливість завантажити розклад в вигляді файлу на власний комп'ютер;
- вивантаження розкладу з комп'ютеру в застосунок – адміністратор має мати можливість вивантажити розклад з раніше завантаженого файлу на сервер;
- додання користувачів – адміністратор має мати можливість додавати нових користувачів до системи;
- додання уроків і курсів – адміністратор має мати можливість додати нові уроки і курси до системи;

					КПІ.ІТ-9308.045440.01.91	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		8

4.1.4 Додаткові вимоги:

4.2 Вимоги до надійності

Передбачити контроль введення інформації та захист від некоректних дій користувача. Забезпечити цілісність інформації в базі даних (якщо в ПЗ передбачена наявність БД).

4.3 Умови експлуатації

Умови експлуатації згідно СанПін 2.2.2.542 – 96.

4.3.1 Вид обслуговування

Вимоги до виду обслуговування не висуваються.

4.3.2 Обслуговуючий персонал

Адміністратор повинен додавати користувачів, уроки і курси, а також редагувати розклад.

4.4 Вимоги до складу і параметрів технічних засобів

Мінімальна конфігурація технічних засобів:

- тип процесору: Intel Core i5;
- об'єм ОЗП: 4 Гб;
- підключення до мережі Інтернет зі швидкістю від 20 мегабіт;

Рекомендована конфігурація технічних засобів:

- тип процесору: Intel Core i7;
- об'єм ОЗП: 8 Гб;
- підключення до мережі Інтернет зі швидкістю від 100 мегабіт;

4.5 Вимоги до інформаційної та програмної сумісності

Програмне забезпечення повинно працювати під управлінням операційних систем сімейства WIN32 (Windows'XP, Windows NT і т.д.) або Unix.

4.5.1 Вимоги до вхідних даних

HTTP запити з можливими URL параметрами.

4.5.2 Вимоги до вихідних даних

Відповіді сервера на HTTP запити, в HTML або JSON форматі.

4.5.3 Вимоги до мови розробки

Розробку виконати на мові програмування: Java

4.5.4 Вимоги до середовища розробки

Розробку виконати на платформі Apache NetBeans

4.5.5 Вимоги до представлення вихідних кодів

Вихідний код програми має бути представлений у вигляді проекту NetBeans.

4.6 Вимоги до маркування та пакування

Вимоги до маркування та пакування не висуваються.

4.7 Вимоги до транспортування та зберігання

Вимоги до транспортування та зберігання не висуваються.

4.8 Спеціальні вимоги

Спеціальні вимоги не представляються.

					КПІ.ІТ-9308.045440.01.91	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		10

5 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

5.1 Попередній склад програмної документації

У склад супроводжувальної документації повинні входити наступні документи на аркушах формату А4:

- пояснювальна записка;
- технічне завдання;
- текст програми;
- програма та методика тестування;
- керівництво користувача;
- керівництво адміністратора.

Графічна частина повинна бути виконана на аркушах формату А3 та містити наступні документи:

- схема структурна варіантів використання;
- схема бази даних;
- схема структурна класів програмного забезпечення;

5.2 Спеціальні вимоги до програмної документації

Програмні модулі, котрі розробляються, повинні бути задокументовані, тобто тексти програм повинні містити всі необхідні коментарі.

6 СТАДІЇ І ЕТАПИ РОЗРОБКИ

№	Назва етапу	Строк	Звітність
1.	Вивчення літератури за тематикою проекту	10.03	
2.	Аналіз існуючих методів розв'язання задачі	21.03	
3.	Постановка та формалізація задачі	30.03	Технічне завдання
4.	Розробка інформаційного забезпечення	14.04	
5.	Алгоритмізація задачі	21.04	
6.	Обґрунтування вибору використаних технічних засобів	28.04	Схема структурна програмного забезпечення та специфікація компонентів (діаграма класів, схема алгоритму)
7.	Розробка програмного забезпечення	15.05	Тексти програмного забезпечення
8.	Налагодження програми	20.05	
9.	Виконання графічних документів	25.05	Графічний матеріал проекту
10.	Оформлення пояснювальної записки	30.05	Технічна документація

7 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ

Тестування розробленого програмного продукту виконується відповідно до “Програми та методики тестування”.

					КПІ.ІТ-9308.045440.01.91	Арк.
						13
Змін.	Арк.	№ докум.	Підп.	Дата.		

Пояснювальна записка до дипломного проєкту

на тему: Програмне забезпечення для організації навчання в дистанційному
режимі

КПІ.IT-9308.045440.02.81

Київ – 2023

ЗМІСТ

ВСТУП	4
1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	6
1.1 Загальні положення.....	6
1.2 Змістовний опис і аналіз предметної області.....	8
1.3 Аналіз існуючих технологій та успішних ІТ-проектів	10
1.3.1 Аналіз відомих алгоритмічних та технічних рішень	10
1.3.2 Аналіз допоміжних засобів розробки	12
1.3.3 Аналіз відомих програмних продуктів.....	18
1.4 Аналіз вимог до програмного забезпечення	24
1.4.1 Розроблення функціональних вимог	34
1.4.2 Розроблення нефункціональних вимог	37
1.5 Постановка задачі	38
Висновки до розділу	39
2 МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	41
2.1 Моделювання та аналіз програмного забезпечення.....	41
2.2 Архітектура програмного забезпечення.....	43
2.3 Конструювання програмного забезпечення.....	47
2.4 Аналіз безпеки даних.....	53
Висновки до розділу	55
3. АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	57
3.1 Аналіз якості ПЗ.....	57

3.2	Опис процесів тестування	58
3.3	Опис контрольного прикладу	67
	Висновки до розділу	71
4.	ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	71
4.1	Розгортання програмного забезпечення.....	72
4.2	Підтримка програмного забезпечення.....	73
	ВИСНОВКИ.....	75
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	77
	ДОДАТКИ.....	78

ВСТУП

Актуальність розробки вебсайту для організації дистанційного навчання полягає в зростаючому попиті на ефективні та зручні засоби для навчання на відстані. Завдяки швидкому розвитку технологій і доступності Інтернету, дистанційне навчання стає все більш поширеним у всьому світі. Різні освітні установи, компанії, тренери та викладачі шукають ефективні рішення для надання освітніх послуг в онлайн-форматі.

Світові тенденції показують, що розвиток технологій у сфері дистанційного навчання є пріоритетним напрямом. Віртуальні класи, інтерактивні платформи для навчання, онлайн-курси та засоби комунікації є поширеними рішеннями, які забезпечують ефективну освіту на відстані. Існують численні підходи до організації дистанційного навчання, включаючи відеоуроки, вебінари, інтерактивні завдання та тести.

Сучасний стан об'єкта розробки, тобто дистанційного навчання, є активно розвиваючимся. Відомі наукові установи та організації, такі як MIT, Harvard, Coursera, edX та інші, вже пропонують широкий спектр онлайн-курсів та ресурсів для дистанційного навчання. Ці установи активно досліджують та розробляють нові методики, технології та платформи для забезпечення високоякісного навчання на відстані.

Провідні вчені й фахівці в галузі дистанційного навчання наголошують на важливості інтерактивності, співпраці та персоналізації у процесі навчання на відстані. Застосування новітніх технологій, таких як штучний інтелект, віртуальна реальність, розширена реальність, дозволяють створювати більш зручні та захоплюючі навчальні середовища.

Можливі сфери застосування вебсайту для організації дистанційного навчання включають:

					КПІ.IT-9308.045440.02.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		4

– Вища освіта: Університети та коледжі можуть використовувати цей вебсайт для надання онлайн-курсів, віддалених лекцій, дистанційної консультації зі студентами тощо.

– Професійне навчання: Компанії та організації можуть використовувати вебсайт для організації онлайн-семінарів, тренінгів та курсів з підвищення кваліфікації для своїх співробітників.

– Навчання на відстані для школярів: Вебсайт може надавати можливість школярам отримувати дистанційну освіту, отримувати доступ до навчальних матеріалів, виконувати завдання та спілкуватися з вчителями.

– Онлайн-курси та навчальні ресурси: Вебсайт може функціонувати як платформа для розміщення онлайн-курсів з різних галузей знань, а також надавати доступ до навчальних матеріалів, відеолекцій, практичних завдань тощо.

– Розробка такого вебсайту дозволить забезпечити доступ до якісної освіти без обмежень простору та часу, що робить його вкрай актуальним у сучасному світі.

					КПІ.IT-9308.045440.02.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		5

1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1 Загальні положення

Аналізуючи предметну область застосунку для організування навчання в дистанційному режимі в формі онлайн-розкладу пар, можна зрозуміти, що такий веб-сайт може бути корисним для різних груп користувачів, таких як студенти, викладачі та адміністратори навчального закладу.

Загальний огляд предметної області включає розгляд таких аспектів [1]:

– Дистанційне навчання: Один з основних напрямків у сфері освіти, який передбачає використання електронних технологій для забезпечення доступу до навчального матеріалу, взаємодії студентів і викладачів на відстані. Дистанційне навчання дозволяє студентам здобувати знання незалежно від місця перебування і гнучко планувати свій навчальний розклад.

– Управління навчальним процесом: Це включає планування, організацію, контроль та оцінку навчального процесу відповідно до навчальних програм і вимог. Ефективне управління навчальним процесом в дистанційному режимі передбачає створення та підтримку розкладу занять, розподіл ресурсів (аудиторій, викладачів, матеріалів тощо) і врахування індивідуальних потреб студентів.

– Розкладування: Це процес планування розкладу занять студентів і викладачів. Він враховує різноманітні фактори, такі як доступні ресурси, обмеження щодо часу і приміщень, індивідуальні вимоги студентів і викладачів, а також інші важливі аспекти.

– Планування ресурсів: Це організація і розподіл різних ресурсів, таких як аудиторії, лабораторії, викладачі, матеріали, для забезпечення ефективного навчального процесу. Врахування наявних ресурсів і обмежень допомагає уникнути конфліктів та забезпечити оптимальне використання ресурсів.

– Технології e-learning: Використання електронних технологій, таких як веб-платформи, відеоконференції, електронні навчальні матеріали, для забезпечення дистанційного навчання і управління навчальним процесом.

Для студентів, такий веб-сайт може стати зручним інструментом для відслідковування свого навчального розкладу, зокрема для перевірки часу та місця проведення пар, в яких вони повинні бути присутніми. Крім того, вони можуть знайти на сайті важливі навчальні матеріали, які допоможуть їм підготуватися до занять.

Викладачі також можуть скористатися таким веб-сайтом для планування своїх занять, відслідковування часу та місця проведення пар, а також для забезпечення доступу своїм студентам до навчальних матеріалів.

Адміністратори навчального закладу можуть використовувати такий веб-сайт для керування навчальним розкладом, забезпечення синхронізації з іншими сервісами та надання доступу до навчальних матеріалів.

Оскільки такі веб-сайти використовуються в освітній сфері, вони повинні відповідати стандартам безпеки даних та захисту приватності. Крім того, вони повинні мати зручний та інтуїтивно зрозумілий інтерфейс для користувачів різних вікових та освітніх рівнів.

У світі існує багато веб-сайтів для навчальних закладів, таких як Blackboard, Moodle, Canvas та інші, які вже зарекомендували себе як ефективні засоби організації та відслідковування навчальних процесів. Проте, кожен з них має свої переваги та недоліки, що залежать від потреб користувачів та специфіки навчального закладу.

Один з переваг веб-сайту для відслідковування і організації навчальних пар може бути його інтеграція з іншими сервісами та програмами, такими як Google Calendar або Microsoft Outlook, для більш зручного відстеження розкладу. Також важливо враховувати можливість додавання сповіщень про найближчі заняття та інші нагадування, що допоможуть користувачам не пропустити важливі події.

Узагалі, успішна розробка веб-сайту для відслідковування та організації навчальних пар потребує багатьох функціональних та технічних аспектів, включаючи розробку бази даних, програмування функціоналу та створення інтуїтивно зрозумілого інтерфейсу. Крім того, такий сайт повинен відповідати законодавству щодо захисту персональних даних та інформації, яка передається користувачами.

1.2 Змістовний опис і аналіз предметної області

На поточний момент розвитку ІТ-технологій програмне забезпечення для організації навчання в дистанційному режимі з упором на організацію і редагування розкладу студентів і викладачів використовує знання предметної області для автоматизації процесів планування, управління та контролю у навчальних закладах. Основні етапи використання знань предметної області у програмному забезпеченні включають:

- Збір і аналіз вхідних даних: Програмне забезпечення отримує дані про студентів, викладачів, навчальні курси, доступні ресурси, а також обмеження, що стосуються часу та приміщень.

- Планування розкладу: На основі отриманих даних програмне забезпечення здійснює планування розкладу з урахуванням наявних ресурсів, обмежень та вимог студентів і викладачів.

- Управління розкладом: Після планування програмне забезпечення надає інтерфейс для редагування розкладу, внесення змін і забезпечення актуальності розкладу.

- Контроль та звітність: Програмне забезпечення відстежує виконання розкладу, надає звіти про прогрес студентів, відвідування, оцінки тощо.

Недоліками поточного стану речей в сфері ІТ для організації навчання в дистанційному режимі з упором на розклад можуть бути:

– Обмежені можливості редагування: Деякі програми не надають достатніх функцій для зручного редагування розкладу, що ускладнює адаптацію до змін.

– Відсутність гнучкості: Деякі системи не дозволяють налагоджувати розклад під потреби конкретного навчального закладу або враховувати індивідуальні потреби студентів та викладачів.

– Дороговизна: Значна частина систем для учбових закладів мають платну підписку, що може бути проблемою для певних навчальних закладів.

Для покращення ситуації в сфері ІТ розробки програмного забезпечення для організації навчання в дистанційному режимі з упором на розклад, можливими шляхами є:

– Розширення можливостей редагування: Розробка інтуїтивно зрозумілих інтерфейсів для редагування розкладу, що дозволять зручно вносити зміни і адаптуватись до потреб користувачів.

– Інтеграція з іншими системами: Підтримка інтеграції з іншими навчальними та управлінськими системами, такими як системи управління навчальним процесом, електронні журнали тощо.

– Гнучкість розкладу: Розробка алгоритмів та інструментів для налагодження розкладу з урахуванням специфіки кожного навчального закладу, а також індивідуальних потреб студентів і викладачів.

В рамках дипломного проекту може бути обраний шлях розробки програмного забезпечення, яке поєднує всі ці покращення: зручний інтерфейс для редагування розкладу, інтеграцію з іншими системами та гнучкість розкладу. Для цього можна використовувати сучасні технології розробки, такі як веб-додатки або мобільні додатки, з функціоналом, що задовольняє потреби користувачів і спрощує організацію навчального процесу в дистанційному режимі.

1.3 Аналіз існуючих технологій та успішних ІТ-проектів

1.3.1 Аналіз відомих алгоритмічних та технічних рішень

Існує багато підходів до розробки веб-застосунків. Основні види архітектур для веб-додатків включають наступні[2]:

– Монолітна архітектура: Монолітна архітектура є традиційним підходом до розробки веб-додатків. У такій архітектурі увесь функціонал додатка розміщений в одному монолітному програмному модулі, що запускається на сервері. Весь код додатка, як правило, компілюється в один файл, що спрощує розгортання та масштабування. Проте цей підхід може стати проблематичним у великих проектах, оскільки зростання функціональності може призвести до збільшення розміру моноліту і ускладнення розробки.

– Сервіс-орієнтована архітектура (SOA): Сервіс-орієнтована архітектура передбачає побудову додатка у вигляді набору послуг або сервісів. Кожен сервіс відповідає за конкретну функціональність і може бути розгорнутий та масштабований незалежно. Сервіси взаємодіють між собою за допомогою стандартизованих протоколів, таких як REST або SOAP. SOA дозволяє створювати гнучкі та розширювані додатки, але вимагає додаткового управління та витрат на комунікацію між сервісами.

– Клієнт-серверна архітектура: В цій архітектурі клієнт (наприклад, веб-браузер) взаємодіє з сервером через мережу. Клієнт звертається до сервера для отримання ресурсів, таких як HTML-сторінки, зображення або дані. Сервер обробляє запити та надсилає результати клієнту. Ця архітектура дозволяє розділити завдання між клієнтом та сервером, сприяючи масштабованості та ефективності.

– Мікросервісна архітектура: У мікросервісній архітектурі додаток розбивається на невеликі незалежні сервіси, які працюють разом для

виконання функціональних блоків. Кожен сервіс має свою власну базу коду та базу даних, і взаємодія між сервісами відбувається через API. Це дозволяє гнучко масштабувати та розгортати окремі компоненти додатку, полегшує супровід та поліпшує відновлення системи.

– Багатошарова архітектура (наприклад, MVC або MVVM): В цій архітектурі додаток розділяється на логічні шари або компоненти, які взаємодіють між собою. Наприклад, Model-View-Controller (MVC) розділяє логіку додатку на модель (дані та бізнес-логіку), представлення (відображення інтерфейсу користувача) та контролер (управління потоком даних та подій). Це спрощує розробку та тестування, робить додаток більш масштабованим та підтримуваним.

Кожна архітектура має свої переваги та недоліки, і вибір залежить від конкретних вимог, масштабу проекту та ресурсів, які доступні для розробки та супроводження додатку. Для даного проекту, найкращим варіантом буде багатошарова архітектура MVC завдяки її гнучкості і легкості розробки.

Важливим аспектом організації навчання є розробка розкладу[3]. Генерація розкладу навчальних закладів є складним процесом, який вимагає урахування різноманітних факторів, таких як доступні ресурси, вимоги програми навчання, обмеження часу та інші фактори. Основні алгоритми, які застосовуються для генерації розкладів навчальних закладів, включають:

– Жадібний алгоритм: Цей алгоритм базується на прийнятті найкращого рішення на кожному кроці, не враховуючи загальної оптимальності. Наприклад, розклад може починатися з призначення першого доступного викладача на першу доступну годину, а потім продовжуватися знаходженням наступних найкращих варіантів для інших уроків. Цей алгоритм досить простий і швидкий, але не завжди гарантує оптимальний розклад.

– Генетичні алгоритми: Генетичні алгоритми використовують метаевристичний підхід, імітуючи процес еволюції. Вони працюють зі збіркою

потенційних розкладів і використовують оператори схрещування, мутації та оцінки, щоб створити нові покоління розкладів. Цей алгоритм здатний досягати оптимальних або наближених оптимальних рішень, але вимагає багато обчислювальних ресурсів і тривалого часу для виконання.

– Пошук з локальними обмеженнями: Цей алгоритм постійно перевіряє розклад на виконання обмежень, таких як недопустиме перекриття часу або конфлікти ресурсів. Він намагається знайти оптимальні зміни в розкладі, щоб виправити ці проблеми. Алгоритм може застосовуватися ітеративно, поступово покращуючи розклад, аж до досягнення задовільного рівня.

– Методи математичного програмування: Методи лінійного програмування, динамічного програмування та інші математичні підходи можуть використовуватися для формалізації задачі генерації розкладу та пошуку оптимального рішення. Вони вимагають формалізації проблеми у вигляді математичних моделей та використовують оптимізаційні методи для знаходження розкладу з мінімальними витратами або іншими критеріями.

Ці алгоритми можуть застосовуватися окремо або комбінуватися для генерації розкладів навчальних закладів. Вибір конкретного алгоритму залежить від обмежень системи, вимог до розкладу та інших факторів, що впливають на процес генерації розкладу. Для даного проекту було вибрано жадібний алгоритм що, хоч і не завжди видає оптимальний результат, є досить гнучким і може бути модифікованим багатьма способами для підлаштування під потреби навчального закладу.

1.3.2 Аналіз допоміжних засобів розробки

Spring - це потужний фреймворк розробки програмного забезпечення для платформи Java. Він надає розширену підтримку для розробки інтернет-додатків та створення універсальних, масштабованих та легко змінюваних додатків.

					КПІ.IT-9308.045440.02.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		12

Spring пропонує багато різноманітних модулів, що дозволяють розробникам зосередитися на конкретних аспектах додатку, таких як керування залежностями, обробка веб-запитів, доступ до баз даних, безпека та багато іншого. Ці модулі можна використовувати окремо або в поєднанні з іншими, залежно від потреб проекту.

Одним з ключових аспектів Spring є інверсія керування (Inversion of Control, IoC)[4]. Вона дозволяє розробникам відокремлювати бізнес-логіку від інфраструктурних деталей, що спрощує тестування та розширення додатку. Замість того, щоб створювати об'єкти вручну, розробники використовують контейнер Spring, який автоматично керує створенням та керуванням об'єктами.

Spring також надає підтримку для аспектно-орієнтованого програмування (Aspect-Oriented Programming, AOP). Це дозволяє розробникам відокремлювати кросс-різновидові задачі, такі як журналювання, транзакції та безпека, від основного коду додатку. За допомогою аспектів розробники можуть легко впроваджувати ці функції в додаток без руйнування основної структури.

Основними перевагами використання Spring в цьому проекті є:

- Інверсія керування (Inversion of Control, IoC): Spring використовує принцип IoC, що дозволяє відокремити бізнес-логіку від інфраструктурних деталей. Фреймворк бере на себе керування створенням та керуванням об'єктами, що спрощує розробку, тестування та розширення додатків.

- Розширені можливості: Spring надає розширену підтримку для різних аспектів розробки, таких як керування транзакціями, безпека, кешування, планування завдань та багато іншого. Це дозволяє розробникам зосередитися на бізнес-логіці додатку, не вдаючись у деталі реалізації цих аспектів.

- Легкість тестування: Spring дозволяє легко тестувати додатки завдяки принципу IoC та підтримці вбудованих контейнерів для запуску тестів.

Розробники можуть легко підменювати залежності та виконувати модульне тестування різних компонентів додатку.

– Легкість інтеграції: Spring має велику екосистему та надає підтримку для інтеграції з іншими популярними фреймворками та бібліотеками, такими як Hibernate, JPA, Thymeleaf, Spring Security та інші. Це дозволяє розробникам використовувати найкращі інструменти для свого проекту та полегшує розробку комплексних додатків.

– Масштабованість: Spring надає потужні інструменти для масштабування додатків. Він підтримує розподілені системи, кешування, використання балансувальників навантаження та інші техніки, що допомагають підтримувати високу продуктивність додатків при зростанні обсягів даних та навантаження.

Узагальнюючи, Spring - це потужний фреймворк, який спрощує розробку веб-додатків на платформі Java, забезпечуючи широкий спектр функцій та підтримку для різних аспектів розробки. Він дозволяє розробникам писати більш чистий, модульний та легко тестований код, що сприяє покращенню продуктивності та розширюваності додатків.

Іншим важливим фреймворком для розробки цього програмного забезпечення є Hibernate[5]. Hibernate - це потужний фреймворк для об'єктно-реляційного відображення (Object-Relational Mapping, ORM) у Java. Він надає зручний спосіб зберігання та отримання об'єктів з бази даних, дозволяючи розробникам працювати з об'єктами, а не з SQL-запитами безпосередньо.

Основна ідея Hibernate полягає в тому, що розробники можуть визначати модель даних у вигляді класів Java, а Hibernate автоматично забезпечує відповідність між цими класами та таблицями в базі даних. Він використовує анотації або файл конфігурації для визначення співставлення між класами та таблицями, а також відношеннями між ними.

Hibernate надає багато функцій, які полегшують роботу з базами даних, таких як генерація SQL-запитів, кешування, керування транзакціями та індексація. Він підтримує різні режими взаємодії з базою даних, включаючи режим "лінивої" загрузки, коли дані з бази даних витягуються тільки при необхідності.

Одним з переваг Hibernate є його переносимість. Він може працювати з різними Системами Управління Базами Даних (СУБД), такими як MySQL, PostgreSQL, Oracle та іншими, дозволяючи розробникам працювати з об'єктами незалежно від конкретного СУБД.

Узагальнюючи, Hibernate - це потужний фреймворк ORM, який дозволяє розробникам працювати з об'єктами у Java, спрощуючи взаємодію з базою даних та полегшуючи розробку переносимих додатків. Він забезпечує високий рівень абстракції та автоматизацію при роботі з базами даних.

Цей фреймворк має наступні переваги:

- Інтеграція з іншими системами: Підтримка інтеграції з іншими навчальними та управлінськими системами, такими як системи управління навчальним процесом, електронні журнали тощо.

- Об'єктно-реляційне відображення (ORM): Hibernate надає механізм ORM, що дозволяє зв'язати об'єктну модель додатку з реляційною базою даних. Це спрощує роботу з базою даних, оскільки розробникам не потрібно вручну писати SQL-запити та керувати мапінгом між об'єктами та таблицями. ORM дозволяє автоматично зберігати, оновлювати та вибирати дані з бази даних за допомогою простих методів.

- Керування транзакціями: Hibernate надає механізми керування транзакціями, що спрощують роботу з операціями, які потребують атомарності та консистентності даних. Він дозволяє використовувати декларативний підхід до визначення транзакцій, що знижує кількість необхідного коду та спрощує управління транзакціями.

– Кешування: Hibernate підтримує механізми кешування, що дозволяють зберігати певні дані у пам'яті, щоб зменшити навантаження на базу даних та покращити продуктивність. Це особливо корисно для часто використовуваних даних, які не змінюються часто.

– Підтримка транзакційної безпеки: Hibernate забезпечує можливість використання механізму оптимістичного блокування, що дозволяє уникнути колізій під час одночасного доступу до даних кількома користувачами. Він також надає підтримку для ізоляції транзакцій та управління конкурентним доступом до даних.

– Масштабованість: Hibernate дозволяє створювати потужні та масштабовані додатки, оскільки він підтримує роботу з реляційними базами даних різних типів і конфігурацій. Він дозволяє ефективно працювати з великим обсягом даних та забезпечує широкі можливості для оптимізації запитів та доступу до даних.

– Підтримка стандартів: Hibernate реалізує стандарти Java Persistence API (JPA), що забезпечує переносимість між різними реалізаціями ORM. Це означає, що ви можете легко змінити реалізацію Hibernate на інший ORM-фреймворк, якщо знадобиться.

Загалом, використання фреймворку Hibernate дозволяє спростити роботу з базою даних, забезпечити швидкий доступ до даних та зберегти консистентність даних. Він є потужним інструментом для розробки веб-додатків, які використовують реляційну базу даних.

В якості середовища для розробки програмного забезпечення було вибрано Apache Net Beans. Це має кілька переваг[6]:

– Інтегроване середовище розробки (IDE): Apache NetBeans - це потужне інтегроване середовище розробки, яке надає широкі можливості для розробки веб-додатків. Воно має багато корисних функцій, таких як автодоповнення

коду, налагоджувач, система керування версіями, рефакторинг коду та багато іншого. Це полегшує розробку, покращує продуктивність та забезпечує зручне середовище для роботи.

– Підтримка Java та інших мов програмування: Apache NetBeans базується на платформі Java, що робить його ідеальним для розробки веб-додатків на основі Java. Він надає широкі можливості для програмування на Java, включаючи підтримку Java EE, Spring Framework, Hibernate і багато інших технологій, які є основою розробки проекту. Крім того, Apache NetBeans також підтримує інші мови програмування, такі як PHP та JavaScript, що дозволяє веб-додатки.

– Підтримка інтеграції з Apache-технологіями: Apache NetBeans є проектом Apache Software Foundation, тому він має добру підтримку інтеграції з іншими Apache-технологіями. Наприклад, ви можете легко інтегрувати ваш проект з Apache Tomcat або Apache Maven для розгортання та управління додатком. Це полегшує роботу з цими технологіями і забезпечує їх оптимальне використання.

– Активна спільнота користувачів та розробників: Apache NetBeans має велику спільноту користувачів та розробників, що сприяє активному розвитку та підтримці. Можливо можете знайти багато ресурсів, документацію, форуми та інші джерела, що допоможуть у вирішенні питань та проблем під час розробки. Також можна активно спілкуватися з іншими розробниками, обмінюватися досвідом та отримувати підтримку в разі потреби.

Загалом, використання Apache NetBeans для розробки цього проекту дозволить зручно працювати в потужному середовищі розробки з багатими функціональними можливостями та підтримкою різних мов програмування. Він полегшить роботу, забезпечить високу продуктивність та допоможе створити якісний веб-додаток.

1.3.3 Аналіз відомих програмних продуктів

Веб-сайт schedule.kpi.ua є офіційним ресурсом для перегляду розкладів занять студентами та викладачами Національного технічного університету України "Київський політехнічний інститут" (КПІ). Функція цього веб-сайту полягає у наданні зручного та швидкого доступу до інформації про розклад занять на певний день, тиждень або семестр для кожної групи та факультету КПІ.

На головній сторінці сайту можна вибрати потрібний факультет та навчальний курс, після чого користувач отримає доступ до розкладу занять на поточний тиждень. Також можна вибрати інший тиждень або семестр та переглянути розклад занять на цей період. У розкладі занять вказується час та місце проведення занять, прізвище викладача, назва дисципліни та група студентів, які беруть участь у занятті.

Окрім цього, на сайті schedule.kpi.ua можна знайти корисну інформацію про академічний календар, а також про розклад сесії та іспитів для кожної групи. Користувачі можуть також скористатися функцією пошуку для знаходження потрібної інформації швидко та зручно.

У загальному, функція веб-сайту schedule.kpi.ua полягає у забезпеченні студентів та викладачів КПІ необхідною та актуальною інформацією про розклад занять та інші важливі події, що стосуються навчального процесу.

На рисунку 1.1 можна побачити головну сторінку цього вебсайту.

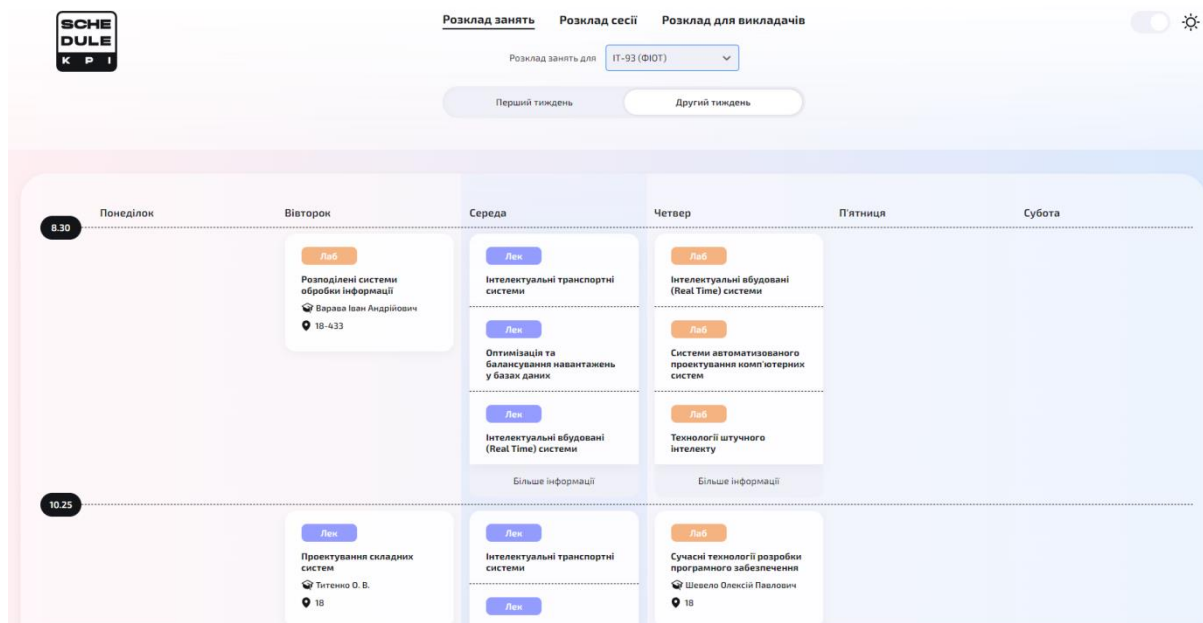


Рисунок 1.1 – Головна сторінка вебсайту schedule.kpi.ua

My Study Life - це додаток та веб-сервіс для планування навчального процесу, який дозволяє студентам та викладачам керувати своїм навчанням та графіком занять. Основним функціоналом My Study Life є наступні можливості.

- Календар - користувачі можуть створювати розклад занять, додавати дати екзаменів та терміни здачі робіт. Календар автоматично розподіляє заняття згідно з розкладом, а також дозволяє створювати нагадування для важливих подій.

- Розклад занять - користувачі можуть створювати розклад занять згідно з навчальним планом та додавати інформацію про викладачів та аудиторії.

- Задачі та домашні завдання - користувачі можуть створювати список завдань, що потрібно виконати та прикріплювати до них файли та додаткові відомості.

- Нотатки - користувачі можуть зберігати свої нотатки та матеріали до певних предметів прямо в додатку, щоб з легкістю знайти їх в майбутньому.

- Пошук - користувачі можуть шукати інформацію про свої курси, викладачів та аудиторії прямо в додатку.

– Сповіщення - користувачі отримують сповіщення про важливі події та нагадування про завдання та заняття.

– Синхронізація - додаток синхронізується між різними пристроями, що дозволяє користувачам звертатися до свого графіку та інформації з будь-якого пристрою з Інтернет-підключенням.

На рисунку 1.2 і 1.3 можна побачити приклади використання цього сервісу.

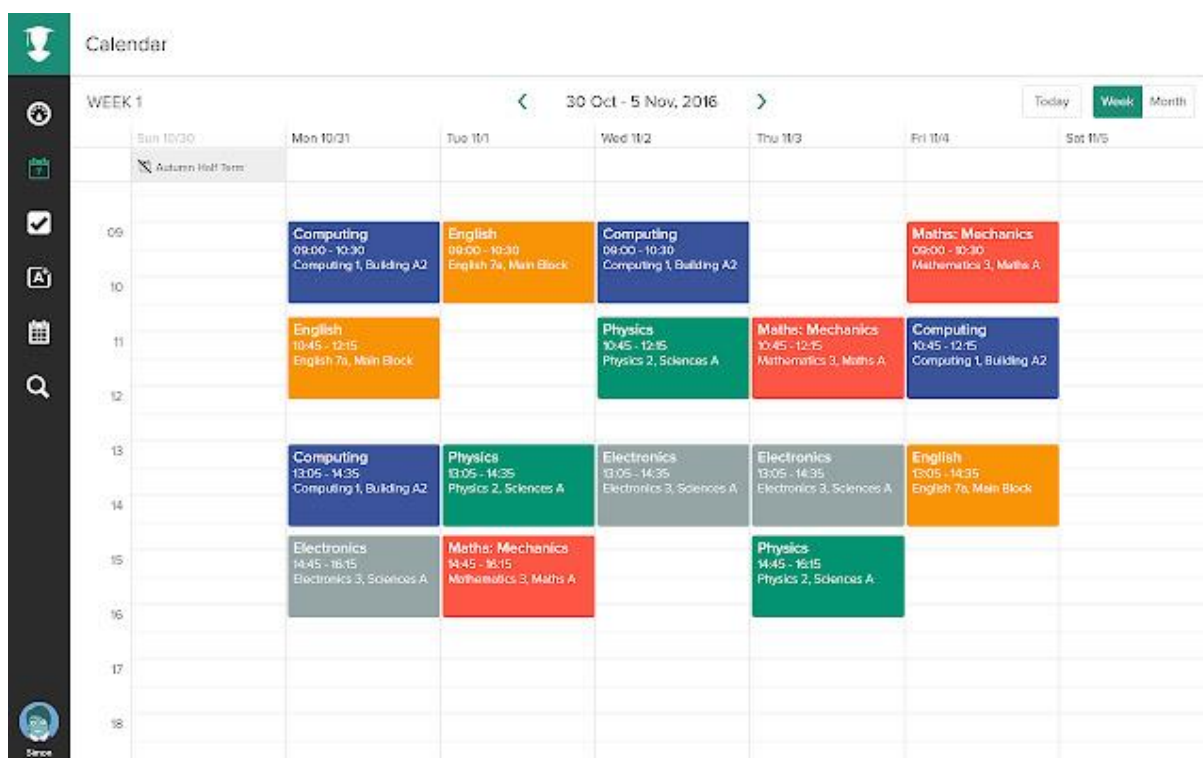


Рисунок 1.2 – Приклад розкладу створеного в My Study Life

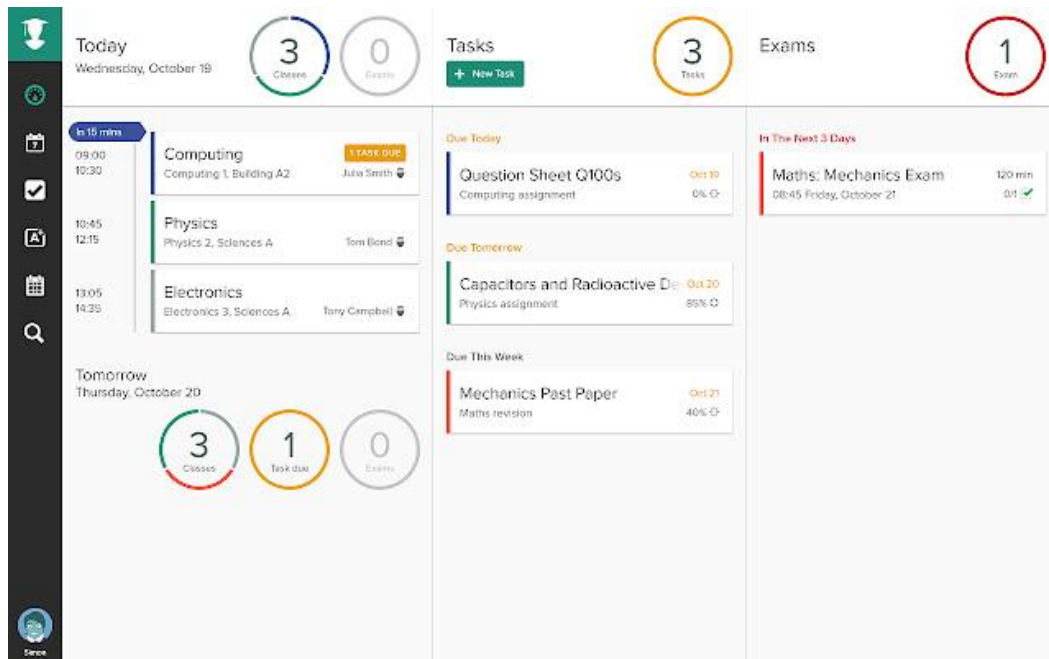


Рисунок 1.3 – Приклад використання My Study Life

TimeEdit - це веб-додаток для керування розкладами, який використовують в навчальних закладах, організаціях та компаніях. Основною метою TimeEdit є полегшення процесу планування та організації робочого часу, розкладів занять, термінів і подій.

Основні можливості TimeEdit:

- Створення та керування розкладом занять, що включає розклади курсів, лекцій, семінарів, лабораторних робіт та інших типів занять.
- Планування і керування подіями, такими як зустрічі, конференції та тренінги.
- Керування ресурсами, які використовуються в навчанні та роботі, такі як аудиторії, обладнання, матеріали тощо.
- Створення розкладів для викладачів та студентів з можливістю індивідуального налаштування та персоналізації.
- Пошук інформації про курси, викладачів та інші корисні матеріали.
- Можливість інтеграції з іншими системами, такими як Google Calendar, Microsoft Outlook тощо.

Загалом, TimeEdit дозволяє організувати робочий час і планування великої кількості подій та дозволяє уникнути плутанини в розкладах.

На рисунку 1.4 можна побачити приклад розкладу створеного за допомогою цього сервісу.

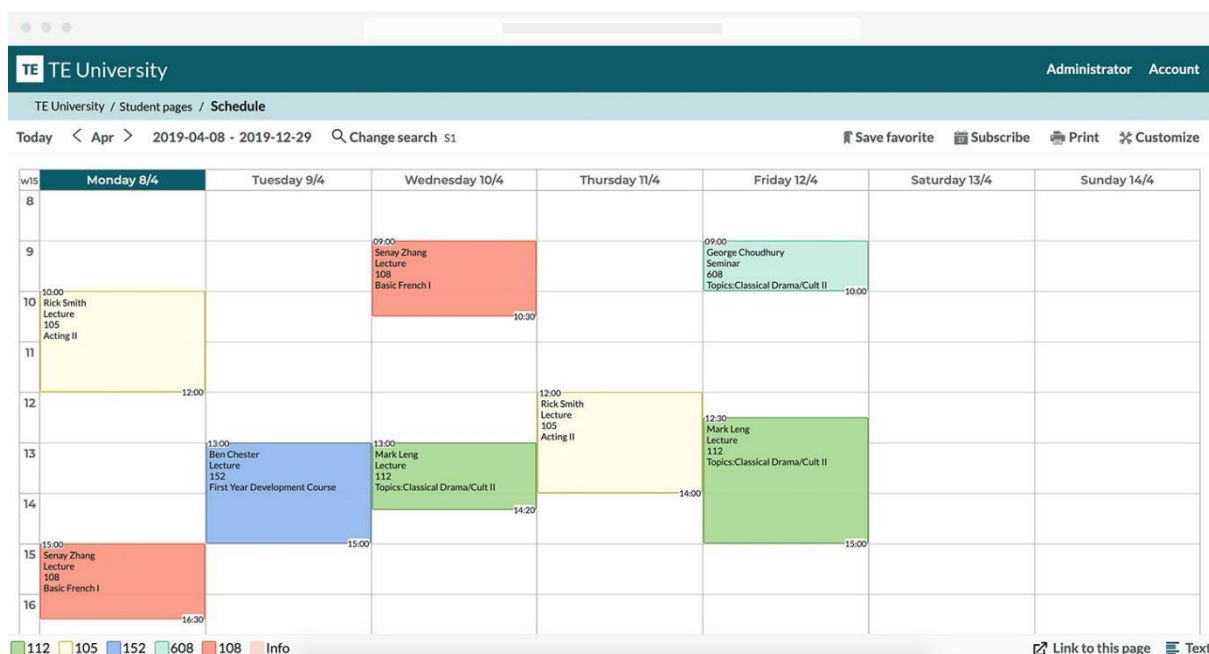


Рисунок 1.4 – Приклад розкладу створеного в TimeEdit

aSc TimeTables - це популярний інструмент для автоматичної генерації навчальних розкладів. Він надає багато функцій і можливостей для створення оптимального розкладу уроків. Ось огляд його функціоналу:

– Введення даних: aSc TimeTables дозволяє вводити різні дані, необхідні для створення розкладу. Це включає вчителів, класи, аудиторії, предмети, розкладні групи, перерви, дні тижня та інші параметри. Ви можете встановлювати обмеження і вказувати пріоритети для кожного з цих елементів.

– Автоматичне генерування розкладу: aSc TimeTables використовує потужні алгоритми розкладу, які автоматично створюють розклад уроків на основі введених даних. Він враховує обмеження, такі як доступність вчителів, аудиторій та класів, перекривання уроків і бажання вчителів. Застосунок пропонує кілька варіантів розкладу, серед яких можна вибрати найбільш підходящий.

– Оптимізація розкладу: aSc TimeTables надає можливість оптимізувати розклад, використовуючи різні параметри і правила. Ви можете встановлювати пріоритети для деяких уроків, задавати обмеження щодо розташування уроків у певних днях або періодах, а також налаштовувати часові інтервали для уроків.

– Візуалізація розкладу: aSc TimeTables надає зручний інтерфейс для відображення та редагування розкладу. Ви можете переглядати розклад у зручному графічному вигляді, змінювати розташування уроків за допомогою перетягування, вибирати альтернативні варіанти розкладу та додавати ручні зміни, якщо потрібно.

– Експорт та друк розкладу: aSc TimeTables дозволяє експортувати розклад у різні формати, такі як PDF, HTML, Excel і інші. Ви можете надрукувати розклад або експортувати його для подальшого використання.

– Звіти та статистика: aSc TimeTables надає різні звіти та статистику про розклад, такі як список вчителів, класів, аудиторій, перерв, розкладні групи і багато іншого. Це допомагає адміністраторам шкіл або університетів аналізувати та оптимізувати розклад.

aSc TimeTables є потужним інструментом для автоматичної генерації навчальних розкладів з багатим функціоналом, який дозволяє ефективно створювати та налаштовувати розклади відповідно до потреб користувача.

Як ми бачимо, існує немало застосунків і вебсайтів які можуть бути використані для організації дистанційного навчання. Однак, можна помітити

деякі тренди в наявному функціоналі. Програми призначені для організації календарів і подій, такі як Google Calendar/Classroom, зазвичай не мають функціоналу для генерації розкладу навчального закладу, в той час як програми створені для генерації розкладу зазвичай існують лише для цього і не мають функціоналу для допомоги самоорганізації студентів по ходу навчання. До того ж, ці програми зазвичай є платними, що робить їх менш привабливими варіантами для деяких навчальних закладів.

Таким чином, можна зробити висновок що безкоштовний Open-Source застосунок здатний на генерування розкладу і подальше використання протягом навчального процесу має свою нішу серед схожих продуктів і буде привабливим для певних користувачів.

1.4 Аналіз вимог до програмного забезпечення

Головною функцією програмного забезпечення є перегляд групового розкладу пар і подій, а також генерація цього розкладу Адміністратором, більше функцій можна побачити на рисунку 1.3.

В таблицях 1.2 - 1.14 наведені варіанти використання програмного забезпечення. UseCase діаграму наведено серед графічних матеріалів проекту, креслення “Схема Структурна Варіантів Використань”.

Таблиця 1.1 - Варіант використання UC-1

Use case name	Авторизація користувача
Use case ID	UC-01
Goals	Авторизація користувача в системі
Actors	Гість (незарєєстрований користувач)
Trigger	Користувач бажає авторизуватися
Pre-conditions	-

Продовження таблиці 1.1

Flow of Events	Користувач переходить на сторінку авторизації. В поля реєстрації вводяться відповідні дані - пошта і пароль - або користувач вибирає авторизацію через зовнішній сервіс, такий як Google, і вводить дані там.
Extension	В випадку введення некорректних даних користувач отримує сповіщення про помилку.
Post-Condition	Користувач авторизується в свій аккаунт.

Таблиця 1.2 - Варіант використання UC-2

Use case ID	UC-02
Goals	Перегляд розкладу певної групи
Actors	Будь-який користувач
Trigger	Користувач бажає подивитися розклад пар за певними параметрами.
Pre-conditions	-
Flow of Events	Користувач знаходиться на головній сторінці сайту. В відповідному меню, він вибирає необхідну йому групу.
Extension	-
Post-Condition	Розклад на сторінці оновлюється відповідно до вибору.

Таблиця 1.3 - Варіант використання UC-3

Use case ID	UC-03
Goals	Перегляд розкладу певного викладача
Actors	Будь-який користувач
Trigger	Користувач бажає подивитися розклад пар за певними параметрами.
Pre-conditions	-
Flow of Events	Користувач знаходиться на головній сторінці сайту. В відповідному меню, він вибирає необхідного йому викладача.
Extension	-
Post-Condition	Розклад на сторінці оновлюється відповідно до вибору.

Таблиця 1.4 - Варіант використання UC-4

Use case ID	UC-04
Goals	Перегляд власного розкладу
Actors	Студент або Викладач
Trigger	Користувач бажає подивитися власний розклад пар та подій.
Pre-conditions	-
Flow of Events	Користувач знаходиться на головній сторінці сайту і натискає відповідну кнопку.
Extension	-
Post-Condition	Розклад на сторінці оновлюється відповідно до вибору.

Таблиця 1.5 - Варіант використання UC-5

Use case ID	UC-05
Goals	Створення події
Actors	Студент або Викладач
Trigger	Користувач бажає створити нову подію.
Pre-conditions	Користувач має бути авторизований в системі.
Flow of Events	Користувач натискає відповідну кнопку і переходить на сторінку створення події, де заповнює відповідні поля - час, опис, а також групи яким подія буде видима.
Extension	Студент може вибрати лише свою групу, в той час як викладач може вибрати будь-які групи.
Post-Condition	Створено нову подію з вибраними параметрами.

Таблиця 1.6 - Варіант використання UC-6

Use case ID	UC-06
Goals	Редагування події
Actors	Студент або Викладач
Trigger	Користувач-викладач бажає відредагувати дані про створену ним подію
Pre-conditions	Користувач має бути авторизований в системі.
Flow of Events	Користувач переходить на сторінку відповідної події, де може змінити дані в відповідних полях.
Extension	-
Post-Condition	Дані про подію змінено.

Таблиця 1.7 - Варіант використання UC-7

Use case ID	UC-07
Goals	Видалення події
Actors	Студент або Викладач
Trigger	Користувач-викладач бажає видалити створену ним подію.
Pre-conditions	Користувач має бути авторизований в системі.
Flow of Events	Користувач переходить на сторінку відповідної події, де натискає відповідну кнопку.
Extension	-
Post-Condition	Подію видалено.

Таблиця 1.8 - Варіант використання UC-8

Use case ID	UC-08
Goals	Перегляд даних користувача
Actors	Будь-який користувач
Trigger	Користувач бажає переглянути дані певного зареєстрованого користувача
Pre-conditions	-
Flow of Events	Користувач переходить на сторінку цікавого йому аккаунта.
Extension	-
Post-Condition	Користувач бачить певний набір відкритих даних

Таблиця 1.9 - Варіант використання UC-9

Use case ID	UC-09
Goals	Редагування персональних даних
Actors	Студент або Викладач
Trigger	Користувач бажає відредагувати персональні дані
Pre-conditions	Користувач має бути авторизований в системі
Flow of Events	Користувач переходить на сторінку власного аккаунта, де змінює відповідні поля.
Extension	-
Post-Condition	Персональні дані (відкриті або закриті) користувача змінено

Таблиця 1.10 - Варіант використання UC-10

Use case ID	UC-10
Goals	Перегляд даних курсу
Actors	Будь-який користувач
Trigger	Користувач бажає переглянути дані певного курсу
Pre-conditions	-
Flow of Events	Користувач переходить на сторінку цікавого йому курсу.
Extension	-
Post-Condition	Користувач бачить певний набір відкритих даних

Таблиця 1.11 - Варіант використання UC-11

Use case ID	UC-11
Goals	Редагування даних курсу
Actors	Викладач
Trigger	Користувач-викладач бажає відредагувати дані про курс на якому викладає
Pre-conditions	Користувач має бути авторизований в системі як викладач
Flow of Events	Користувач переходить на сторінку відповідного курсу, де змінює відповідні поля.
Extension	-
Post-Condition	Дані про курс змінено

Таблиця 1.12 - Варіант використання UC-12

Use case ID	UC-12
Goals	Додавання, редагування, та видалення користувача
Actors	Адміністратор
Trigger	Адміністратор бажає змінити інформацію про користувачів системи.
Pre-conditions	-
Flow of Events	Адміністратор, використовуючи власний інтерфейс, заходить в список користувачів. Там він може створити нового, заповнивши необхідні поля, відредагувати існуючого, або видалити існуючого користувача.
Extension	-

Продовження таблиці 1.12:

Post-Condition	Залежно від дій Адміністратора, користувача додано, відредаговано, чи змінено.
----------------	--

Таблиця 1.13 - Варіант використання UC-13

Use case ID	UC-13
Goals	Додавання, редагування, та видалення групи
Actors	Адміністратор
Trigger	Адміністратор бажає змінити інформацію про групи
Pre-conditions	-
Flow of Events	Адміністратор, використовуючи власний інтерфейс, заходить в список груп. Там він може створити нову, заповнивши необхідні поля, відредагувати існуючу, або видалити існуючу групу.
Extension	-
Post-Condition	Залежно від дій Адміністратора, групу додано, відредаговано, чи змінено.

Таблиця 1.14 - Варіант використання UC-14

Use case ID	UC-14
Goals	Додавання, редагування, та видалення курсу
Actors	Адміністратор
Trigger	Адміністратор бажає змінити інформацію про курси в системі.
Pre-conditions	-

Продовження таблиці 1.14:

Flow of Events	Адміністратор, використовуючи власний інтерфейс, заходить в список курсів. Там він може створити новий, заповнивши необхідні поля, відредагувати існуючий, або видалити існуючий курс.
Extension	-
Post-Condition	Залежно від дій Адміністратора, курс додано, відредаговано, чи змінено.

Таблиця 1.15 - Варіант використання UC-15

Use case ID	UC-15
Goals	Додавання, редагування, та видалення уроку
Actors	Адміністратор
Trigger	Адміністратор бажає змінити інформацію про уроки в розкладі.
Pre-conditions	-
Flow of Events	Адміністратор, використовуючи власний інтерфейс, заходить в список уроків. Там він може створити новий, заповнивши необхідні поля, відредагувати існуючий, або видалити існуючий урок.
Extension	-
Post-Condition	Залежно від дій Адміністратора, урок додано, відредаговано, чи змінено.

Таблиця 1.16 - Варіант використання UC-16

Use case ID	UC-16
Goals	Редагування розкладу
Actors	Адміністратор
Trigger	Адміністратор бажає змінити розклад уроків/пар.
Pre-conditions	-
Flow of Events	Адміністратор, використовуючи власний інтерфейс, заходить в редактор розкладу. Він може змінювати положення пар або генерувати новий розклад. Після редагування, він нажимає кнопку “Зберегти”.
Extension	-
Post-Condition	Залежно від дій Адміністратора, розклад змінено.

Таблиця 1.17 - Варіант використання UC-17

Use case ID	UC-17
Goals	Завантаження розкладу
Actors	Адміністратор
Trigger	Адміністратор бажає завантажити створений розклад
Pre-conditions	Адміністратор знаходиться в редакторі розкладу.
Flow of Events	Адміністратор нажимає відповідну кнопку.
Extension	-
Post-Condition	Excel-файл з розкладом завантажується до комп'ютера Адміністратора.

Таблиця 1.18 - Варіант використання UC-18

Use case ID	UC-18
Goals	Вивантаження розкладу
Actors	Адміністратор
Trigger	Адміністратор бажає вивантажити раніше завантажений розклад назад на сервер.
Pre-conditions	Адміністратор знаходиться в редакторі розкладу.
Flow of Events	Адміністратор нажимає відповідну кнопку, після чого вибирає потрібний файл.
Extension	-
Post-Condition	Розклад в редакторі змінюється відповідно до вмісту файлу.

1.4.1 Розроблення функціональних вимог

В таблицях 1.15 – 1.20 наведений опис функціональних вимог до програмного забезпечення.

Таблиця 1.15 – Функціональна вимога FR-01

Назва	Авторизація користувача
Опис	Система повинна надавати можливість авторизації користувачеві шляхом введення пошти і паролю, або за допомогою використання аккаунта Google.

Таблиця 1.16 – Функціональна вимога FR-02

Назва	Перегляд розкладу
-------	-------------------

Продовження таблиці 1.16:

Опис	Система повинна надавати можливість користувачеві перегляд розклад певної групи, викладача, або власний розклад що включає в себе події.
------	--

Таблиця 1.17 – Функціональна вимога FR-03

Назва	Створення і редагування подій
Опис	Система повинна надавати можливість користувачеві створювати, редагувати, і видаляти події, видимі йому і певним групам.

Таблиця 1.18 – Функціональна вимога FR-04

Назва	Редагування персональних даних
Опис	Система повинна надавати можливість користувачеві редагувати персональні дані, видимі іншим користувачам.

Таблиця 1.19 – Функціональна вимога FR-05

Назва	Редагування даних курсу
Опис	Система повинна надавати можливість викладачам редагувати дані курсів на яких вони викладають.

Таблиця 1.20 – Функціональна вимога FR-06

Назва	Редагування даних адміністратором
Опис	Система повинна надавати можливість адміністратору створювати, редагувати, і видаляти користувачів, курси, групи, і уроки.

Таблиця 1.21 – Функціональна вимога FR-07

Назва	Редагування розкладу
Опис	Система повинна надавати можливість адміністратору редагувати розклад уроків/пар.

Таблиця 1.22 – Функціональна вимога FR-08

Назва	Збереження розкладу
Опис	Система повинна надавати можливість адміністратору завантажувати створений розклад на власний комп'ютер, і вивантажити розклад з власного комп'ютера на сервер.

В таблиці 1.23 можна побачити модель вимог.

Таблиця 1.23 – Модель вимог.

Вимога	Код	Пріоритет	Ризик
Авторизація користувача	FR-01	Високий	Низький
Валідація полів	FR-01.1	Низький	Низький
Перегляд розкладу	FR-02	Високий	Низький
Сортування розкладу	FR-02.1	Високий	Низький
Створення і редагування подій	FR-03	Середній	Низький
Видалення подій	FR-03.1	Низький	Низький
Редагування персональних даних	FR-04	Низький	Низький
Редагування даних курсу	FR-05	Високий	Низький
Редагування даних адміністратором	FR-06	Середній	Низький
Редагування розкладу адміністратором	FR-07	Високий	Низький
Автоматична генерація розкладу	FR-07.1	Середній	Низький
Збереження розкладу	FR-08	Низький	Низький
Вивантаження розкладу	FR-08.1	Низький	Низький

На рисунку 1.5 можна побачити матрицю трасування вимог.

	FR-01	FR-02	FR-03	FR-04	FR-05	FR-06	FR-07	FR-08
UC-01								
UC-02								
UC-03								
UC-04								
UC-05								
UC-06								
UC-07								
UC-08								
UC-09								
UC-10								
UC-11								
UC-12								
UC-13								
UC-14								
UC-15								
UC-16								
UC-17								
UC-18								

Рисунок 1.5 – Матриця трасування вимог

1.4.2 Розроблення нефункціональних вимог

Основні нефункціональні вимоги включають в себе:

- Ефективність та продуктивність: Система повинна бути швидкою та ефективною в роботі з великим обсягом даних та багатьма користувачами.
- Надійність: Система повинна бути надійною та не відмовляти у роботі навіть при великих навантаженнях або при виникненні помилок.
- Стійкість: Система повинна бути стійкою до злому, хакерських атак та вірусів, що можуть загрожувати безпеці користувачів.
- Масштабованість: Система повинна бути масштабованою та мати можливість збільшення її потужності при зростанні обсягу даних та користувачів.
- Сумісність: Система повинна бути сумісною з різними операційними системами та браузерами, що використовуються користувачами.

– Підтримка безперервної роботи: Система повинна мати можливість підтримувати безперервну роботу протягом 24 годин на добу, 7 днів на тиждень.

– Легкість установки та налаштування: Система повинна бути легкою в установці та налаштуванні, щоб користувачі могли швидко розпочати використання системи.

– Підтримка стандартів: Система повинна підтримувати стандарти веб-розробки, такі як HTML, CSS та JavaScript, щоб забезпечити сумісність з різними браузерами та платформами.

– Зручна документація: Система повинна мати детальну та зручну для користування документацію, щоб допомогти користувачам в розумінні функціоналу системи та розв'язанні проблем.

1.5 Постановка задачі

Метою проекту є розробка веб-сайту для організації та відслідковування навчальних пар. Система повинна забезпечувати реалізацію динамічного та інтерактивного розкладу навчальних пар, спрощеного доступу до навчальних матеріалів та синхронізації з зовнішніми сервісами задля відображення інформації в одному місці. Система створюється для підтримки групового навчання в заданому середовищі і централізованій системі навчального закладу, без необхідності (але і без можливості) самотійно реєструватися в системі або записуватися на курси – програмне забезпечення передбачає наявний розклад створений навчальним закладом. Індивідуальна гнучкість буде забезпечуватися за рахунок можливості створення подій, видимих як індивідуально користувачу, так і певним групам студентів.

Для досягнення цієї мети поставлено наступні задачі:

– Авторизація користувачів;

- Створення ієрархії ролей (викладач, студент, адміністратор), з різним рівнем можливостей;
- Створення адміністратором нових аккаунтів користувачів і розподілення їх по групам;
- Створення адміністратором розкладу що складається з уроків/пар;
- Створення алгоритму автоматичної генерації розкладу на основі наявної інформації;
- Створення системи завантаження і вивантаження розкладу адміністратором;
- Відображення розкладу;
- Створення користувачами подій, видимих їм і певним групам (залежно від ролі користувача);

Висновки до розділу

У розділі "Аналіз вимог програмного забезпечення" було проведено детальне дослідження та аналіз різних аспектів, пов'язаних з програмним забезпеченням для задачі, що вивчається. Проведений аналіз допоміг уявити загальну картину вимог до програмного забезпечення і зрозуміти важливі аспекти, які потрібно врахувати під час розробки.

Початкові загальні положення надали загальну інформацію про характер проекту та його мету, створюючи основу для подальшого аналізу. Змістовий опис і аналіз предметної області розкрив важливі аспекти, пов'язані з контекстом, в якому функціонуватиме програмне забезпечення. Цей аналіз допоміг зрозуміти основні вимоги до функціональності та визначити потреби користувачів.

Аналіз існуючих технологій та успішних ІТ-проектів надав цінний інсайт у те, які рішення та практики вже успішно використовуються в аналогічних проектах. Цей аналіз вказав на потенційні можливості і виклики,

які пов'язані з використанням технологій та запровадженням програмного забезпечення в реальне середовище.

Аналіз вимог до програмного забезпечення був виконаний з метою ідентифікації конкретних вимог та функціональних характеристик, які має мати розроблюване програмне забезпечення. Цей аналіз став основою для подальшого проектування та розробки системи, забезпечуючи чітку постановку завдання та визначення вимог.

В цілому, робота над цим розділом надала цінну інформацію та базовий фундамент для подальшої розробки програмного забезпечення. Було визначено ключові вимоги та характеристики, а також проведений загальний огляд предметної області та існуючих технологій. Цей аналіз дозволить ефективно переходити до наступних етапів розробки та реалізації системи з врахуванням потреб і очікувань користувачів.

					КПІ.IT-9308.045440.02.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		40

2 МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Моделювання та аналіз програмного забезпечення

Для моделювання та аналізу програмного забезпечення, було вирішено використати стандарт BPMN. Це стандарт для моделювання бізнес-процесів за допомогою графічних символів. Він надає єдиний мовний засіб для опису процесів, що дозволяє бізнес-аналітикам, проектним командам та стейкхолдерам узгоджувати, спілкуватися і визначати бізнес-процеси.

Було визначено наступні ключові бізнес-процеси для даної роботи:

- авторизація користувача (вчителя чи студента);
- створення користувачем події;
- редагування користувачем події;
- редагування користувачем-викладачем курсу;
- перегляд користувачем розкладу за певними параметрами;
- редагування адміністратором розкладу;

Модель цих процесів присутні на рисунку 2.1. Присутні й інші бізнес-процеси, що задовільняють решту функціоналу роботи.

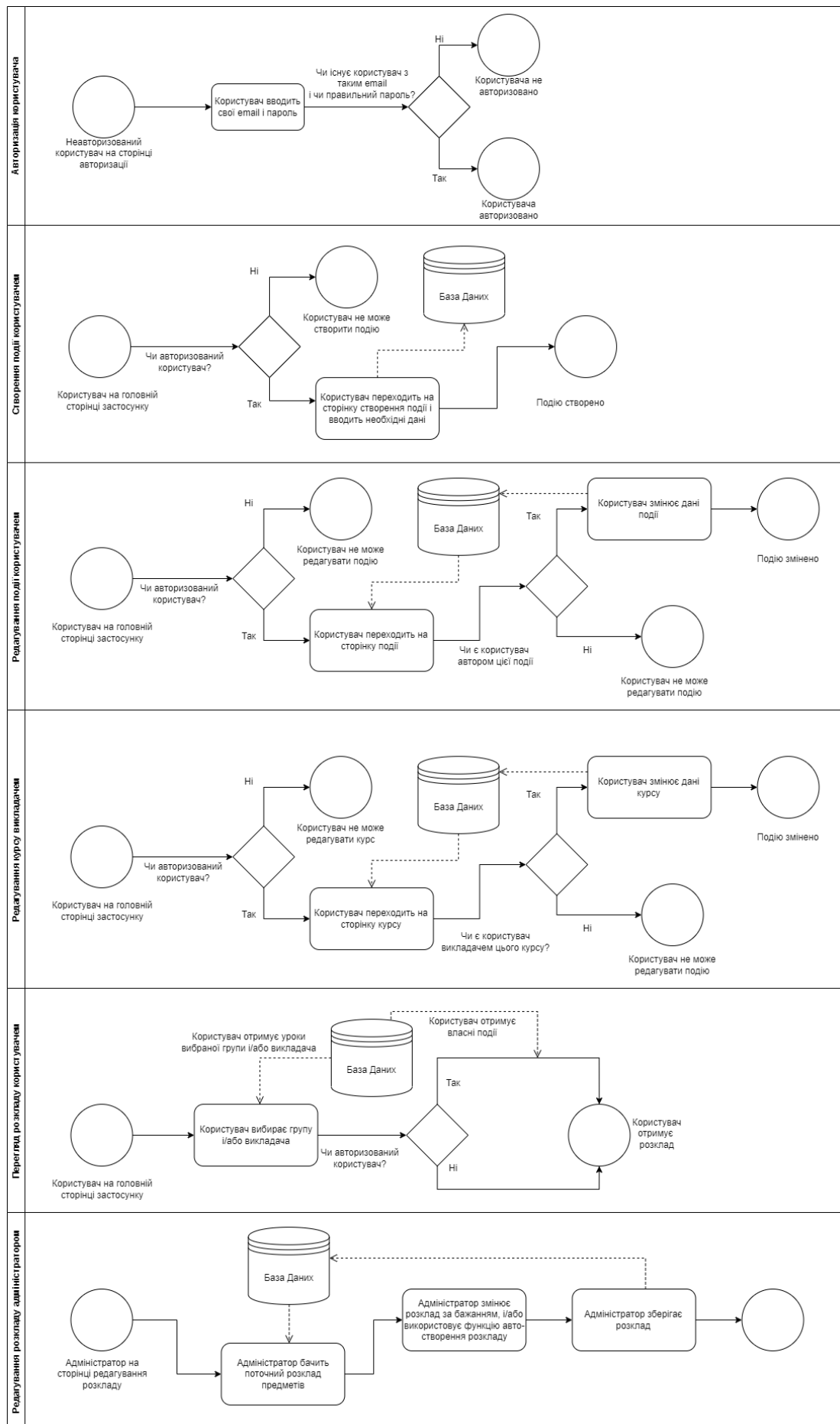


Рисунок 2.1 – Ключові бізнес-процеси застосунку

2.2 Архітектура програмного забезпечення

Для даного проекту було вибрано багат шарову архітектуру, з використанням паттерну MVC і фреймворку Java Spring. Основними причинами такого вибору є:

- Розділення відповідальностей: Багат шарова архітектура дозволяє відокремити різні компоненти проекту, такі як представлення, бізнес-логіку та доступ до даних, і призначити відповідні відповідальності кожному шару. Це робить код більш організованим, підтримуваним та розширюваним.

- Масштабованість: Багат шарова архітектура дозволяє розширювати і масштабувати додаток залежно від потреб користувачів та бізнесу. Кожен шар можна розглядати як окремий компонент, який можна змінювати та замінювати без впливу на інші шари.

- Використання стандартів: Spring MVC використовує стандарти, такі як Java Servlet API та JavaServer Pages (JSP), що дозволяє зробити додаток більш сумісним та переносимим між різними середовищами.

Узагалі, використання багат шарової архітектури з використанням Spring MVC робить проект більш структурованим, масштабованим та легким для розуміння та розширення.

Багат шарова архітектура (Multilayer Architecture) може бути використана в даному проекті для покращення модульності, розширюваності та підтримки принципів розділення відповідальностей. Вона розбиває систему на логічні шари, кожен з яких відповідає за свої конкретні завдання. Основними шарами є:

- Представлення (Presentation Layer): Цей шар відповідає за відображення інтерфейсу користувача та обробку взаємодії з користувачем. Він може включати компоненти, такі як веб-сторінки, шаблони, контролери і

представлення. Цей шар забезпечує взаємодію користувача з системою та передає запити до наступних шарів.

– Бізнес-логіка (Business Logic Layer): Цей шар містить логіку, пов'язану з обробкою бізнес-правил, обчисленнями та маніпуляціями з даними. Він включає компоненти, такі як сервіси, менеджери, доменні об'єкти тощо. Цей шар відповідає за виконання функціональних операцій, таких як обробка запитів, перевірка прав доступу, виконання бізнес-правил та взаємодія з базою даних.

– Доступ до даних (Data Access Layer): Цей шар забезпечує доступ до даних та взаємодію з базою даних або іншими джерелами даних. Він містить компоненти, такі як репозиторії, провайдери доступу до даних, моделі даних тощо. Цей шар відповідає за збереження та отримання даних з різних джерел, а також за виконання операцій з базою даних.

Використання багатошарової архітектури дозволить створити добре організовану, модульну та розширювану систему, яка буде масштабована та забезпечуватиме високу якість програмного продукту. Кожен шар виконує конкретну функцію і має чітко визначені обов'язки, що спрощує розробку, тестування та підтримку системи.

MVC (Model-View-Controller) є архітектурним патерном, що використовується для організації коду в програмах. Основною ідеєю паттерну є розділення програмного застосування на три компоненти: Модель (Model), Вид (View) та Контролер (Controller). Кожен з цих компонентів виконує свою функцію і взаємодіє з іншими компонентами за допомогою визначених інтерфейсів.

– Модель (Model) відповідає за управління даними та бізнес-логікою програми. Вона представляє собою структуру даних та методи для роботи з цими даними. Модель незалежна від інших компонентів і не залежить від того, як дані відображаються або контролюються.

– Вид (View) відповідає за представлення даних користувачу. Він відображає інформацію з Моделі та забезпечує інтерфейс для взаємодії користувача з програмою. Вид може бути графічним інтерфейсом, веб-сторінкою або будь-яким іншим способом візуалізації даних.

– Контролер (Controller) координує взаємодію Моделі та Виду. Він отримує вхідні дані від користувача через Вид, виконує необхідні дії з Моделлю та оновлює Вид залежно від результатів. Контролер також відповідає за маршрутизацію запитів та управління подіями.

Загальна ідея MVC полягає в тому, щоб розділити логіку програми на незалежні компоненти, що спрощує розробку, тестування та зміни програмного коду.

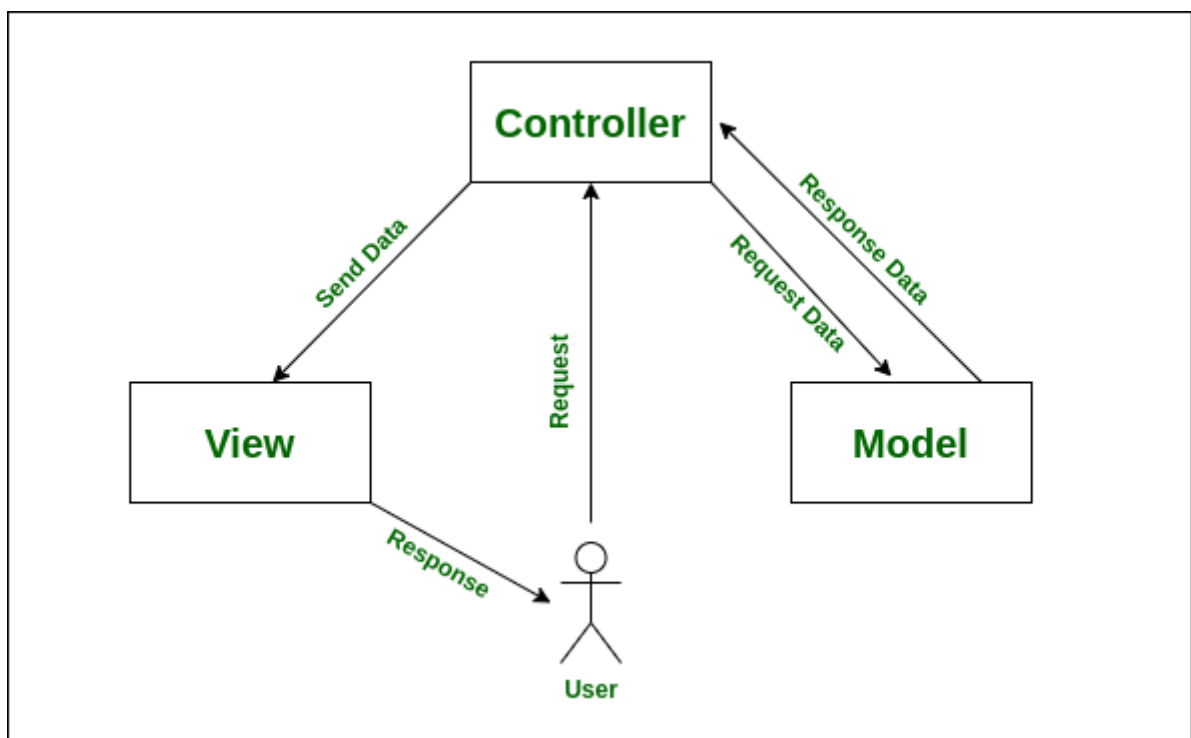


Рисунок 2.5 – MVC паттерн

Крім того, щоб підвищити модульність програми, було вирішено використати Headless CMS (Content Management System). Це тип системи управління контентом, яка відрізняється від традиційних CMS тим, що не має

фронтенд-компоненту, пов'язаного з відображенням контенту на веб-сторінках. У звичайних CMS веб-сторінки зазвичай генеруються із використанням серверного рендерингу, де контент і дизайн злиті разом.

У Headless CMS контент та його управління цілком відокремлені від презентації. Це означає, що контент може бути збережений в CMS, а потім використаний на різних платформах, таких як веб-сайти, мобільні додатки, розширені реальності, чат-боти та інші.

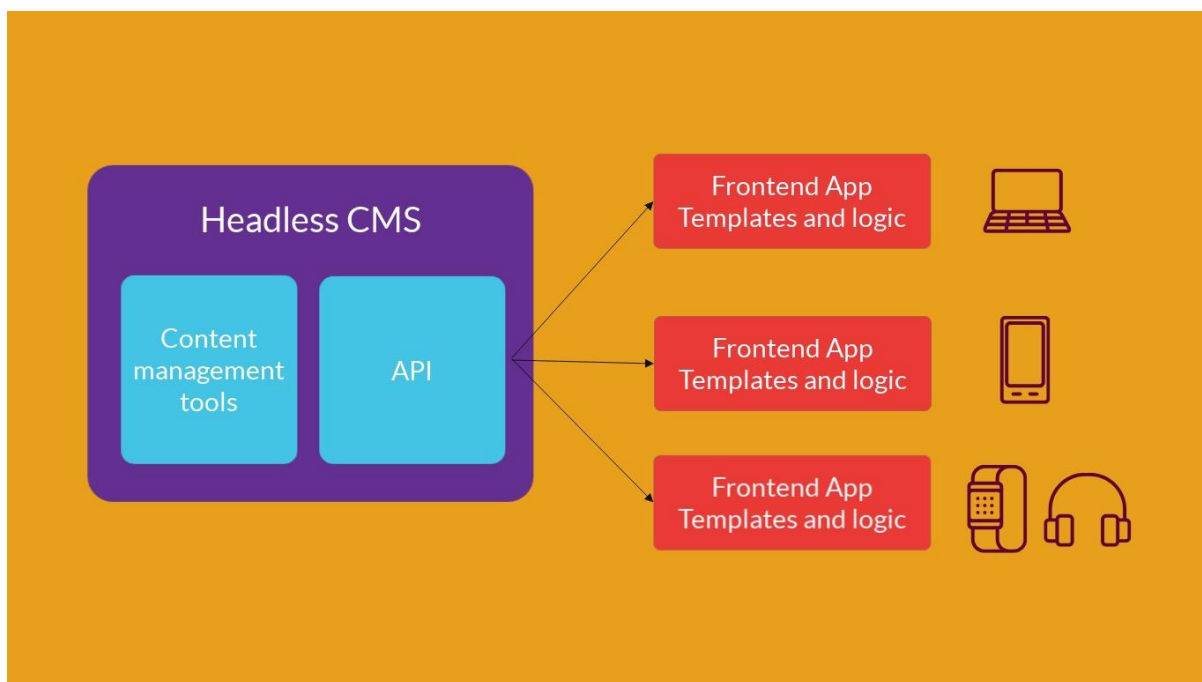


Рисунок 2.6 – Headless CMS

Основна ідея за Headless CMS полягає в тому, що контент використовується за допомогою API (Application Programming Interface). Це означає, що розробники можуть звертатися до CMS за допомогою API-запитів і отримувати контент у структурованому форматі, такому як JSON або XML. Цей контент потім можна відобразити на будь-якому каналі чи платформі, за бажанням розробника

Headless CMS надає більшу гнучкість і швидкість розробки, оскільки розробники можуть використовувати будь-які технології або фреймворки для

створення фронтенду. Вони можуть використовувати свої улюблені мови програмування, бібліотеки та інструменти, не залежно від CMS. Це також дозволяє розробникам швидко змінювати візуальний дизайн або змінювати вміст без впливу на зв'язані з цим канали доставки контенту.

2.3 Конструювання програмного забезпечення

2.3.1 Конструювання алгоритму створення розкладу

Алгоритми створення розкладу це складна тема що може мати багато підходів і рішень, різної степені ефективності. Для даного проекту було вирішено створити простий але ефективний алгоритм що працює методом перебору, ставлячи один урок за раз на найбільш оптимальне місце в розкладі. Протягом роботи, алгоритм звертає увагу на наступні фактори:

- День тижня: різні дні тижня мають різні рівні пріоритету для проведення занять. Так, алгоритм швидше поставить урок на понеділок-п'ятницю ніж на суботу.

- Час проведення заняття: різний час проведення занять має різний пріоритет для алгоритму;

- Заповнення “вікон”: алгоритм має більший пріоритет на розміщення занять безпосередньо перед або після інших занять, що дозволяє зменшити кількість “вікон” в розкладі.

Алгоритм працює по принципу підрахування “ваги” можливої позиції заняття. Для кожної позиції, алгоритм підраховує суму коефіцієнтів цих факторів і розміщує заняття на позицію з найбільшою такою сумою. Таким чином, ми отримуємо гнучкий алгоритм який можна легко модифікувати, додаючи власні фактори оцінення позиції – наприклад, пріоритет до розміщення певних занять в певні дні.

Також алгоритм підтримує і більш жорсткі умови – наприклад, викладач може бути доступним лише в певний час, і алгоритм буде розміщувати його заняття тільки в той час.

2.3.2 Конструювання коду програми

В якості мови програмування, було вибрано Java з використанням фреймворку Spring MVC. Основними причинами цього є:

- Розширена екосистема: Java має велику та активну спільноту розробників, що призводить до наявності розширеної екосистеми інструментів, бібліотек та фреймворків. Це робить його одним з найпопулярніших виборів для розробки веб-додатків.

- Об'єктно-орієнтована природа: Java є об'єктно-орієнтованою мовою програмування, що сприяє розробці модульного та підтримуваного коду. Це дозволяє створювати компоненти, які легко масштабувати, розширювати та підтримувати в майбутньому.

- Підтримка великих проектів: Java має довгу історію використання для розробки великих корпоративних проектів. Вона надає ряд функцій, таких як пакетування, модульність та механізми управління пам'яттю, що дозволяють ефективно керувати складністю проектів.

- Платформна незалежність: Однією з ключових переваг Java є його платформна незалежність. Програми, написані на Java, можуть працювати на різних операційних системах без необхідності вносити зміни в код. Це робить їх більш гнучкими та переносимими.

Основні класи проекту можна побачити серед графічних матеріалів цієї роботи, на кресленні “Структурна схема класів програмного забезпечення”. Опис цих класів наведено в таблиці 2.1.

Таблиця 2.1 – Основні класи проекту

Клас/Інтерфейс	Рівень	Опис
CourseRepository	Data Access Layer	Інтерфейс для роботи з сутністю курсів бази даних
GroupRepository	Data Access Layer	Інтерфейс для роботи з сутністю груп бази даних
LessonRepository	Data Access Layer	Інтерфейс для роботи з сутністю уроків бази даних
UserRepository	Data Access Layer	Інтерфейс для роботи з сутністю користувачів бази даних
EventRepository	Data Access Layer	Інтерфейс для роботи з сутністю подій бази даних
SlotRepository	Data Access Layer	Інтерфейс для роботи з сутністю слотів бази даних
ScheduleService	Business Layer	Клас для реалізації логіки побудови розкладу уроків і подій
UserService	Business Layer	Клас для реалізації логіки взаємодії з користувачами
CreatorService	Business Layer	Клас для реалізації логіки редагування розкладу, включаючи алгоритм автозаповнення.
ExcelService	Business Layer	Клас для реалізації логіки імпорту і експорту excel-файлів з розкладом
EventService	Business Layer	Клас для реалізації логіки взаємодії з подіями
GroupService	Business Layer	Клас для реалізації логіки взаємодії з групами
MainController	Presentation Layer	Контроллер MVC для виконання базових функцій;
UserController	Presentation Layer	Контроллер MVC для роботи з користувачами
EventController	Presentation Layer	Контроллер MVC для роботи з подіями.
CreatorController	Presentation Layer	Контроллер MVC для роботи з редактором розкладу

Продовження таблиці 2.1:

CourseController	Presentation Layer	Контроллер MVC для роботи з курсами.
ApiController	Presentation Layer	Контроллер MVC для зовнішнього API

Для розробки візуальної частини було вибрано HTML фреймворк bootstrap. Bootstrap - це потужний фреймворк для розробки веб-інтерфейсів, заснованих на мові розмітки HTML, CSS та JavaScript. Він надає набір готових компонентів, стилів і скриптів, які допомагають розробникам швидко та ефективно створювати сучасні та адаптивні веб-додатки.

Основні переваги використання фреймворку Bootstrap включають:

– Адаптивний дизайн: Bootstrap надає гнучкі та готові до використання класи CSS, які дозволяють легко створювати адаптивний дизайн. Це означає, що ваші веб-сторінки будуть відображатися на різних пристроях та екранах розмірів правильно і коректно.

– Готові компоненти: Bootstrap надає широкий набір готових компонентів, таких як кнопки, форми, меню, каруселі, модальні вікна та багато інших. Це дозволяє розробникам ефективно використовувати ці компоненти без необхідності вручну писати весь код з нуля.

– Стилзація: Bootstrap має набір стилів CSS, які допомагають швидко та легко стилізувати ваші веб-елементи. Ви можете застосовувати класи до елементів HTML, щоб надати їм різні стилі, такі як розташування, кольори, розміри, та інші.

– Перехідні ефекти та анімації: Bootstrap надає можливості для додавання перехідних ефектів та анімацій до вашого веб-інтерфейсу. Це допомагає зробити ваші сторінки більш привабливими та динамічними.

– Кросбраузерна сумісність: Bootstrap гарантує сумісність з різними веб-браузерами, що дозволяє вашим веб-додаткам працювати на різних платформах без проблем.

Узагальнюючи, Bootstrap - це потужний фреймворк для розробки веб-інтерфейсів на базі HTML, який спрощує роботу розробників, надаючи готові компоненти, адаптивний дизайн, стилізацію та інші можливості для швидкої та ефективної розробки сучасних веб-додатків.

2.3.3 Конструювання Баз Даних

В якості бази даних для цієї роботи було вибрано базу даних MySQL. Використання бази даних MySQL в даному проєкті має кілька переваг:

– Надійність та стабільність: MySQL є однією з найпоширеніших та найстабільніших реляційних баз даних. Вона має довгу історію розвитку, широке співтовариство користувачів та активну підтримку. Це забезпечує надійність та стабільність роботи бази даних.

– Широкі можливості: MySQL надає багато можливостей для роботи з даними. Вона підтримує широкий спектр типів даних, індексів, обмежень цілісності та операцій над даними. Багато різноманітних запитів та операцій можна виконати безпосередньо на рівні бази даних.

– Швидкодія: MySQL володіє доброю швидкодією та продуктивністю. Вона оптимізована для роботи з великими обсягами даних та високою навантаженістю. Індексування, кешування та інші оптимізації допомагають забезпечити ефективне виконання запитів.

– Масштабованість: MySQL має можливості для масштабування. Вона підтримує розподілені системи баз даних та реплікацію для підвищення

продуктивності та надійності. Це дозволяє розширювати базу даних зростанням обсягів даних та кількості користувачів.

– Інтеграція зі Spring: MySQL добре інтегрується з Java-фреймворком Spring. Spring Data JPA надає зручний спосіб взаємодії з базою даних MySQL, спрощує розробку і зменшує кількість коду. Крім того, Spring Framework надає інструменти для керування транзакціями, кешуванням та іншими аспектами роботи з базою даних.

Загалом, використання бази даних MySQL дозволяє забезпечити надійне зберігання та управління даними в проєкті, забезпечує широкі можливості роботи з даними, швидкодію та масштабованість. Інтеграція з Java Spring спрощує розробку та забезпечує зручний доступ до бази даних.

Структура створеної бази даних знаходиться в графічних матеріалах дипломного проєкту, креслення “Схема Бази Даних”. В таблиці 2.2 наведений опис таблиць цієї бази даних.

Таблиця 2.2 – Таблиці бази даних

Таблиця	Опис
lesson	Зберігає інформацію про статичні уроки/пари;
course	Зберігає інформацію про курси.предмети;
users	Зберігає інформацію про користувачів; поле type розрізняє викладачів і студентів;
groups	Зберігає інформацію про групи студентів;
event	Зберігає інформацію про події;
slots	Зберігає інформацію про слоти (комбінації дня і часу, в які можуть проводитися уроки)
users_events	Допоміжна таблиця для реалізації “Many-to-Many” зв’язку між користувачами і подіями;

Продовження таблиці 2.2:

Groups_lessons	Допоміжна таблиця для реалізації “Many-to-Many” зв’язку між групами і уроками;
Users_slots	Допоміжна таблиця для реалізації “Many-to-Many” зв’язку між користувачами і слотами. Цей зв’язок показує слоти в які викладачі не можуть проводити пари.

В таблиці 2.3 наведено утіліти використані під час розробки програмного коду.

Таблиця 2.3 – використані утіліти

№ п/п	Назва утиліти	Опис застосування
1	Apache NetBeans	Головне середовище розробки програмного забезпечення
2	Swagger	Програмне забезпечення необхідне для тестування rest запитів. Використовувалось для тестування API інтерфейсів, та клієнтських запитів.
3	XAMPP	Програмне забезпечення яке надає легкий графічний інтерфейс для доступу до бази даних.

2.4 Аналіз безпеки даних

Використання багатошарової архітектури Java Spring MVC може забезпечувати деякі важливі механізми для забезпечення безпеки даних у веб-додатках. Ось декілька аспектів безпеки, які можуть бути забезпечені цією архітектурою:

- редагування користувачем події;
- Розділення логіки: Багатошарова архітектура розділяє логіку додатку на окремі шари, такі як презентаційний, бізнес-логіка та рівень доступу до даних. Це дозволяє розмістити логіку безпеки відповідно до потреб проекту. Так, в застосунку присутні фільтри безпеки на презентаційному рівні для перевірки автентифікації та авторизації користувачів перед доступом до захищених ресурсів.

- Використання інструментів безпеки Spring: Фреймворк Spring надає ряд інструментів і бібліотек, що спрощують реалізацію безпеки в додатку. Наприклад, можна використовувати Spring Security для автентифікації, авторизації та захисту ресурсів. Spring Security надає розширені можливості, такі як обробка ролей, налаштування правил доступу та потенційна інтеграція з різними механізмами аутентифікації.

- Захист від атак на рівні бази даних: Використання багатошарової архітектури дозволяє розділити рівень доступу до даних, що дозволяє застосувати механізми захисту від атак на рівні бази даних. Наприклад, можна використовувати параметризовані запити (prepared statements) та ORM-фреймворки, такі як Hibernate, для запобігання SQL-ін'єкціям.

- Обробка валідації та перевірки даних: Забезпечення безпеки даних також включає валідацію та перевірку даних, які вводяться користувачами. Spring MVC надає можливості валідації даних на рівні презентації за допомогою анотацій, таких як @Valid, які дозволяють перевіряти коректність введених даних та запобігати вразливостям, таким як XSS або CSRF.

- Захист від перехоплення сесій: Spring Security надає засоби для запобігання атак на перехоплення сесій, таких як використання токенів (наприклад, JWT) та захищення передачі даних за допомогою шифрування.

Важливо пам'ятати, що безпека даних - це комплексне завдання, яке вимагає не лише використання правильних технологій та фреймворків, але й належної конфігурації, правильного використання інструментів та свідомого

підходу до проектування та розробки додатку. При використанні багатошарової архітектури Java Spring MVC з дотриманням відповідних практик безпеки, можна забезпечити надійний рівень захисту даних у веб-додатку.

Висновки до розділу

У розділі "Моделювання та конструювання програмного забезпечення" було проведено ретельне моделювання та аналіз програмного забезпечення, розроблено архітектуру програмного забезпечення, виконано конструювання програмного забезпечення та проведено аналіз безпеки даних. Цей розділ є ключовим для процесу розробки програмного продукту, оскільки в ньому визначається структура, функціональність та безпека програмного забезпечення.

Моделювання та аналіз програмного забезпечення здійснювалися з метою створення чіткого опису функціональності та взаємодії компонентів програмного продукту. Цей аналіз дозволив уточнити вимоги та встановити оптимальний спосіб організації програмного забезпечення. Крім того, він допоміг виявити потенційні проблеми та ризики, пов'язані з функціональністю та взаємодією компонентів системи.

Архітектура програмного забезпечення була розроблена з метою визначення структури та організації програмного продукту. Це включало в себе визначення компонентів системи, їх функціональних залежностей та способу взаємодії. Архітектура визначила загальний фреймворк для подальшої реалізації та розвитку програмного забезпечення.

Конструювання програмного забезпечення передбачало розробку конкретних компонентів та модулів, які входять до складу програмного продукту. Цей процес включав в себе програмування, тестування та інтеграцію компонентів з метою створення функціонально працездатного програмного забезпечення.

					КПІ.IT-9308.045440.02.81	Арк. 55
Змін.	Арк.	№ докум.	Підп.	Дата.		

Аналіз безпеки даних був важливою складовою розділу, оскільки забезпечення безпеки даних є критично важливим аспектом для багатьох програмних продуктів. В цьому контексті було проведено оцінку потенційних загроз та ризиків безпеки, розроблені заходи для захисту даних та визначені стратегії реагування на можливі інциденти.

У цілому, розділ "Моделювання та конструювання програмного забезпечення" надав фундаментальні знання та інструменти для розробки функціонального, структурованого та безпечного програмного забезпечення. Цей розділ є важливим етапом у процесі розробки, оскільки визначає основні принципи та архітектуру програмного продукту, що послужить фундаментом для подальшої реалізації та тестування програмного забезпечення.

					КПІ.IT-9308.045440.02.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		56

3. АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Аналіз якості ПЗ

Аналіз якості коду включає оцінку якісних характеристик програмного коду з метою виявлення можливих проблем, покращення ефективності, збільшення надійності та зручності обслуговування. Код програми було проаналізовано за наступними метриками:

– Кількість рядків коду (Lines of Code, LOC): Вказує на загальну кількість фізичних рядків коду. Висока кількість LOC може свідчити про більшу складність програми і потенційну складність обслуговування.

– Метрика Bugs: Ця метрика вказує на кількість виявлених помилок (багів) у програмному коді. Помилки можуть включати синтаксичні помилки, помилки логіки, некоректну обробку помилок та інші проблеми, які можуть впливати на коректність функціонування програми. Висока кількість багів може призвести до непередбачуваної поведінки, збоїв або незадовільної роботи програми.

– Метрика Vulnerability: Ця метрика виявляє потенційні вразливості програмного коду, які можуть бути використані зловмисниками для злому системи або отримання несанкціонованого доступу до даних. Вразливості можуть включати слабкі місця в захисті, недостатню перевірку введених даних, використання застарілих алгоритмів шифрування та інші проблеми, які можуть підірвати безпеку системи. Вимірювання цієї метрики допомагає виявити потенційні точки вразливості і вжити заходи для їх усунення.

– Метрика Security Hotspots: Ця метрика виявляє "гарячі точки" безпеки, тобто частини програмного коду, які мають великий потенціал для вразливостей і можуть бути пріоритетними для перевірки та вдосконалення. Враховуються різні фактори, такі як комплексність коду, наявність критичних даних, використання небезпечних функцій або API, недостатні заходи безпеки

тощо. Виявлення Security Hotspots дозволяє зосередити зусилля на найбільш значущих аспектах безпеки та запобігти можливим атакам чи пору

Ці метрики допомагають оцінити якість програмного коду Java, і їх використання може сприяти поліпшенню читабельності, ефективності та підтримки кодової бази.

На рисунку 3.1 можна побачити результати перевірки коду за допомогою системи SonarQube.

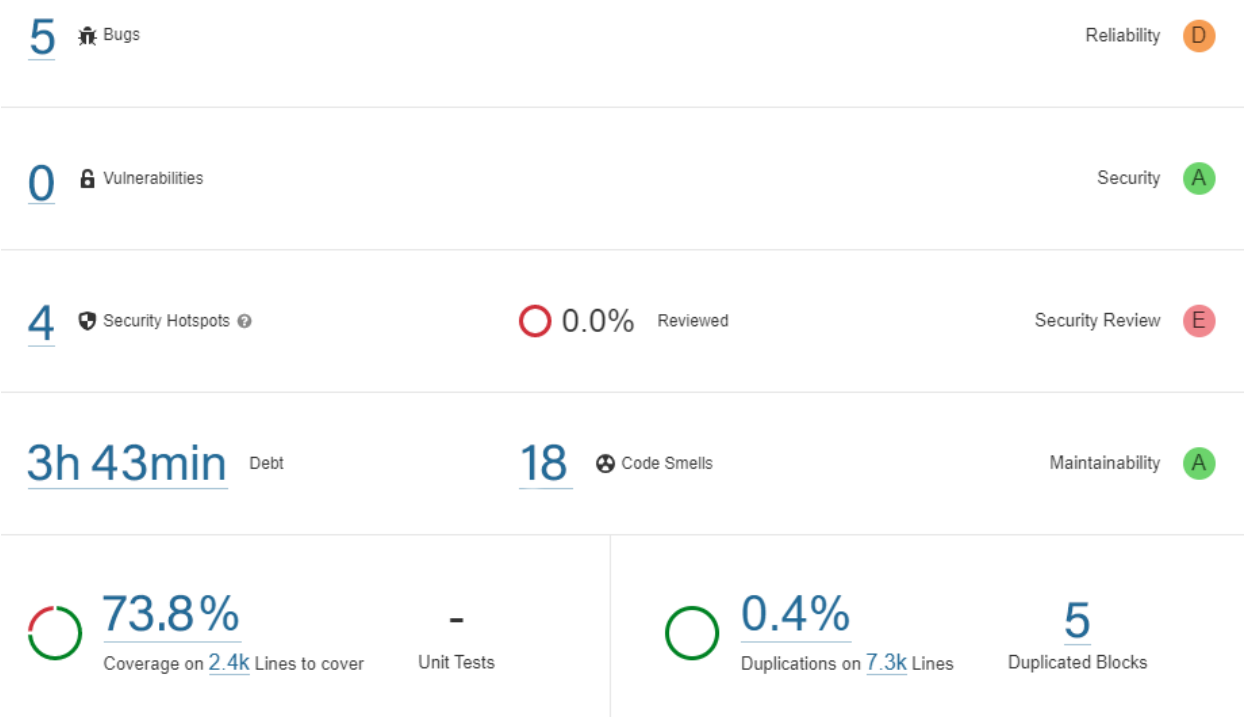


Рисунок 3.1 – Звіт аналізу коду SonarQube

3.2 Опис процесів тестування

Таблиця 3.1 – Тест 1.1

Тест	Авторизація користувача
Модуль	Авторизація користувача
Номер тесту	1.1
Початковий стан системи	Користувач знаходиться на сторінці авторизації
Вхідні данні	Електронна пошта, пароль

Продовження таблиці 3.1:

Опис проведення тесту	У відповідні поля вводяться коректна електронна пошта і пароль, після чого користувач натискає кнопку підтвердження.
Очікуваний результат	Авторизація проходить успішно, користувач зберігається в сесії і перенаправляється на основну сторінку.
Фактичний результат	Авторизація проходить успішно, користувач зберігається в сесії і перенаправляється на основну сторінку.

Таблиця 3.2 – Тест 1.2

Тест	Некоректна авторизація користувача
Модуль	Авторизація користувача
Номер тесту	1.2
Початковий стан системи	Користувач знаходиться на сторінці авторизації
Вхідні данні	Електронна пошта, пароль
Опис проведення тесту	У відповідні поля вводяться некоректна електронна пошта і пароль, після чого користувач натискає кнопку підтвердження.
Очікуваний результат	Вебсайт не підтверджує введені дані, залишаючи користувача на сторінці авторизації.
Фактичний результат	Вебсайт не підтверджує введені дані, залишаючи користувача на сторінці авторизації.

Таблиця 3.3 – Тест 2.1

Тест	Відображення розкладу
Модуль	Відображення і фільтрування розкладу
Номер тесту	2.1

Продовження таблиці 3.3:

Початковий стан системи	Користувач знаходиться на сторінці розкладу
Вхідні данні	-
Опис проведення тесту	Користувач вибирає необхідну йому групу і/або викладача в відповідних полях.
Очікуваний результат	Вебсайт відображає розклад всіх пар вибраної групи/викладача.
Фактичний результат	Вебсайт відображає розклад всіх пар вибраної групи/викладача.

Таблиця 3.4 – Тест 2.2

Тест	Відображення подій
Модуль	Відображення і фільтрування розкладу
Номер тесту	2.2
Початковий стан системи	Авторизований користувач знаходиться на сторінці розкладу
Вхідні данні	-
Опис проведення тесту	Користувач вибирає необхідний йому тиждень.
Очікуваний результат	В додаток до розкладу групи/викладача, вебсайт відображає всі події видимі для даного користувача і які відбувається в вибраний тиждень.
Фактичний результат	В додаток до розкладу групи/викладача, вебсайт відображає всі події видимі для даного користувача і які відбувається в вибраний тиждень.

Таблиця 3.5 – Тест 3.1

Тест	Створення подій
Модуль	Створення і редагування подій
Номер тесту	3.1
Початковий стан системи	Авторизований користувач знаходиться на сторінці створення події
Вхідні данні	Назва події, опис події, , дата події, і час.
Опис проведення тесту	Користувач вводить дані в відповідні поля і натискає кнопку підтвердження.
Очікуваний результат	Нова подія за авторством користувача створюється і додається до розкладу.
Фактичний результат	Нова подія за авторством користувача створюється і додається до розкладу.

Таблиця 3.6 – Тест 3.2

Тест	Редагування подій
Модуль	Створення і редагування подій
Номер тесту	3.2
Початковий стан системи	Авторизований користувач знаходиться на сторінці події створеної ним
Вхідні данні	Назва події, опис події, , дата події, і час.
Опис проведення тесту	Користувач змінює дані в відповідних полях і натискає кнопку підтвердження.
Очікуваний результат	Дані про подію змінюються.
Фактичний результат	Дані про подію змінюються.

Таблиця 3.7 – Тест 3.3

Тест	Некорректне редагування подій
Модуль	Створення і редагування подій
Номер тесту	3.3
Початковий стан системи	Авторизований користувач знаходиться на сторінці події не створеної ним
Вхідні данні	-
Опис проведення тесту	Користувач переходить на сторінку події не створеної ним в спробах її відредагувати.
Очікуваний результат	Система не надає користувачу можливості відредагувати подію.
Фактичний результат	Система не надає користувачу можливості відредагувати подію.

Таблиця 3.8 – Тест 4.1

Тест	Редагування курсів
Модуль	Редагування курсів
Номер тесту	4.1
Початковий стан системи	Авторизований користувач знаходиться на сторінці курсу для якого він є викладачем.
Вхідні данні	Назва і опис курсу
Опис проведення тесту	Користувач змінює дані в відповідних полях і натискає кнопку підтвердження.
Очікуваний результат	Дані про курс змінюються.
Фактичний результат	Дані про курс змінюються.

Таблиця 3.9 – Тест 4.2

Тест	Некоректне редагування курсів
Модуль	Редагування курсів
Номер тесту	4.2
Початковий стан системи	Авторизований користувач знаходиться на сторінці викладачем якого він не є.
Вхідні данні	-
Опис проведення тесту	Користувач переходить на сторінку курсу викладачем якого він не є в спробах його відредагувати.
Очікуваний результат	Система не надає користувачу можливості відредагувати курс.
Фактичний результат	Система не надає користувачу можливості відредагувати курс.

Таблиця 3.10 – Тест 5.1

Тест	Ручне редагування розкладу адміністратором.
Модуль	Редагування розкладу адміністратором.
Номер тесту	5.1
Початковий стан системи	Адміністратор знаходиться на сторінці редагування розкладу.
Вхідні данні	-
Опис проведення тесту	Адміністратор, знаходячись в редакторі, змінює позиції різних уроків/пар. При бажанні, натискаючи відповідні кнопки, він може скинути позиції всіх предметів або повернути розклад до попереднього стану. Закінчивши зміни, Адміністратор натискає кнопку збереження.
Очікуваний результат	Розклад в застосунку змінюється згідно з дій Адміністратора.

Продовження таблиці 5.1:

Фактичний результат	Розклад в застосунку змінюється згідно з дій Адміністратора.
---------------------	--

Таблиця 3.11 – Тест 5.2

Тест	Некорректне редагування розкладу адміністратором.
Модуль	Редагування розкладу адміністратором.
Номер тесту	5.2
Початковий стан системи	Адміністратор знаходиться на сторінці редагування розкладу.
Вхідні данні	-
Опис проведення тесту	Адміністратор, знаходячись в редакторі, намагається перемістити урок на позицію для нього не валідну.
Очікуваний результат	Адміністратор не може виконати цю дію.
Фактичний результат	Адміністратор не може виконати цю дію.

Таблиця 3.12 – Тест 5.3

Тест	Автоматичне генерування розкладу адміністратором.
Модуль	Редагування розкладу адміністратором.
Номер тесту	5.3
Початковий стан системи	Адміністратор знаходиться на сторінці редагування розкладу.
Вхідні данні	-
Опис проведення тесту	Адміністратор натискає кнопку автоматичного розподілення уроків.

Продовження таблиці 3.12:

Очікуваний результат	Всі уроки що не є в даний момент назначені в певні слоти автоматично розподіляються по оптимальним позиціям.
Фактичний результат	Всі уроки що не є в даний момент назначені в певні слоти автоматично розподіляються по оптимальним позиціям.

Таблиця 3.13 – Тест 6.1

Тест	Завантаження розкладу адміністратором.
Модуль	Завантаження і вивантаження розкладу адміністратором.
Номер тесту	6.1
Початковий стан системи	Адміністратор знаходиться на сторінці редагування розкладу.
Вхідні данні	-
Опис проведення тесту	Адміністратор натискає кнопку завантаження розкладу.
Очікуваний результат	Excel-файл з інформацією про розклад з редактору завантажується на комп'ютер адміністратора.
Фактичний результат	Excel-файл з інформацією про розклад з редактору завантажується на комп'ютер адміністратора.

Таблиця 3.14 – Тест 6.2

Тест	Вивантаження розкладу адміністратором.
Модуль	Завантаження і вивантаження розкладу адміністратором.
Номер тесту	6.2
Початковий стан системи	Адміністратор знаходиться на сторінці вивантаження розкладу.
Вхідні данні	-

Продовження таблиці 3.14:

Опис проведення тесту	Адміністратор вибирає excel-файл з розкладом зі свого комп'ютера, потім натискає кнопку підтвердження.
Очікуваний результат	Розклад з файлу вивантажується в редактор розкладу.
Фактичний результат	Розклад з файлу вивантажується в редактор розкладу.

Таблиця 3.14 – Результати тестування

1.1	Авторизація користувача	Пройдено
1.2	Некорректна авторизація користувача	Пройдено
2.1	Відображення розкладу	Пройдено
2.2	Відображення подій	Пройдено
3.1	Створення подій	Пройдено
3.2	Редагування подій	Пройдено
3.3	Некорректне редагування подій	Пройдено
4.1	Редагування курсів	Пройдено
4.2	Некорректне редагування курсів	Пройдено
5.1	Ручне редагування розкладу адміністратором.	Пройдено
5.2	Некорректне редагування розкладу адміністратором.	Пройдено
5.3	Автоматичне генерування розкладу адміністратором.	Пройдено
6.1	Завантаження розкладу адміністратором.	Пройдено
6.2	Вивантаження розкладу адміністратором.	Пройдено

3.3 Опис контрольного прикладу

Було проведено тестування функціоналу пов'язаного з переглядом і створенням розкладів з врахуванням усіх можливих розгалуджень. Першим кроком було відкрито головну сторінку застосунку (рисунок 3.1) та обрано певну групу для перегляду її розкладу. Наступним кроком було натискання кнопки авторизації і введення даних користувача (рисунок 3.2). Після цього вебсайт було перенаправлено назад на головну сторінку, тепер з наявним функціоналом для користувача (рисунок 3.3). Наступним кроком було натискання кнопки створення події і ввід необхідних даних (рисунок 3.4), після чого цю подію можна побачити в розкладі. Далі було натиснуто на один з предметів і переглянуто інформацію про нього (рисунок 3.5). Після повернення на головну сторінку, було натиснуто на кнопку виходу з аккаунту, після чого було залогінено використовуючи дані адміністратора. Далі, натиснувши відповідну кнопку, було перейдено на сторінку редагування розкладу (рисунок 3.6). Спочатку було натиснуто кнопку завантаження поточного розкладу в вигляді excel-файлу. Потім було проведено ряд змін в розкладі, включаючи можливість генерації нового, і всі зміни було збережено – в цьому було переконано після перевірки головної сторінки (рисунок 3.7). Після цього в редактор було вивантажено попередньо закачаний файл з розкладом, збережено зміни, і розклад повернувся до попереднього стану (рисунок 3.8).

Teachers:

 Groups:

IT1

Prev (2023-06-05)

 Week: 2023-06-12 - 2023-06-17

Next (2023-06-19)

Log In

#	Пн	Вт	Ср	Чт	Пт	Сб
8:45		Lecture Math Mikola Tkachenko		Lecture Math Mikola Tkachenko		
9:30	Lab Math Lubov Koval	Lab Physics Stepan Makar	Lecture Algorithms Lubov Koval	Lecture Physics Stepan Makar	Lab Physics Stepan Makar	
10:15	Lab Math Lubov Koval	Lab Programming Oksana Kolosok		Lab English Maxim Mohylo	Lecture Physics Stepan Makar	
11:00	Lab Security Oksana Kolosok		Lab Algorithms Lubov Koval	Lab English Maxim Mohylo	Lab Programming Oksana Kolosok	
11:45		Lecture Programming Oleh Mityk			Lecture Security Oksana Kolosok	
12:30						

Рисунок 3.1 – Головна сторінка застосунку

Авторизація

anton@sm.sm

Email address

.....

Password

☒ Remember me

Forgot password?

Sign in

Рисунок 3.2 – Сторінка авторизації застосунку

Teachers:

 Groups:

ITI

Prev (2023-05-22)

Week: 2023-05-29 - 2023-06-03

Next (2023-06-05)

Log out

New Event

#	Пн	Вт	Ср	Чт	Пт	Сб
8:45		Lecture Math (asd) Mikola Tkachenko		Lecture Math (asd) Mikola Tkachenko		
9:30	Lab Math (asd) Lubov Koval	Lab Physics (asd) Stepan Makar	Lecture Algorithms (A great lesson) Lubov Koval	Lecture Physics (asd) Stepan Makar	Lab Physics (asd) Stepan Makar	
10:15	Lab Math (asd) Lubov Koval	Lab Programming (asd) Oksana Kolosok		Lab English (asd) Maxim Mohylko event My Own Event (Cool) Anton Grigoryev	Lecture Physics (asd) Stepan Makar	
11:00	Lab Security (asd) Oksana Kolosok		Lab Algorithms (A great lesson) Lubov Koval	Lab English (asd) Maxim Mohylko	Lab Programming (asd) Oksana Kolosok	
11:45		Lecture Programming (asd) Oleh Mityk			Lecture Security (asd) Oksana Kolosok	
12:30						

Рисунок 3.3 – Головна сторінка застосунку після авторизації

[Back to Schedule](#)

Name

NewEvent

Description

Something Different

Date

16.06.2023



Pair

5

Submit

Рисунок 3.4 – Сторінка створення нової події

[Back to Schedule](#)

Name

Algorythms

Description

A great lesson

Рисунок 3.5 – Сторінка перегляду інформації про курс

[Back to main page](#)
[Save changes](#)

Teachers: ---
Groups: IT1

[Reset all](#)
[Assign all](#)
[Get current schedule](#)
[Download](#)
[Upload](#)

#	Пн	Вт	Ср	Чт	Пт	Сб
8:45	+	+ Lecture Math Mikola Tkachenko [IT1, IT2, IT3, IT4]	+	+ Lecture Math Mikola Tkachenko [IT1, IT2, IT3, IT4]	+	+
9:30	+ Lab Math Lubov Koval [IT1]	+ Lab Physics Stepan Makar [IT1]	+ Lecture Algorithms Lubov Koval [IT1, IT2, IT3, IT4]	+ Lecture Physics Stepan Makar [IT1, IT2]	+ Lab Physics Stepan Makar [IT1]	+
10:15	+ Lab Math Lubov Koval [IT1]	+ Lab Programming Oksana Kolosok [IT1]	+	+ Lab English Maxim Mohylo [IT1]	+ Lecture Physics Stepan Makar [IT1, IT2]	+
11:00	+ Lab Security Oksana Kolosok [IT1]	+	+ Lab Algorithms Lubov Koval [IT1]	+ Lab English Maxim Mohylo [IT1]	+	+ Lab Programming Oksana Kolosok [IT1]
11:45	+	+ Lecture Programming Oleh Mityk [IT1, IT2, IT3, IT4]	+	+	+	+ Lecture Security Oksana Kolosok [IT1, IT2, IT3, IT4]

Unassigned Lessons

+

Рисунок 3.6 – Редактор розкладів

Teachers: ---
Groups: IT1

[Prev \(2023-06-05\)](#)
Week: 2023-06-12 - 2023-06-17
[Next \(2023-06-19\)](#)

[Log out](#)
[Edit schedule](#)

#	Пн	Вт	Ср	Чт	Пт	Сб
8:45	Lab Math (asd) Lubov Koval					
9:30	Lab English (asd) Maxim Mohylo	Lab Physics (asd) Stepan Makar	Lab Algorithms (A great lesson) Lubov Koval	Lecture Programming (asd) Oleh Mityk	Lecture Math (asd) Mikola Tkachenko	
10:15	Lab English (asd) Maxim Mohylo	Lecture Security (asd) Oksana Kolosok	Lecture Physics (asd) Stepan Makar	Lecture Physics (asd) Stepan Makar		
11:00	Lab Programming (asd) Oksana Kolosok	Lab Physics (asd) Stepan Makar	Lab Security (asd) Oksana Kolosok	Lecture Math (asd) Mikola Tkachenko	Lab Math (asd) Lubov Koval	
11:45				Lab Programming (asd) Oksana Kolosok	Lecture Algorithms (A great lesson) Lubov Koval	
12:30						

Рисунок 3.7 – Змінений розклад

Teachers: ---
Groups: IT1

[Prev \(2023-06-05\)](#)
Week: 2023-06-12 - 2023-06-17
[Next \(2023-06-19\)](#)

[Log out](#)
[Edit schedule](#)

#	Пн	Вт	Ср	Чт	Пт	Сб
8:45		Lecture Math (asd) Mikola Tkachenko		Lecture Math (asd) Mikola Tkachenko		
9:30	Lab Math (asd) Lubov Koval	Lab Physics (asd) Stepan Makar	Lecture Algorithms (A great lesson) Lubov Koval	Lecture Physics (asd) Stepan Makar	Lab Physics (asd) Stepan Makar	
10:15	Lab Math (asd) Lubov Koval	Lab Programming (asd) Oksana Kolosok		Lab English (asd) Maxim Mohylo	Lecture Physics (asd) Stepan Makar	
11:00	Lab Security (asd) Oksana Kolosok		Lab Algorithms (A great lesson) Lubov Koval	Lab English (asd) Maxim Mohylo	Lab Programming (asd) Oksana Kolosok	
11:45		Lecture Programming (asd) Oleh Mityk			Lecture Security (asd) Oksana Kolosok	
12:30						

Рисунок 3.8 – Відновлений розклад

Висновки до розділу

У розділі "Аналіз Якості та тестування програмного забезпечення" було проведено детальний аналіз якості програмного забезпечення, описано процеси тестування та наведено опис контрольного прикладу. Цей розділ є ключовим для забезпечення якості та надійності програмного забезпечення, а також для виявлення та виправлення можливих дефектів.

Аналіз якості програмного забезпечення включав в себе оцінку різних аспектів, таких як функціональність, надійність, продуктивність, ефективність, безпека та користувальницький досвід. Цей аналіз дозволив визначити важливі критерії якості та установити метрики для оцінки відповідності програмного забезпечення цим критеріям.

Опис процесів тестування включав в себе опис різних типів тестування. Були визначені критерії прийнятності, тестові сценарії та методи тестування для кожного типу тестування. Цей аналіз допоміг забезпечити комплексне тестування програмного забезпечення з метою виявлення та усунення можливих дефектів.

Опис контрольного прикладу надав конкретний сценарій для тестування програмного забезпечення. Цей сценарій описував вхідні дані, очікувані результати та очікувані критерії якості для конкретного функціонального модуля або функції програмного забезпечення. Цей контрольний приклад служив основою для проведення конкретних тестів та оцінки відповідності програмного забезпечення визначеним критеріям якості.

Загалом, розділ "Аналіз Якості та тестування програмного забезпечення" допоміг у визначенні критеріїв якості, організації тестування та виявленні потенційних проблем програмного забезпечення. Цей розділ відіграє важливу роль у процесі забезпечення якості та надійності програмного продукту перед його впровадженням та використанням.

4. ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Розгортання програмного забезпечення

Процес розгортання веб-додатку, розробленого на мові програмування Java з використанням фреймворку Spring MVC і бази даних MySQL, можна поділити на кілька кроків:

- Підготовка серверного середовища: Переконайтеся, що сервер, на якому плануєте розгорнути додаток, налаштований для підтримки Java-програм та веб-додатків. Встановіть Java Runtime Environment (JRE) та веб-сервер, такий як Apache Tomcat або Jetty.

- Підготовка бази даних: Створіть базу даних MySQL та відповідні таблиці для зберігання даних вашого додатку. Забезпечте правильні налаштування доступу до бази даних, створивши користувача з необхідними привілеями.

- Налаштування проекту: У вашому проекті налаштуйте підключення до бази даних MySQL, використовуючи відповідні налаштування у файлі конфігурації (наприклад, у файлі `application.properties` або `application.yml`).

- Збірка проекту: Виконайте збірку вашого проекту за допомогою Maven або Gradle, щоб отримати готовий WAR-файл (Web Application Archive), який буде розгортатися на веб-сервері.

- Розгортання на сервері: Скопіюйте отриманий WAR-файл до каталогу розгортання веб-сервера. Запустіть веб-сервер та переконайтеся, що ваш додаток успішно розгортається.

- Тестування: Перевірте, що ваш додаток працює на сервері. Перевірте функціональність, взаємодію з базою даних та інші аспекти, щоб переконатися, що все працює правильно.

- Налаштування додаткових параметрів: Залежно від потреб вашого додатку, налаштуйте додаткові параметри, такі як рівень журналізації,

безпеку, зовнішні сервіси, електронну пошту тощо. Виконайте необхідні зміни у файлі конфігурації.

– Моніторинг та підтримка: Налаштуйте систему моніторингу та журналювання, щоб слідкувати за працездатністю та продуктивністю вашого додатку. Забезпечте регулярне оновлення додатку та вирішення виявлених проблем.

Процес розгортання може варіюватися залежно від обраного веб-сервера, контейнера Servlet, інструментів збірки та інших факторів. Впевніться, що ви дотримуєтеся рекомендацій відповідних документаційних матеріалів для використовуваних технологій та фреймворків.

4.2 Підтримка програмного забезпечення

Моніторинг та журналювання: Встановіть систему моніторингу, яка дозволяє відстежувати працездатність та продуктивність додатку. Це включає контроль навантаження на сервер, швидкість відгуку, використання ресурсів (пам'ять, процесор) та інші метрики. Також налаштуйте журналювання, щоб збирати журнали подій, помилок та винятків, що виникають у додатку.

– Перевірка та усунення помилок: Регулярно перевіряйте журнали та моніторинг, щоб виявляти помилки та проблеми. Аналізуйте журнали, спостерігайте за попередженнями та помилками, які можуть виникати. Відстежуйте звіти про винятки та реагуйте на них, виправляючи та видаляючи причини помилок.

– Резервне копіювання та відновлення: Забезпечте регулярне резервне копіювання бази даних та конфігураційного коду додатку. Це дозволить відновити систему у разі виникнення непередбачуваних проблем або втрати даних. Перевірте, чи резервні копії коректно створюються та можуть бути відновлені.

– Моніторинг безпеки: Проводьте регулярну перевірку на предмет потенційних вразливостей та забезпечуйте безпеку додатку. Використовуйте сканери вразливостей, проводьте аудит безпеки коду та налаштувань середовища для виявлення та усунення проблем безпеки.

Процес підтримки додатку є постійним та динамічним, вимагає систематичного підходу та уваги до деталей. Необхідно забезпечити безперебійну роботу додатку, а також вчасно реагувати на зміни вимог користувачів та бізнес-логіки.

					КПІ.IT-9308.045440.02.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		74

ВИСНОВКИ

У ході проведення дипломного проекту був розроблений веб-застосунок для організації навчання в дистанційному режимі з використанням мови програмування Java та фреймворку Spring MVC. В процесі роботи були виконані наступні кроки:

– Проведено аналіз існуючих програмних продуктів, що займаються організацією навчання в дистанційному режимі. Це дозволило визначити їх переваги та недоліки, а також виявити можливості для покращення та унікальні особливості, які можна внести у розробку нашого веб-застосунку.

– Сформульовані вимоги до програмного забезпечення, враховуючи потреби користувачів та бізнес-вимоги. Вимоги були розбиті на функціональні та нефункціональні, що дало змогу чітко визначити очікувані функції та властивості застосунку.

– Розроблений веб-застосунок мовою програмування Java з використанням фреймворку Spring MVC. Була використана багатоваршова архітектура, що дозволила розділити логіку застосунку на окремі компоненти та забезпечити їх взаємодію. Використання Spring MVC спростило розробку контролерів, моделей та представлень, забезпечивши швидку і просту реалізацію веб-інтерфейсу.

– Проведено тестування розробленого програмного забезпечення з метою виявлення помилок та перевірки відповідності вимогам.

На основі проведеної роботи можна зробити такі висновки:

– Веб-застосунок, розроблений з використанням мови програмування Java та фреймворку Spring MVC, дозволяє організувати навчання в дистанційному режимі ефективно та зручно. Використання Java і Spring MVC надає потужність та гнучкість для розробки та розширення функціоналу застосунку.

– Багатошарова архітектура дозволяє розділити логіку застосунку на окремі компоненти, що сприяє підтримці коду, його розширенню та змінам без зайвого впливу на інші компоненти.

– Підтримка і супровід розробленого застосунку є важливим етапом після завершення розробки. Це включає забезпечення безперебійної роботи системи, виявлення та усунення помилок, а також розширення функціоналу відповідно до змінних потреб користувачів.

У цілому, розробка веб-застосунку для організації навчання в дистанційному режимі з використанням мови програмування Java та фреймворку Spring MVC є успішним проектом, який дозволяє ефективно використовувати технології для поліпшення процесу навчання та забезпечення задоволення потреб користувачів.

					КПІ.IT-9308.045440.02.81	Арк. 76
Змін.	Арк.	№ докум.	Підп.	Дата.		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1) Ruth Colvin Clark. e-Learning and the Science of Instruction / ed. by Richard E. Mayer. 2016.
- 2) Richard N. Taylor. Software Architecture: Foundations, Theory, and Practice. 2009.
- 3) Michael L. Pinedo. Scheduling Theory, Algorithms, and Systems. 2016.
- 4) Spring Framework Documentation :: Spring Framework. *Spring | Home*. URL: <https://docs.spring.io/spring-framework/reference/>.
- 5) Hibernate ORM 6.2.5.Final User Guide. *docs.jboss.org*. URL: https://docs.jboss.org/hibernate/orm/current/userguide/html_single/Hibernate_User_Guide.html.
- 6) Welcome to Apache NetBeans. *Welcome to Apache NetBeans*. URL: <https://netbeans.apache.org/>.

ДОДАТКИ

ДОДАТОК А ЗВІТ ПОДІБНОСТІ



Ім'я користувача:
Лісовиченко Олег Іванович

Дата перевірки:
03.06.2023 07:53:39 EEST

Дата звіту:
03.06.2023 07:56:10 EEST

ID перевірки:
1015403976

Тип перевірки:
Doc vs Internet + Library

ID користувача:
76913

Назва документа: IT-93_Григор'єв_ПЗ

Кількість сторінок: 61 Кількість слів: 9892 Кількість символів: 75133 Розмір файлу: 1.15 MB ID файлу: 1015067682

100 слів позначені як "вилучені" та не враховуються у підрахунку слів

14.8%
Схожість

Найбільша схожість: 8.92% з джерелом з Бібліотеки (ID файлу: 1015058496)

1.6% Джерела з Інтернету

94

Сторінка 63

14.4% Джерела з Бібліотеки

252

Сторінка 64

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Деякі джерела вилучено автоматично (фільтри вилучення: кількість знайдених слів є меншою за 8 слів та 0%)

Немає вилучених Інтернет-джерел

0% Вилученого тексту з Бібліотеки

1

Сторінка 64

Змін.	Арк.	№ докум.	Підп.	Дата.

КПІ.IT-9308.045440.02.81

Арк.

78

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

_____ Едуард ЖАРІКОВ

“ ____ ” _____ 2023 р.

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ОРГАНІЗАЦІЇ НАВЧАННЯ В
ДИСТАНЦІЙНОМУ РЕЖИМІ**

Текст програми

КПІ.IT-9308.045440.03.12

“ПОГОДЖЕНО”

Керівник проєкту:

Константин НОСОВ

Нормоконтроль:

Максим ГОЛОВЧЕНКО

Виконавець:

Антон ГРИГОР'ЄВ

Київ – 2023

Файл CreatorService.java:

```
@Service
public class CreatorService {

    @Autowired
    private GroupRepository groupRepository;

    @Autowired
    private LessonRepository lessonRepository;

    @Autowired
    private SlotRepository slotRepository;

    public List<Lesson> FullReset(List<Lesson> lessons)
    {
        for (Lesson l:lessons)
        {
            l.setSlot(null);
        }
        return lessons;
    }

    public List <Lesson> AssignLessons(List<Lesson>lessons)
    {
        List <Lesson> unassigned = lessons.stream().filter(l ->
l.getDay()==0||l.getNumber()==0).collect(Collectors.toList());
        Collections.shuffle(unassigned);
        ListIterator <Lesson> i = unassigned.listIterator();
        while (i.hasNext())
        {
            Lesson l = i.next();
            Slot f = null;
            Integer a=0;
            List <Slot> slots = slotRepository.findAll();
            Collections.shuffle(slots);
            for (Slot s:slots)
            {
                if (CheckLesson(l, s.getDay(), s.getNumber(), lessons) == 0 && CheckTeacher(s, l))
                {
                    Integer a1 = s.getPriority();
                    a1+=CheckLesson(l, l.getDay(), l.getNumber()+1, lessons);
                    a1+=CheckLesson(l, l.getDay(), l.getNumber()-1, lessons);
                    if (a1>a)
                    {
                        f=s;
                        a=a1;
                    }
                }
            }
            l.setSlot(f);
        }
        return lessons;
    }

    private boolean CheckTeacher(Slot s, Lesson l)
    {
        for (User t: s.getUsers())
```

					КПІ.ІТ-9308.045440.03.12	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		2

```

        {
            if (l.getTeacher().getId().equals(t.getId())) return false;
        }
        return true;
    }

    private Integer CheckLesson(Lesson ls, Integer d, Integer n, List<Lesson> lessons)
    {
        List <Lesson> test = lessons.stream().filter((l) -> l.getDay().equals(d) &&
l.getNumber().equals(n)).collect(Collectors.toList());
        Integer res = 0;
        for (Lesson l:test)
        {
            if (l.getTeacher()==ls.getTeacher())
            {

                res++;
            }
            for (Group g1: l.getGroups())
            {
                for (Group g2: ls.getGroups())
                {

                    if (g1.getName().equals(g2.getName()))
                    {
                        res++;
                        break;
                    }
                }
            }
        }
        return res;
    }

    public void SaveAll(List<Lesson> lessons) {
        lessonRepository.saveAll(lessons);
    }

    public List<Lesson> ProcessSingle(List<SlotData> slotslist, List<Lesson> lessons) {
        for (SlotData item : slotslist) {
            Optional<Slot> slotOpt = slotRepository.findSlotInfo(item.getDay(), item.getNumber());

            Slot slot = null;
            if (slotOpt.isPresent()) {
                slot = slotOpt.get();
            }
            for (Lesson l:lessons)
            {
                if (item.getId().equals(l.getId()))
                {
                    l.setSlot(slot);
                }
            }
        }
        return lessons;
    }

    public List<DayDisplay> GetWeekLessons(Integer teacher_id, Integer group_id, List<Lesson>lessons) {
        //List<Lesson> lessons = GetAllLessons();
        if (teacher_id > 0) {
            lessons = lessons.stream().filter((l) -> l.getTeacher().getId().equals(teacher_id)).collect(Collectors.toList());
        }
        if (group_id > 0) {

```

					КПІІ.IT-9308.045440.03.12	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		3

```

        Group group = groupRepository.getReferenceById(group_id);
        lessons = lessons.stream().filter((l) -> ContainsGroup(l.getGroups(), group)).collect(Collectors.toList());

    }
    List<DayDisplay> week = new LinkedList<>();

    for (int i = 0; i < 7; i++) {
        int day = i;
        List<PairDisplay> daypairs = new ArrayList<>();
        for (int j = 0; j < 7; j++) {
            int number = j;
            List<Lesson> pairlessons = lessons.stream().filter((p) -> p.getDay() == day && p.getNumber() ==
number).collect(Collectors.toList());
            if (!pairlessons.isEmpty()) {
                PairDisplay pair = new PairDisplay(number, paircomments.get(number),
prepareLessinsDisplay(pairlessons));
                daypairs.add(pair);
            }
        }
        DayDisplay dayDisplay = new DayDisplay(weekdays.get(day), day, daypairs);
        week.add(dayDisplay);
    }

    return week;
}

private boolean ContainsGroup(List<Group> groups, Group group)
{

    for (Group g:groups)
    {
        if (g.getName().equals(group.getName())) return true;
    }
    return false;
}

private List<EventDisplay> prepareLessinsDisplay(List<Lesson> list) {
    List<EventDisplay> items = new LinkedList<>();
    for (Lesson lesson : list) {
        items.add(new EventDisplay(lesson));
    }
    return items;
}

public HashMap<String, List<EventDisplay>>> PrepareLessonsToHash(List<DayDisplay> lessons) {
    HashMap<String, List<EventDisplay>>> list = new HashMap<>();
    for (DayDisplay day : lessons) {
        for (PairDisplay pair : day.getDaylessons()) {
            list.put(day.getDay() + "_" + pair.getNumber(), pair.getLessons());
        }
    }
    return list;
}

public List<Lesson> GetAllLessons()
{
    return lessonRepository.findAll();
}

}

```

Файл EventService.java:

```
@Service
public class EventService {

    @Autowired
    private EventRepository eventRepository;

    @Autowired
    private UserRepository userRepository;

    @Autowired
    private CourseRepository courseRepository;

    @Autowired
    private LessonRepository lessonRepository;

    public Integer CcreateEvent(String name, String description, Date date, List<Group> groups) {
        Event event = new Event();
        event.setName(name);
        event.setDescription(description);
        event.setDay(date);
        event.setGroups(groups);
        return eventRepository.saveAndFlush(event).getId();
    }

    public List<Event> GetAllEvents() {
        return eventRepository.findAll();
    }

    public Event GetEvent(Integer id) {
        return eventRepository.findById(id).orElse(null);
    }

    public void DeleteEvent(Integer id) {
        eventRepository.deleteById(id);
    }

    public Event UpdateEvent(EventForm form) {
        Event event;
        if (form.getId() == null || form.getId()==0) {
            event = new Event();
            event.setUser(userRepository.getById(3));
        } else {
            event = eventRepository.findById(form.getId()).orElse(new Event());
        }

        event.setName(form.getName());
        event.setDescription(form.getDescription());
        try {
            Date date1 = new SimpleDateFormat("yyyy-MM-dd").parse(form.getDate());
            event.setDay(date1);
        } catch (ParseException ignore) {
        }
        event.setNumber(form.getNumber());
        event.setGroups(null);

        return eventRepository.saveAndFlush(event);
    }

    public Course GetCourseByLesson(Integer id)
    {
    }
```

					КПІ.ІТ-9308.045440.03.12	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		5

```

        return lessonRepository.getById(id).getCourse();
    }

    public Course GetCourse(Integer id)
    {
        return courseRepository.getById(id);
    }

    public Course UpdateCourse(CourseForm form) {
        Course course;
        course = courseRepository.findById(form.getId()).orElse(new Course());

        course.setName(form.getName());
        course.setDescription(form.getDescription());

        return courseRepository.saveAndFlush(course);
    }
}

```

Файл ExcelService.java:

```

@Service
public class ExcelService {

    @Autowired
    SlotRepository slotRepository;

    @Autowired
    LessonRepository lessonRepository;

    public XSSFWorkbook GetScheduler(HashMap<String, List<Lesson>> displaylist) {
        XSSFWorkbook workbook = new XSSFWorkbook();
        XSSFSheet sheet = workbook.createSheet("Scheduler");

        CreateHeader(workbook, sheet);

        CellStyle headerstyle = createRowStyle(workbook);
        CellStyle style = createRowStyle(workbook);

        for (int r : ScheduleService.paircomments.keySet()) {

            Row row = sheet.createRow(r);
            Cell cell0 = row.createCell(0, CellType.STRING);
            cell0.setCellStyle(headerstyle);
            cell0.setCellValue(ScheduleService.paircomments.get(r));
            row.setHeightInPoints((float) 70);

            for (Integer coll : ScheduleService.weekdays.keySet()) {

                String key = coll + "_" + r;
                String value = "";
                if (displaylist.containsKey(key)) {
                    for (Lesson lesson : displaylist.get(key)) {
                        value += String.format("%d;%s;%s;%s;%s\n", lesson.getId(), lesson.getType(),
                            lesson.getCourse().getName(), lesson.getTeacher(), lesson.getGroups());
                    }
                }
                Cell cell = row.createCell(coll, CellType.STRING);
                cell.setCellStyle(style);
                cell.setCellValue(value.trim());
            }
        }
    }
}

```

					КПІ.ІТ-9308.045440.03.12	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		6

```

    }
}

return workbook;
}

private void CreateHeader(XSSFWorkbook workbook, XSSFSheet sheet) {

    CellStyle style = createHeadingStyle(workbook);

    Row row = sheet.createRow(0);
    Cell cell0 = row.createCell(0, CellType.STRING);
    cell0.setCellValue("#");
    sheet.setColumnWidth(0, 1500);

    for (Integer coll : ScheduleService.weekdays.keySet()) {
        Cell cell = row.createCell(coll, CellType.STRING);
        cell.setCellStyle(style);
        cell.setCellValue(ScheduleService.weekdays.get(coll));
        sheet.setColumnWidth(coll, 14 * 1000);
    }
}

public List<Lesson> ProcessFile(File excel) throws EncryptedDocumentException, IOException {
    Workbook workbook = WorkbookFactory.create(excel);
    Sheet sheet = workbook.getSheetAt(0);

    List<LessonTmpl> list = new ArrayList<>();
    DataFormatter dataFormatter = new DataFormatter();

    for (int r = 1; r <= 6; r++) {
        Row row = sheet.getRow(r);
        for (int c = 1; c <= 6; c++) {
            Cell cell = row.getCell(c);
            String key = c + "_" + r;
            if (cell != null) {
                String cellValue = dataFormatter.formatCellValue(cell).trim();
                String[] lessons = cellValue.split("\n");
                for (String lesson : lessons) {
                    String[] blocks = lesson.split(";");
                    if(blocks.length == 5){
                        Integer id = Integer.valueOf(blocks[0]);
                        String type = blocks[1];
                        String title = blocks[2];
                        String teacher = blocks[3];
                        String groups_block = blocks[4];
                        String[] groups = groups_block.replaceAll("\\[", "").replaceAll("\\]", "").split(",");

                        System.out.println(key + "=>" + title + " / " + teacher + " / " + groups.length);

                        LessonTmpl tmpl = new LessonTmpl(id, type, c, ScheduleService.weekdays.get(c), r,
ScheduleService.paircomments.get(r), title, teacher, groups_block);
                        list.add(tmpl);
                    }
                }
            }
        }
    }
    List <Lesson> res = new ArrayList<>();
    for (LessonTmpl t:list)
    {
        Lesson l = lessonRepository.getReferenceById(t.getId());
        l.setSlot(slotRepository.findSlotInfo(t.getDay(), t.getNumber()).orElse(null));
    }
}

```

```

        res.add(l);
    }
    return res;
}

public static CellStyle createHeadingStyle(XSSFWorkbook workbook) {
    CellStyle style = workbook.createCellStyle();

    Font font = workbook.createFont();
    font.setFontName("Courier New");
    font.setBold(true);
    font.setFontHeightInPoints((short) 14);
    //font.setUnderline(Font.U_SINGLE);
    //font.setColor(HSSFColor.HSSFColorPredefined.DARK_GREEN.getIndex());
    style.setFont(font);

    style.setAlignment(HorizontalAlignment.CENTER);
    style.setVerticalAlignment(VerticalAlignment.CENTER);
    return style;
}

public static CellStyle createRowStyle(XSSFWorkbook workbook) {
    CellStyle style = workbook.createCellStyle();

    /*
    Font font = workbook.createFont();
    font.setFontName("Courier New");
    font.setFontHeightInPoints((short)14);
    style.setFont(font);
    */

    style.setAlignment(HorizontalAlignment.LEFT);
    style.setVerticalAlignment(VerticalAlignment.TOP);
    style.setWrapText(true);

    return style;
}

public File saveFile(MultipartFile storingFile) throws IOException {

    String filename = (storingFile.getOriginalFilename() != null) ? storingFile.getOriginalFilename() :
    storingFile.getName();
    File dir = new File("/tmp/");
    if (!dir.exists()) {
        dir.mkdirs();
    }
    File loadFile = new File("/tmp/", filename);

    BufferedOutputStream stream = new BufferedOutputStream(new FileOutputStream(loadFile));
    stream.write(storingFile.getBytes());
    stream.flush();
    stream.close();
    return loadFile;
}
}

```

Файл GroupService.java:

```

@Service
public class GroupService {

```

```

    @Autowired

```

					КПІ.ІТ-9308.045440.03.12	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		8

```

GroupRepository groupRepository;

public List<Group> GetGroups()
{
    return groupRepository.findAll();
}

public Group GetGroupById (Integer id)
{
    return groupRepository.getReferenceById(id);
}
}

```

Файл ScheduleService.java:

```

@Service
public class ScheduleService {

    public static HashMap<Integer, String> weekdays = new HashMap<Integer, String>() {
        {
            put(1, "Пн");
            put(2, "Вт");
            put(3, "Ср");
            put(4, "Чт");
            put(5, "Пт");
            put(6, "Сб");
        }
    };

    public static HashMap<Integer, String> paircomments = new HashMap<Integer, String>() {
        {
            put(1, "8:45");
            put(2, "9:30");
            put(3, "10:15");
            put(4, "11:00");
            put(5, "11:45");
            put(6, "12:30");
        }
    };

    @Autowired
    private LessonRepository lessonRepository;

    @Autowired
    private EventRepository eventRepository;

    public List<DayDisplay> GetWeekLessons(Integer teacher_id, Integer group_id) {

        Specification where = Specification.where(null);
        if (teacher_id > 0) {
            where = where.and((item, cq, cb) -> {
                return cb.equal(item.get("teacherId"), teacher_id);
            });
        }
        if (group_id > 0) {
            where = where.and((item, cq, cb) -> {
                return cb.equal(item.join("groups", JoinType.INNER).get("id"), group_id);
            });
        }
        List<Lesson> lessons = lessonRepository.findAll(where);

        List<DayDisplay> week = new LinkedList<>();
    }
}

```

```

        for (int i = 0; i < 7; i++) {
            int day = i;
            List<PairDisplay> daypairs = new ArrayList<>();
            for (int j = 0; j < 7; j++) {
                int number = j;
                List<Lesson> pairlessons = lessons.stream().filter((p) -> p.getDay() == day && p.getNumber() ==
number).collect(Collectors.toList());
                if (!pairlessons.isEmpty()) {
                    PairDisplay pair = new PairDisplay(number, paircomments.get(number),
prepareLessinsDisplay(pairlessons));
                    daypairs.add(pair);
                }
            }
            DayDisplay dayDisplay = new DayDisplay(weekdays.get(day), day, daypairs);
            week.add(dayDisplay);
        }

        return week;
    }

    private List<EventDisplay> prepareLessinsDisplay(List<Lesson> list) {
        List<EventDisplay> items = new LinkedList<>();
        for (Lesson lesson : list) {
            items.add(new EventDisplay(lesson));
        }
        return items;
    }

    public HashMap<String, List<EventDisplay>> PrepareLessonsToHash(List<DayDisplay> lessons) {
        HashMap<String, List<EventDisplay>> list = new HashMap<>();
        for (DayDisplay day : lessons) {
            for (PairDisplay pair : day.getDaylessons()) {
                list.put(day.getDay() + "_" + pair.getNumber(), pair.getLessons());
            }
        }
        return list;
    }

    public HashMap<String, List<Lesson>> PrepareLessonsToExcel(List<Lesson> lessons) {
        HashMap<String, List<Lesson>> list = new HashMap<>();
        for (Lesson lesson : lessons) {
            if (lesson.getSlot() != null) {

                String key = lesson.getSlot().getDay() + "_" + lesson.getSlot().getNumber();
                if (list.containsKey(key)) {
                    list.get(key).add(lesson);
                } else {
                    List<Lesson> splotlist = new ArrayList<>();
                    splotlist.add(lesson);
                    list.put(key, splotlist);
                }
            }
        }
        return list;
    }

    public HashMap<String, List<EventDisplay>> PrepareEventsToHash(LocalDate monday, LocalDate saturday,
User user) {
        HashMap<String, List<EventDisplay>> list = new HashMap<>();

        try {
            Date startWeek = new SimpleDateFormat("yyyy-MM-dd").parse(monday.toString());

```

```

        Date endWeek = new SimpleDateFormat("yyyy-MM-dd").parse(saturday.toString());
        List<Event> allevents = eventRepository.findEventsByWeek(startWeek, endWeek);
        List<Event> events = allevents.stream().filter(e -> e.getUser() == user ||
e.getGroups().contains(user.getGroup())).collect(Collectors.toList());

        for (Event event : events) {
            EventDisplay item = new EventDisplay(event);
            String key = item.getDay() + "_" + item.getNumber();
            if (list.containsKey(key)) {
                list.get(key).add(item);
            } else {
                List<EventDisplay> items = new ArrayList<>();
                items.add(item);
                list.put(key, items);
            }
        }
        return list;
    } catch (ParseException exc) {

    }

    return list;
}

public HashMap<String, List<EventDisplay>> MergeList(HashMap<String, List<EventDisplay>> lessons,
HashMap<String, List<EventDisplay>> events) {
    for (String key : events.keySet()) {
        if (lessons.containsKey(key)) {
            lessons.get(key).addAll(events.get(key));
        } else {
            lessons.put(key, events.get(key));
        }
    }
    return lessons;
}

public List<Lesson> GetAllLessons() {
    return lessonRepository.findAll();
}

}

```

Файл UserService.java:

```

@Service
public class UserService {

    @Autowired
    private UserRepository userRepository;

    public Integer CcreateTeacher (String firstName, String lastName, String email)
    {
        User teach = new User ();
        teach.setFirstName(firstName);
        teach.setLastName(lastName);
        teach.setEmail(email);
        teach.setType(1);
        return userRepository.saveAndFlush(teach).getId();
    }

    public Integer CcreateStudent (String firstName, String lastName, String email)
    {
        User teach = new User ();
    }
}

```

```

        teach.setFirstName(firstName);
        teach.setLastName(lastName);
        teach.setEmail(email);
        teach.setType(2);
        return userRepository.saveAndFlush(teach).getId();
    }

    public List<User> GetAllTeachers ()
    {
        List <User> teachers = new ArrayList<>();
        for (User user:userRepository.findAll())
        {
            if (user.getType()==1)
                teachers.add(user);
        }
        return teachers;
    }

    public List<User> GetAllStudents ()
    {
        List <User> students = new ArrayList<>();
        for (User user:userRepository.findAll())
        {
            if (user.getType()==2)
                students.add(user);
        }
        return students;
    }

    public User GetUser (Integer id)
    {
        return userRepository.findById(id).orElse(null);
    }

    public User GetUserByEmail (String email)
    {
        List<User> users = userRepository.findAll();
        for (User u:users)
        {
            System.out.println(u);
            if (u.getEmail().equals(email)) return u;
        }
        return null;
    }

    public void DeleteUser (Integer id)
    {
        userRepository.deleteById(id);
    }

    public User UpdateUser (TeacherForm form)
    {
        User user;
        if(form.getId() == 0){
            user = new User();
        }else{
            user = userRepository.findById(form.getId()).orElse(new User());
        }

        user.setEmail(form.getEmail());
        user.setFirstName(form.getFirstName());
        user.setLastName(form.getLastName());
    }

```

```

        return userRepository.saveAndFlush(user);
    }

    public void SaveUser(User user)
    {
        userRepository.saveAndFlush(user);
    }
}

```

Файл ApiController.java:

```

@CrossOrigin
@RestController
@RequestMapping("/api")
public class ApiController {

    @Autowired
    UserService teacherService;

    @Autowired
    ScheduleService scheduleService;

    @Autowired
    GroupService groupService;

    @GetMapping(value = "/teachers")
    public ResponseEntity<List<User>> teachers() {

        List<User> list = teacherService.GetAllTeachers();
        return new ResponseEntity<>(list, HttpStatus.OK);
    }

    @GetMapping(value = "/teacher/{id:\\d+}")
    public ResponseEntity<User> teacher(@PathVariable(name = "id") Integer id) {

        User teacher = teacherService.GetUser(id);
        return new ResponseEntity<>(teacher, HttpStatus.OK);
    }

    @GetMapping(value = "/delteacher/{id:\\d+}")
    public ResponseEntity<String> delteacher(@PathVariable(name = "id") Integer id) {

        teacherService.DeleteUser(id);
        return new ResponseEntity<>("Ok", HttpStatus.OK);
    }

    @GetMapping(value = "/groups")
    public ResponseEntity<List<Group>> groups() {

        List<Group> list = groupService.GetGroups();
        return new ResponseEntity<>(list, HttpStatus.OK);
    }

    @PostMapping(value = "updateteacher")
    public ResponseEntity<User> updateteacher(@ModelAttribute("TeacherForm")TeacherForm teacherForm) {
        User teacher = teacherService.UpdateUser(teacherForm);
        return new ResponseEntity<>(teacher, HttpStatus.OK);
    }

    @GetMapping(value = "/lessons", produces = MediaType.APPLICATION_JSON_VALUE)
    public ResponseEntity<Object> lessons(@RequestParam(name = "teacher_id", required = false)
Optional<Integer> var_teacher_id,

```

					КПІ.ІТ-9308.045440.03.12	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		13

```

        @RequestParam(name = "group_id", required = false) Optional<Integer> var_group_id) {

    Integer teacher_id = var_teacher_id.orElseGet(() -> 0);
    Integer group_id = var_group_id.orElseGet(() -> 0);

    List<DayDisplay> lessons = scheduleService.GetWeekLessons(teacher_id, group_id);

    return new ResponseEntity<>(lessons, HttpStatus.OK);
}
}

```

Файл CourseController.java:

```

@Controller
@RequestMapping("/course")
public class CourseController {

    @Autowired
    EventService eventService;

    @Autowired
    private HttpSession httpSession;

    @GetMapping(value = "/edit/{id:\\d+}")
    public String edit(Model model, @PathVariable(name = "id") Integer id) {
        Course course=null;
        if (id != 0) {
            course = eventService.GetCourse(id);
        }
        model.addAttribute("course", course);

        if (CheckCourses((Integer)httpSession.getAttribute("loggedIn"), course)) {
            return "editcourse";
        } else {
            return "viewcourse";
        }
    }

    private boolean CheckCourses(Integer t, Course c)
    {
        for (Lesson l: c.getLessons())
        {
            if (l.getTeacher().getId().equals(t)) return true;
        }
        return false;
    }

    @PostMapping(value = "/update")
    public String edit(Model model, @ModelAttribute("CourseForm") CourseForm courseForm) {
        Course course = eventService.UpdateCourse(courseForm);
        model.addAttribute("course", course);

        return "redirect:/course/edit/" + course.getId();
    }
}

```

Файл CreatorController.java:

```

@Controller

```

					КПІ.IT-9308.045440.03.12	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		14

```

@RequestMapping("/createschedule")
public class CreatorController {

    @Autowired
    UserService userService;

    @Autowired
    GroupService groupService;

    @Autowired
    CreatorService creatorService;

    @Autowired
    private HttpSession httpSession;

    @GetMapping(value = "")
    public String schedule(Model model,
        // @RequestParam(name = "monday", required = false) Optional<String> var_monday,
        // @RequestParam(name = "teacher_id", required = false) Optional<Integer> var_teacher_id,
        // @RequestParam(name = "group_id", required = false) Optional<Integer> var_group_id) {
        Integer teacher_id = var_teacher_id.orElseGet(() -> 0);
        Integer group_id = var_group_id.orElseGet(() -> 0);
        if (httpSession.getAttribute("lessons") == null) {
            httpSession.setAttribute("lessons", creatorService.GetAllLessons());
        }

        List<DayDisplay> lessons = creatorService.GetWeekLessons(teacher_id, group_id, (List<Lesson>)
httpSession.getAttribute("lessons"));

        HashMap<String, List<EventDisplay>> displaylist = creatorService.PrepareLessonsToHash(lessons);
        model.addAttribute("displaylist", displaylist);

        model.addAttribute("weekdays", ScheduleService.weekdays);
        model.addAttribute("paircomments", ScheduleService.paircomments);

        List<User> teachers = userService.GetAllTeachers();
        model.addAttribute("teachers", teachers);

        if (teacher_id > 0) {
            List<Slot> slotslist = userService.GetUser(teacher_id).getSlots();
            List<String> slots = new ArrayList<>();
            for (Slot slot : slotslist) {
                String key = slot.getDay() + "_" + slot.getNumber();
                slots.add(key);
            }
            model.addAttribute("slots", slots);
        } else {
            model.addAttribute("slots", new ArrayList<String>());
        }

        List<Group> groups = groupService.GetGroups();
        model.addAttribute("groups", groups);

        model.addAttribute("teacher_id", teacher_id);
        model.addAttribute("group_id", group_id);

        return "schedule";
    }

    @GetMapping(value = "/saveall")
    public ResponseEntity<Object> saveslots() {
        creatorService.SaveAll((List<Lesson>)httpSession.getAttribute("lessons"));
    }
}

```

```

        return new ResponseEntity<>("ok", HttpStatus.OK);
    }

    @PostMapping(value = "/savesingle")
    public ResponseEntity<Object> savesingle(@RequestBody List<SlotData> slots) {
        httpSession.setAttribute("lessons", creatorService.ProcessSingle(slots,
        (List<Lesson>)httpSession.getAttribute("lessons")));
        return new ResponseEntity<>(slots, HttpStatus.OK);
    }

    @GetMapping(value = "/reset")
    public String resetSlots() {
        httpSession.setAttribute("lessons",
        creatorService.FullReset((List<Lesson>)httpSession.getAttribute("lessons")));
        return "redirect:/createschedule/";
    }

    @GetMapping(value = "/current")
    public String GetCurrentSlots() {
        httpSession.setAttribute("lessons", creatorService.GetAllLessons());
        return "redirect:/createschedule/";
    }

    @GetMapping(value = "/assign")
    public String AssignLessons() {
        httpSession.setAttribute("lessons",
        creatorService.AssignLessons((List<Lesson>)httpSession.getAttribute("lessons")));
        return "redirect:/createschedule/";
    }
}

```

Файл EventController.java:

```

@Controller
@RequestMapping("/event")
public class EventController {

    @Autowired
    UserService userService;

    @Autowired
    EventService eventService;

    @Autowired
    private HttpSession httpSession;

    @GetMapping(value = "/edit/{id:\\d+}")
    public String edit(Model model, @PathVariable(name = "id") Integer id) {
        Event event;
        if (id != 0) {
            event = eventService.GetEvent(id);
        } else {
            event = new Event();
            User u;
            Integer loggedIn = (Integer) httpSession.getAttribute("loggedIn");
            if (loggedIn != null) {
                u = userService.GetUser(loggedIn);
            } else {
                u = userService.GetAllStudents().get(0);
            }
        }
    }
}

```

```

        event.setUser(u);
    }
    model.addAttribute("event", event);

    if (event.getUser().getId() == httpSession.getAttribute("loggedIn")) {
        return "editevents";
    } else {
        return "viewevent";
    }
}

@PostMapping(value = "/update")
public String edit(Model model, @ModelAttribute("EventForm") EventForm eventForm) {
    System.out.println(eventForm);
    Event event = eventService.UpdateEvent(eventForm);
    model.addAttribute("event", event);

    return "redirect:/event/edit/" + event.getId();
}
}

```

Файл MainController.java:

```

@Controller
@RequestMapping("")
public class MainController {

    @Autowired
    UserService userService;

    @Autowired
    ScheduleService scheduleService;

    @Autowired
    GroupService groupService;

    @Autowired
    EventService eventService;

    @Autowired
    CreatorService creatorService;

    @Autowired
    ExcelService excelCreator;

    @Autowired
    private HttpSession httpSession;

    @GetMapping(value = {"/index", ""})
    public String index() {

        return "redirect:/lessons/";
    }

    @GetMapping(value = "teaches")
    public String teaches(Model model) {

        List<User> list = userService.GetAllTeachers();
    }
}

```

					КПІ.IT-9308.045440.03.12	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		17

```

        model.addAttribute("list", list);

        return "teaches";
    }

    @GetMapping("/login")
    public String login() {

        return "login";
    }

    @PostMapping("/login")
    public String loggedin(@RequestParam(name = "login") String login, @RequestParam(name = "password") String
password) {

        User u = userService.GetUserByEmail(login);
        if (login.equals("admin@sm.sm")) {
            HttpSession.setAttribute("loggedIn", 0);
            return "redirect:/lessons";
        } else
        if (u != null) {
            HttpSession.setAttribute("loggedIn", u.getId());
            return "redirect:/lessons";
        } else
            return "redirect:/login";
    }

    @GetMapping("/logout")
    public String logout() {
        HttpSession.setAttribute("loggedIn", null);
        return "redirect:/lessons";
    }

    @GetMapping(value = "/lessons")
    public String lessons(Model model,
        @RequestParam(name = "monday", required = false) Optional<String> var_monday,
        @RequestParam(name = "teacher_id", required = false) Optional<Integer> var_teacher_id,
        @RequestParam(name = "group_id", required = false) Optional<Integer> var_group_id) {
        Integer teacher_id = var_teacher_id.orElseGet(() -> 0);
        Integer group_id = var_group_id.orElseGet(() -> 0);

        LocalDate currentDate = (var_monday.isPresent()) ? LocalDate.parse(var_monday.get()) : LocalDate.now();
        LocalDate currentMonday = currentDate.with(TemporalAdjusters.previousOrSame(DayOfWeek.MONDAY));
        model.addAttribute("currentMonday", currentMonday);
        LocalDate currentSaturday = currentMonday.with(TemporalAdjusters.next(DayOfWeek.SATURDAY));
        model.addAttribute("currentSaturday", currentSaturday);

        LocalDate prevMonday = currentMonday.with(TemporalAdjusters.previous(DayOfWeek.MONDAY));
        model.addAttribute("prevMonday", prevMonday);

        LocalDate nextMonday = currentMonday.with(TemporalAdjusters.next(DayOfWeek.MONDAY));
        model.addAttribute("nextMonday", nextMonday);

        List<DayDisplay> lessons = scheduleService.GetWeekLessons(teacher_id, group_id);

        HashMap<String, List<EventDisplay>>> displaylist = scheduleService.PrepareLessonsToHash(lessons);
        HashMap<String, List<EventDisplay>>> displayresult;
        if (HttpSession.getAttribute("loggedIn") != null && (Integer)HttpSession.getAttribute("loggedIn") > 0) {
            HashMap<String, List<EventDisplay>>> events = scheduleService.PrepareEventsToHash(currentMonday,
currentSaturday, userService.GetUser((Integer) HttpSession.getAttribute("loggedIn")));

            displayresult = scheduleService.MergeList(displaylist, events);
        } else {

```

```

        displayresult = displaylist;
    }
    model.addAttribute("displaylist", displayresult);

    model.addAttribute("weekdays", ScheduleService.weekdays);
    model.addAttribute("paircomments", ScheduleService.paircomments);

    List<User> teachers = userService.GetAllTeachers();
    model.addAttribute("teachers", teachers);

    List<Group> groups = groupService.GetGroups();
    model.addAttribute("groups", groups);

    model.addAttribute("teacher_id", teacher_id);
    model.addAttribute("group_id", group_id);

    if (httpSession.getAttribute("loggedIn") == null) {
        return "lessons";
    } else if ((Integer) httpSession.getAttribute("loggedIn") == 0) {
        return "lessons_admin";
    } else if ((Integer) httpSession.getAttribute("loggedIn") > 0) {
        return "lessons_user";
    } else {
        return "lessons";
    }
}

@GetMapping(value = "/lecture/edit/{id:\\d+}")
public String lessonEdit(Model model, @PathVariable(name = "id") Integer id)
{
    Integer courseId=eventService.GetCourseByLesson(id).getId();
    return "redirect:/course/edit/"+courseId;
}

@GetMapping(value = "/lab/edit/{id:\\d+}")
public String labEdit(Model model, @PathVariable(name = "id") Integer id)
{
    Integer courseId=eventService.GetCourseByLesson(id).getId();
    return "redirect:/course/edit/"+courseId;
}

@GetMapping("/download")
public ResponseEntity<Object> download() {

    if (httpSession.getAttribute("lessons") == null) {
        httpSession.setAttribute("lessons", creatorService.GetAllLessons());
    }
    List<Lesson> lessons = (List<Lesson>) httpSession.getAttribute("lessons");
    HashMap<String, List<Lesson>> displaylist = scheduleService.PrepareLessonsToExcel(lessons);
    XSSFWorkbook workbook = excelCreator.GetScheduler(displaylist);

    try {
        File file = new File("/tmp/scheduler.xlsx");
        workbook.write(new FileOutputStream(file));
        workbook.close();

        InputStreamResource resource = new InputStreamResource(new FileInputStream(file.getAbsolutePath()));

        return ResponseEntity.ok()
            .contentType(MediaType.ALL.valueOf(Files.probeContentType(file.toPath())))
            .header(HttpHeaders.CONTENT_DISPOSITION, "attachment;filename=" + file.getName())
            .body(resource);
    } catch (IOException e) {
        e.printStackTrace(System.out);
    }
}

```

```

        return new ResponseEntity<>(e.getLocalizedMessage(), HttpStatus.OK);
    }
}

@GetMapping("/import")
public String importExcel() {

    return "import";
}

@PostMapping(value = "/upload", consumes = {"multipart/form-data"}, produces =
MediaType.APPLICATION_JSON_VALUE)
public ResponseEntity<Object> upload(@RequestParam("file") MultipartFile file) throws IOException{

    File excel = excelCreator.saveFile(file);
    List<Lesson> list = excelCreator.ProcessFile(excel);
    httpSession.setAttribute("lessons", list);
    return new ResponseEntity<>(list, HttpStatus.OK);
}
}

```

Файл UserController.java:

```

@Controller
@RequestMapping("/teacher")
public class UserController {

    @Autowired
    UserService teacherService;

    @GetMapping(value = "/list")
    public String list(Model model) {

        List<User> list = teacherService.GetAllTeachers();
        model.addAttribute("list", list);
        return "teachers";
    }

    @GetMapping(value = "/edit/{id:\\d+}")
    public String edit(Model model, @PathVariable(name = "id") Integer id) {

        User teacher = teacherService.GetUser(id);
        model.addAttribute("teacher", teacher);

        return "editteachers";
    }

    @PostMapping(value = "/update")
    public String edit(Model model, @ModelAttribute("TeacherForm")TeacherForm teacherForm) {

        User teacher = teacherService.UpdateUser(teacherForm);
        model.addAttribute("teacher", teacher);

        return "redirect:/teacher/edit/" + teacher.getId();
    }
}

```

Файл editcourse.html:

```
<html xmlns:layout="http://www.w3.org/1999/xhtml" layout:decorate="~{layouts/index}">
```

					КПІ.IT-9308.045440.03.12	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		20

```

<head>
  <title>TODO supply a title</title>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
  <div class="col-12" layout:fragment="content">
    <div class="col-4">
      <a th:href="@{/lessons}" class="btn btn-sm btn-outline-primary">Back to Schedule</a>
    </div>

    <form action="/course/update" method="post" class="form-horizontal">

      <input type='hidden' name='id' th:value='${course.getId()}'>

      <div class="form-group">
        <small class="form-text text-muted">Name</small>
        <input type="text" name="name" id="name" class="form-control" th:value='${course.getName()}'
placeholder="Name"/>
      </div>

      <div class="form-group">
        <small class="form-text text-muted">Description</small>
        <input type="text" name="description" id="description" class="form-control"
th:value='${course.getDescription()}' placeholder="Description"/>
      </div>
      <div class="form-group">
        <button type="submit" class="btn btn-primary">Submit</button>
      </div>
    </form>
  </div>
</body>
</html>

```

Файл editevents.html:

```

<html xmlns:layout="http://www.w3.org/1999/xhtml" layout:decorate="~{layouts/index}">
  <head>
    <title>TODO supply a title</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <div class="col-12" layout:fragment="content">
      <div class="col-4">
        <a th:href="@{/lessons}" class="btn btn-sm btn-outline-primary">Back to Schedule</a>
      </div>

      <form action="/event/update" method="post" class="form-horizontal">

        <input type='hidden' name='id' th:value='${event.getId()}'>

        <div class="form-group">
          <small class="form-text text-muted">Name</small>
          <input type="text" name="name" id="name" class="form-control" th:value='${event.getName()}'
placeholder="Name"/>
        </div>

        <div class="form-group">
          <small class="form-text text-muted">Description</small>
          <input type="text" name="description" id="description" class="form-control"
th:value='${event.getDescription()}' placeholder="Description"/>
        </div>
      </form>
    </div>
  </body>
</html>

```

```

    <div class="form-group">
      <label for="exampleInputEmail1">Date</label>
      <input type="date" name="date" class="form-control" th:value="{event.FormatDate()}"
placeholder="Date">
    </div>
    <div class="form-group">
      <small class="form-text text-muted">Pair</small>
      <input type="text" name="number" id="number" class="form-control"
th:value="{event.getNumber()}" placeholder="Number"/>
    </div>
    <div class="form-group">
      <button type="submit" class="btn btn-primary">Submit</button>
    </form>
  </div>
</body>
</html>

```

Файл import.html:

```

<html xmlns:layout="http://www.w3.org/1999/xhtml" layout:decorate="~{layouts/index}"> <head>
  <title>TODO supply a title</title>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
  <div class="col-12" layout:fragment="content">
    <div class="row">
      <div class="col-2">
        <a href="/createschedule" class="btn btn-sm btn-outline-primary">Back</a>
      </div>
      <div class="col-8">
        <form action="/import" method="post" class="form-horizontal" >
          <input type="file" name="file" id="fileToUpload" >
          <input type="button" value="Confirm" onclick="FileUpload()">
        </form>
        <table border="1" id="result" class="table table-striped table-condensed"></table>
      </div>
    </div>
    <script>
      function FileUpload()
      {
        var formData = new FormData();
        formData.append('file', $('#fileToUpload')[0].files[0]);

        axios.post("/upload", formData, {headers: {'Content-Type': 'multipart/form-data'}})
          .then(response => {
            console.log(response.data);
            $('#rowTmpl').tmpl(response.data).appendTo('#result');
          });
      }
    </script>
    <script id="rowTmpl" type="text/x-jquery-tmpl">

    <tr><td>${id}</td><td>${daytitle}</td><td>${pair}</td><td>${type}</td><td>${lesson}</td><td>${teacher}</t
d><td>${groups}</td></tr>
    </script>

  </div>
</body>
</html>

```

Файл lessons.html:

```
<html xmlns:layout="http://www.w3.org/1999/xhtml" layout:decorate="~{layouts/index}">
  <head>
    <title>TODO supply a title</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <div class="col-12" layout:fragment="content">
      <div class="row">
        <div class="col-4">
          <form class="form-inline">
            <div class="form-group mb-2">
              <label>Teachers: </label>
              <select class="form-control form-control-sm" name="teacher_id" style='width: 200px'
onchange="this.form.submit()">
                <option value="0">---</option>
                <option th:each="teacher : ${teachers}"
                  th:value="${teacher.id}"
                  th:text="${teacher.firstName} + ' ' + ${teacher.lastName}"
                  th:selected="${teacher.id} == ${teacher_id} ? true : false"></option>
              </select>
            </div>
            <div class="form-group mx-sm-3 mb-2">
              <label>Groups: </label>
              <select class="form-control form-control-sm" name="group_id" style='width: 200px'
onchange="this.form.submit()">
                <option value="0">---</option>
                <option th:each="group : ${groups}"
                  th:value="${group.id}"
                  th:text="${group.name}"
                  th:selected="${group.id} == ${group_id} ? true : false"></option>
              </select>
            </div>
            <input type='hidden' name='monday' th:value="${currentMonday}" />
          </form>
        </div><div class="col-7">
          <a
            th:href="@{/lessons?group_id='${group_id}'&teacher_id='${teacher_id}'&monday='${prevMonday}'}"
            class="btn btn-sm btn-outline-primary">Prev ([[ ${prevMonday} ]])</a>
          Week: [[ ${currentMonday} ]] - [[ ${currentSaturday} ]]
          <a
            th:href="@{/lessons?group_id='${group_id}'&teacher_id='${teacher_id}'&monday='${nextMonday}'}"
            class="btn btn-sm btn-outline-primary">Next ([[ ${nextMonday} ]])</a>
        </div>
        <div class="col-1">
          <a th:href="@{/login}" class="btn btn-sm btn-outline-primary">Log In</a>
        </div>
      </div>
      <table class="table table-striped">
        <thead>
          <tr><td style='width: 12%'>#</td><td style='width: 12%' th:each="weekday: ${weekdays}"> <span
th:text="${weekday.value}"></span> </td></tr>
        </thead>
        <tbody>
          <tr th:each="pair: ${paircomments}">
            <td th:text="${pair.value}"></td>
            <td th:each="weekday: ${weekdays}">
              <div th:with="tplKey=${weekday.key}+'_'+${pair.key}">
                <span th:if="${displaylist.containsKey(tplKey)}">

```

```

        <div th:each="elem: ${displaylist.get(tplKey)}" class='box'>
            <span class="badge badge-pill badge-info" th:if="${elem.type ==
'lab'}">[[${elem.type}]]</span>
            <span class="badge badge-pill badge-primary" th:if="${elem.type ==
'lecture'}">[[${elem.type}]]</span>
            <span class="badge badge-pill badge-warning" th:if="${elem.type ==
'event'}">[[${elem.type}]]</span>
            <a th:href="@{'/'+${elem.type}+'/edit/'+${elem.id}}"> [[${elem.name}]] </a>
            <br />[[${elem.user.firstName} ${elem.user.lastName}]]
        </div>
    </span>
</div>
</td>
</tr>
</tbody>
</table>

<style >
    .box {
        border: 1px solid #666;
        border-radius: .5em;
        padding: 8px;
        margin: 5px;
        background-color: #dee2e6;
    }
</style>
</div>

</body>
</html>

```

Файл schedule.html:

```

<html xmlns:layout="http://www.w3.org/1999/xhtml" layout:decorate="~{layouts/index}">
<head>
    <title>TODO supply a title</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
    <div class="col-12" layout:fragment="content">
        <div class="row">
            <div class="col-4">
                <a href="/lessons" class="btn btn-sm btn-outline-primary">Back to main page</a>
                <a href="javascript:;" class="btn btn-sm btn-outline-primary" onclick="SaveChanges()">Save
changes</a>
            </div>
            <div class="col-4">
                <form class="form-inline">
                    <div class="form-group mb-2">
                        <label>Teachers: </label>
                        <select class="form-control form-control-sm" name="teacher_id" style='width: 200px'
onchange="this.form.submit()">
                            <option value="0">---</option>
                            <option th:each="teacher : ${teachers}"
                                th:value="${teacher.id}"
                                th:text="${teacher.firstName} + ' ' + ${teacher.lastName}"
                                th:selected="${teacher.id} == ${teacher_id} ? true : false"></option>
                        </select>
                    </div>
                    <div class="form-group mx-sm-3 mb-2">
                        <label>Groups: </label>

```

```

        <select class="form-control form-control-sm" name="group_id" style='width: 200px'
onchange="this.form.submit()">
            <option value="0">---</option>
            <option th:each="group : ${groups}"
                th:value="${group.id}"
                th:text="${group.name}"
                th:selected="${group.id} == ${group_id} ? true : false"></option>
        </select>
    </div>
</form>
</div>
<div class="col-4">
    <a href="/createschedule/reset" class="btn btn-sm btn-outline-primary">Reset all</a>
    <a href="/createschedule/assign" class="btn btn-sm btn-outline-primary">Assign all</a>
    <a href="/createschedule/current" class="btn btn-sm btn-outline-primary">Get current schedule</a>
    <a href="/download" class="btn btn-sm btn-outline-primary">Download</a>
    <a href="/import" class="btn btn-sm btn-outline-primary">Upload</a>
</div>
</div>
<div class="row"><div class="col-10">
    <table class="table table-striped">
        <thead>
            <tr><td style='width: 70px'>#</td>
                <td style='width: 16%' th:each="weekday: ${weekdays}"> <span
th:text="${weekday.value}"></span></td>
            </tr>
        </thead>
        <tbody>
            <tr th:each="pair: ${paircomments}">
                <td th:text="${pair.value}"></td>
                <td th:each="weekday: ${weekdays}">
                    <div th:with="tplKey=${weekday.key}+'_'+${pair.key}">
                        <div class="notset" th:attr="data-day=${weekday.key},data-number=${pair.key}"
th:if="${slots.contains(tplKey)}">+</div>
                        <div class="box emptyslot" th:attr="data-day=${weekday.key},data-number=${pair.key}"
th:if="${!slots.contains(tplKey)}">+</div>
                        <span th:if="${displaylist.containsKey(tplKey)}">
                            <div th:each="elem: ${displaylist.get(tplKey)}" class="box lesson" draggable="true"
th:id="'lesson_' + ${elem.id}" th:attr="data-day=${elem.day},data-number=${elem.number}">
                                <span class="badge badge-pill badge-info" th:if="${elem.type ==
'lab'}">[[${elem.type}]]</span>
                                <span class="badge badge-pill badge-primary" th:unless="${elem.type ==
'lab'}">[[${elem.type}]]</span>
                                [[${elem.name}]]
                                <br />[[${elem.user.firstName} ${elem.user.lastName}]]
                                <br />[[${elem.groups}]]
                            </div>
                        </span>
                    </td>
                </tr>
            </tbody>
        </table>
</div>
<div class="col-2">
    <h5>Unassigned Lessons</h5>
    <div class="box emptyslot" th:attr="data-day=0,data-number=0">+</div>
    <span th:if="${displaylist.containsKey('0_0')}">
        <div th:each="elem: ${displaylist.get('0_0')}" class="box lesson" draggable="true" th:id="'lesson_' +
${elem.id}" th:attr="data-day=${elem.day},data-number=${elem.number}">
            <span class="badge badge-pill badge-info" th:if="${elem.type == 'lab'}">[[${elem.type}]]</span>
        </div>
    </span>
</div>

```

```

        <span class="badge badge-pill badge-primary" th:unless="{elem.type ==
'lab'}">[[${elem.type}]]</span>
        [[${elem.name}]]
        <br />[[${elem.user.firstName} ${elem.user.lastName}]]
        <br />[[${elem.groups}]]
    </div>
</span>

</div>
</div>

<script>

var dragSrcEl = null;
let items = document.querySelectorAll('.box');

function handleDragStart(e) {
    this.style.opacity = '0.4';

    dragSrcEl = this;

    e.dataTransfer.effectAllowed = 'move';
    e.dataTransfer.setData('text/html', this.innerHTML);
    e.dataTransfer.setData('element_id', $(this).attr('id'));
}

function handleDragOver(e) {
    if (e.preventDefault) {
        e.preventDefault();
    }

    e.dataTransfer.dropEffect = 'move';

    return false;
}

function handleDragEnter(e) {
    this.classList.add('over');
}

function handleDragLeave(e) {
    this.classList.remove('over');
}

function handleDrop(e) {
    if (e.stopPropagation) {
        e.stopPropagation(); // stops the browser from redirecting.
    }
    if (dragSrcEl != this) {
        var element_id = e.dataTransfer.getData('element_id');
        var new_day = $(this).data('day');
        var new_number = $(this).data('number');
        if (this.innerHTML == '+') {
            var lessonBox = $(dragSrcEl).remove();
            $(lessonBox).insertAfter(this);
            $(lessonBox).data('day', new_day);
            $(lessonBox).data('number', new_number);
        } else {
            var prev_day = $(dragSrcEl).data('day');
            var prev_number = $(dragSrcEl).data('number');
            var new_id = $(this).attr('id');
            dragSrcEl.innerHTML = this.innerHTML;
            this.innerHTML = e.dataTransfer.getData('text/html');
        }
    }
}

```

```

        $('#'+ element_id).data('day', new_day);
        $('#'+ element_id).data('number', new_number);

        $('#'+ new_id).data('day', prev_day);
        $('#'+ new_id).data('number', prev_number);
    }
    SaveSingle();
}

return false;
}

function handleDragEnd(e) {
    this.style.opacity = '1';

    items.forEach(function (item) {
        item.classList.remove('over');
    });
}

function setEventListener(item) {
    item.addEventListener('dragstart', handleDragStart, false);
    item.addEventListener('dragenter', handleDragEnter, false);
    item.addEventListener('dragover', handleDragOver, false);
    item.addEventListener('dragleave', handleDragLeave, false);
    item.addEventListener('drop', handleDrop, false);
    item.addEventListener('dragend', handleDragEnd, false);
}
document.addEventListener('DOMContentLoaded', (event) => {
    items.forEach(function (item) {
        setEventListener(item);
    });
});

function SaveChanges() {
    axios.get('/createschedule/saveall').then(function (response) {
        console.log(response.data);
        $('#alert').show();
        setTimeout(() => {
            $('#alert').hide();
        }, 1000);
    }).catch(function (error) {
        console.log(error);
    });
};

function SaveSingle() {
    let lessons = document.querySelectorAll('.lesson');
    let data = [];
    lessons.forEach(function (item) {
        var elem_id = $(item).attr('id').replace("lesson_", "");
        var day = $(item).data('day');
        var number = $(item).data('number');
        var lesson = {'id': elem_id, 'day': day, 'number': number};
        data.push(lesson);
    });
    axios.post('/createschedule/savesingle', data).then(function (response) {
        console.log(response.data);
    }).catch(function (error) {
        console.log(error);
    });
}

```

```

    });
    ;
  }
</script>

<style >
  .box {
    border: 2px solid #666;
    border-radius: .5em;
    padding: 8px;
    margin: 5px;
    cursor: move;
  }

  .alertbox {
    border: 1px solid #28a745;
    border-radius: .5em;
    padding: 3px;
    margin: 3px;
    text-align: center;
    color: #28a745;
  }

  .box.over {
    border: 2px dotted #666;
  }

  .lesson {
    background-color: #dee2e6;
  }

  .emptyslot {
    background-color: white;
  }

  .notset {
    border: 1px solid red;
    border-radius: .5em;
    padding: 8px;
    margin: 5px;
    background-color: white;
  }
</style>

</div>
</body>
</html>

```

Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

_____ Едуард ЖАРІКОВ

“ ____ ” _____ 2023 р.

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ОРГАНІЗАЦІЇ НАВЧАННЯ В
ДИСТАНЦІЙНОМУ РЕЖИМІ**

Програма та методика тестування

КПІ.IT-9308.045440.04.51

“ПОГОДЖЕНО”

Керівник проєкту:

Константин НОСОВ

Нормоконтроль:

Максим ГОЛОВЧЕНКО

Виконавець:

Антон ГРИГОР'ЄВ

Київ – 2023

ЗМІСТ

1	ОБ’ЄКТ ВИПРОБУВАНЬ.....	3
2	МЕТА ТЕСТУВАННЯ	4
3	МЕТОДИ ТЕСТУВАННЯ.....	5
4	ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ	6

					КПІ.ІТ-9308.045440.04.51	Арк.
						2
Змін	Арк.	№ докум.	Підп.	Дата.		

1 ОБ'ЄКТ ВИПРОБУВАНЬ

Об'єктом випробування є програмне забезпечення для організації навчання в дистанційному режимі, виконане в вигляді веб-застосунку на основі мови програмування Java.

					КПІ.ІТ-9308.045440.04.51	Арк.
						3
Змін	Арк.	№ докум.	Підп.	Дата.		

2 МЕТА ТЕСТУВАННЯ

Метою тестування є наступне:

- перевірка правильності роботи програмного забезпечення відповідно до функціональних вимог;
- перевірка збереження даних;
- перевірка сумісності веб-додатку з останніми версіями сучасних браузерів (Chrome, Opera, Firefox, ...);
- перевірка сумісності застосунку з різними операційними системами (Windows, Linux, ...);
- знаходження проблем, помилок і недоліків з метою їх усунення;
- перевірка зручності графічного інтерфейсу.

					КПІ.ІТ-9308.045440.04.51	Арк.
Змін	Арк.	№ докум.	Підп.	Дата.		4

3 МЕТОДИ ТЕСТУВАННЯ

Для тестування програмного забезпечення використовуються такі методи:

- статичне тестування – перевіряється програма разом з усією документацією, яка аналізується на предмет дотримання стандартів програмування;
- функціональне тестування – полягає у перевірці відповідності реальної поведінки програмного забезпечення очікувань;
- системне тестування – перевіряється усе програмне забезпечення в цілому;
- мануальне тестування – тестування без використання автоматизації, тест-кейси пише особа, що тестує програмне забезпечення;

					КПІ.ІТ-9308.045440.04.51	Арк.
						5
Змін	Арк.	№ докум.	Підп.	Дата.		

4 ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ

Тестування виконується мануально з використанням наскрізного тестування (End-to-end, E2E, Chain testing) та написанням unit-тестів, з метою знаходження помилок та недоліків як у функціональній частині програмного забезпечення так і в зручності користування. Для того, щоб перевірити працездатність та відмовостійкість застосунку, необхідно провести наступні тестування:

- динамічне тестування на відповідність функціональним вимогам;
- тестування роботи внутрішніх алгоритмів;
- тестування роботи в різних браузерах;
- тестування інтерфейсу користувача;
- тестування зручності використання;

					КПІ.ІТ-9308.045440.04.51	Арк.
Змін	Арк.	№ докум.	Підп.	Дата.		6

Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

_____ Едуард ЖАРІКОВ

“ ____ ” _____ 2023 р.

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ОРГАНІЗАЦІЇ НАВЧАННЯ В
ДИСТАНЦІЙНОМУ РЕЖИМІ**

Керівництво користувача

КПІ.IT-9308.045440.05.34

“ПОГОДЖЕНО”

Керівник проєкту:

Константин НОСОВ

Нормоконтроль:

Максим ГОЛОВЧЕНКО

Виконавець:

Антон ГРИГОР'ЄВ

Київ – 2023

ЗМІСТ

1	ПРИЗНАЧЕННЯ ПРОГРАМИ	3
2	ПІДГОТОВКА ДО РОБОТИ З ПРОГРАМНИМ ЗАБЕЗПЕЧЕННЯМ.....	4
2.1	Системні вимоги для коректної роботи.....	4
2.2	Завантаження застосунку	4
2.3	Перевірка коректної роботи.....	4
3	ВИКОНАННЯ ПРОГРАМИ	5
3.1	Використання програми користувачем	5
3.2	Використання програми адміністратором	8

					КПІ.ІТ-9308.045440.05.34	Арк.
						2
Змін.	Арк.	№ докум.	Підп.	Дата.		

1 ПРИЗНАЧЕННЯ ПРОГРАМИ

Дана система призначена для полегшення роботи викладачів і студентів під час навчання в дистанційному режимі, шляхом надання можливостей відображення, редагування, і генерації розкладу, а також відслідковування певних подій.

					КПІ.ІТ-9308.045440.05.34	Арк.
						3
Змін.	Арк.	№ докум.	Підп.	Дата.		

2 ПІДГОТОВКА ДО РОБОТИ З ПРОГРАМНИМ ЗАБЕЗПЕЧЕННЯМ

2.1 Системні вимоги для коректної роботи

- Наявність пристрою з якого можна отримати доступ в Інтернет;
- наявність доступу до Інтернету;

2.2 Завантаження застосунку

На даний момент, система доступна лише через браузер, так що завантажувати її не потрібно.

2.3 Перевірка коректної роботи

Користувач повинен перейти по посиланню за адресою на сайт цього проекту. На головній сторінці повинен відобразитися розклад в вигляді таблиці, який можна фільтрувати за групою і/або викладачем. Кожен елемент розкладу повинен відображувати тип цього елемента (лекція, лабораторна, подія, тощо), назву, ім'я викладача/організатора, і групи які підписані на цей елемент. В вуглі повинна бути кнопка для авторизації.

					КПІ.ІТ-9308.045440.05.34	Арк.
						4
Змін.	Арк.	№ докум.	Підп.	Дата.		

3 ВИКОНАННЯ ПРОГРАМИ

3.1 Використання програми користувачем

При заході на головну сторінку вебсайту буде відображено поточний розклад уроків/пар (рисунки 3.1).

Рисунок 3.1 – Початкова сторінка додатку

Користувач має змогу відфільтрувати розклад, вибравши необхідну групу або викладача з списків в верхньому лівому вуглі, змінити тиждень що відображується використовуючи кнопки зверху в центрі, а також перейти на сторінку авторизації за кнопкою в верхньому правому вуглі.

На сторінці авторизації (рисунки 3.2) користувач має ввести власну email адресу і пароль, після чого натиснути кнопку підтвердження. Якщо введені дані неправильні, програма запропонує ввести їх ще раз, якщо правильні, користувача перекине назад на головну сторінку, в цей раз з новим доступним функціоналом (рисунки 3.3).

Авторизація

Email address

Password

☒ Remember me

[Forgot password?](#)

Sign in

Рисунок 3.2 – Сторінка авторизації

Teachers	---	Groups	IT1	Prev (2023-05-22)	Week: 2023-05-29 – 2023-06-03	Next (2023-06-05)	Log out	New Event
#	Пн	Вт	Ср	Чт	Пт	Сб		
8:45			Lecture Math (asd) Mikola Tkachenko		Lecture Math (asd) Mikola Tkachenko			
9:30	Lab Math (asd) Lubov Koval	Lab Physics (asd) Stepan Makar	Lecture Algorithms (A great lesson) Lubov Koval	Lecture Physics (asd) Stepan Makar	Lab Physics (asd) Stepan Makar			
10:15	Lab Math (asd) Lubov Koval	Lab Programming (asd) Oksana Kolosok		Lab English (asd) Maxim Mohylyko	Lecture Physics (asd) Stepan Makar			
				event My Own Event (Cool) Anton Grigoryev				
11:00	Lab Security (asd) Oksana Kolosok		Lab Algorithms (A great lesson) Lubov Koval	Lab English (asd) Maxim Mohylyko	Lab Programming (asd) Oksana Kolosok			
11:45			Lecture Programming (asd) Oleh Mityk		Lecture Security (asd) Oksana Kolosok			
12:30								

Рисунок 3.3 – Початкова сторінка додатку після авторизації

Тепер користувач також може бачити події, видимі йому – це або події створені цим користувачем, або події видимі його групі. В правому верхньому вуглу є кнопка виходу з аккаунта, а також кнопка створення нової

події. При натисканні на неї, або при натисканні на назву створеної користувачем події, відкриється вікно редагування події (рисунок 3.4).

Back to Schedule

Name

My Own Event

Description

Cool

Date

01.06.2023

Pair

3

Submit

Рисунок 3.4 – Вікно редагування події

В відповідних полях можна змінити назву події, її опис, дату проведення, і номер уроку/пари в який ця подія проводиться. Натискання кнопки “Submit” збереже внесені зміни, створюючи нову подію якщо її не існує. Кнопка в верхньому лівому вуглу повертає на головну сторінку без збереження. Якщо користувач є викладачем, і перейде по посиланню на урок який він викладає, то також зможе відредагувати дані про навчальний курс подібним способом.

Якщо ж користувач переходить по посиланню на урок або подію на редагування якої у нього немає прав, він побачить лише назву і опис (рисунок 3.5).

					КПІ.ІТ-9308.045440.05.34	Арк.
						7
Змін.	Арк.	№ докум.	Підп.	Дата.		

[Back to Schedule](#)

Name

Algorythms

Description

A great lesson

Рисунок 3.5 – Вікно перегляду інформації про курс

3.2 Використання програми адміністратором

Щоб користуватися власним функціоналом, адміністратор має авторизуватися в власний аккаунт. Після введення правильних даних, адміністратор побачить власну початкову сторінку (рисунок 3.6), і скориставшись кнопкою в верхньому правому вуглу, він може перейти на сторінку редагування розкладу (рисунок 3.7)

Teachers	---	Groups	IT1	Prev (2023-05-22)	Week: 2023-05-29 - 2023-06-03	Next (2023-06-05)	Log out	Edit schedule
#	Пн	Вт	Ср	Чт	Пт	Сб		
8:45		Lecture Math (asd) Mikola Tkachenko		Lecture Math (asd) Mikola Tkachenko				
9:30	Lab Math (asd) Lubov Koval	Lab Physics (asd) Stepan Makar	Lecture Algorythms (A great lesson) Lubov Koval	Lecture Physics (asd) Stepan Makar	Lab Physics (asd) Stepan Makar			
10:15	Lab Math (asd) Lubov Koval	Lab Programming (asd) Oksana Kolosok		Lab English (asd) Maxim Mohylko	Lecture Physics (asd) Stepan Makar			
11:00	Lab Security (asd) Oksana Kolosok		Lab Algorythms (A great lesson) Lubov Koval	Lab English (asd) Maxim Mohylko	Lab Programming (asd) Oksana Kolosok			
11:45		Lecture Programming (asd) Oleh Mityk			Lecture Security (asd) Oksana Kolosok			
12:30								

Рисунок 3.6 – Початкова сторінка адміністратора

[Back to main page](#)
[Save changes](#)

Teachers: Oksana Kolosok
Groups: ---

[Reset all](#)
[Assign all](#)
[Get current schedule](#)
[Download](#)
[Upload](#)

#	Пн	Вт	Ср	Чт	Пт	Сб
8:45	+	+	+	+	+	+
9:30	+	+	+	+	+	+
10:15	+ lab Programming Oksana Kolosok [IT2]	+ lab Programming Oksana Kolosok [IT1]	+ lab Programming Oksana Kolosok [IT2]	+ lab Security Oksana Kolosok [IT2]	+	+
11:00	+	+	+	+	+ lab Programming Oksana Kolosok [IT1]	+
11:45	+	+	+	+	+	+
12:30	+	+	+	+	+	+

Unassigned Lessons

+

lab Security Oksana Kolosok [IT1]

lab Security Oksana Kolosok [IT1, IT2, IT3, IT4]

Рисунок 3.7 – Сторінка редагування розкладу

На сторінці редагування розкладу, адміністратор бачить всі уроки, які, так же як і на головній сторінці, можна фільтрувати за групою і/або викладачем. Справа в колонці знаходяться всі уроки які на даний момент не мають певної позиції. Щоб змінити позицію уроку, необхідно натиснути на його поле і перетягнути його на нову позицію – або в список нерозподілених уроків. Червоним підсвічуються уроки в які даний вчитель не може проводити пари – в них неможливо назначити уроки. Натиснення кнопки “Reset all” перенесе всі уроки в список нерозподілених. Кнопка “Get current schedule” поверне актуальний розклад. Кнопка “Assign All” активує алгоритм що автоматично розмістить всі нерозподілені уроки в оптимальні для них позиції. Всі зміни в редакторі розкладу тимчасові – щоб їх зберегти на сервер, необхідно натиснути кнопку “Save changes”. Альтернативно, Адміністратор може натиснути кнопку “Download” щоб завантажити розклад з редактору в вигляді excel-файлу. Цей файл можна потім вивантажити в редактор, натиснувши кнопку “Upload”.

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

_____ Едуард ЖАРІКОВ

“ ____ ” _____ 2023 р.

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ОРГАНІЗАЦІЇ НАВЧАННЯ В
ДИСТАНЦІЙНОМУ РЕЖИМІ**

Графічний матеріал

КПІ.IT-9308.045440.06.99

“ПОГОДЖЕНО”

Керівник проєкту:

Константин НОСОВ

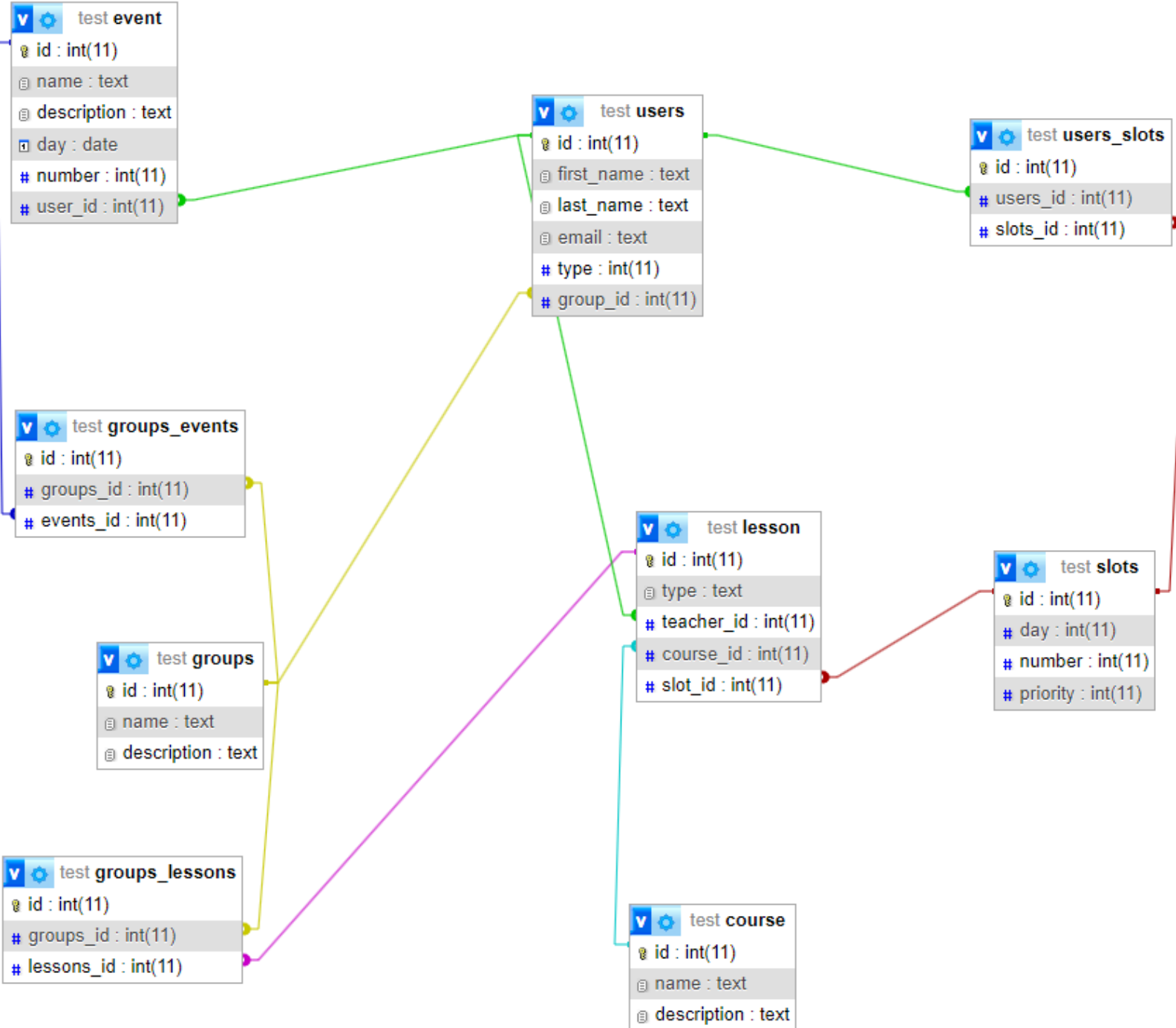
Нормоконтроль:

Максим ГОЛОВЧЕНКО

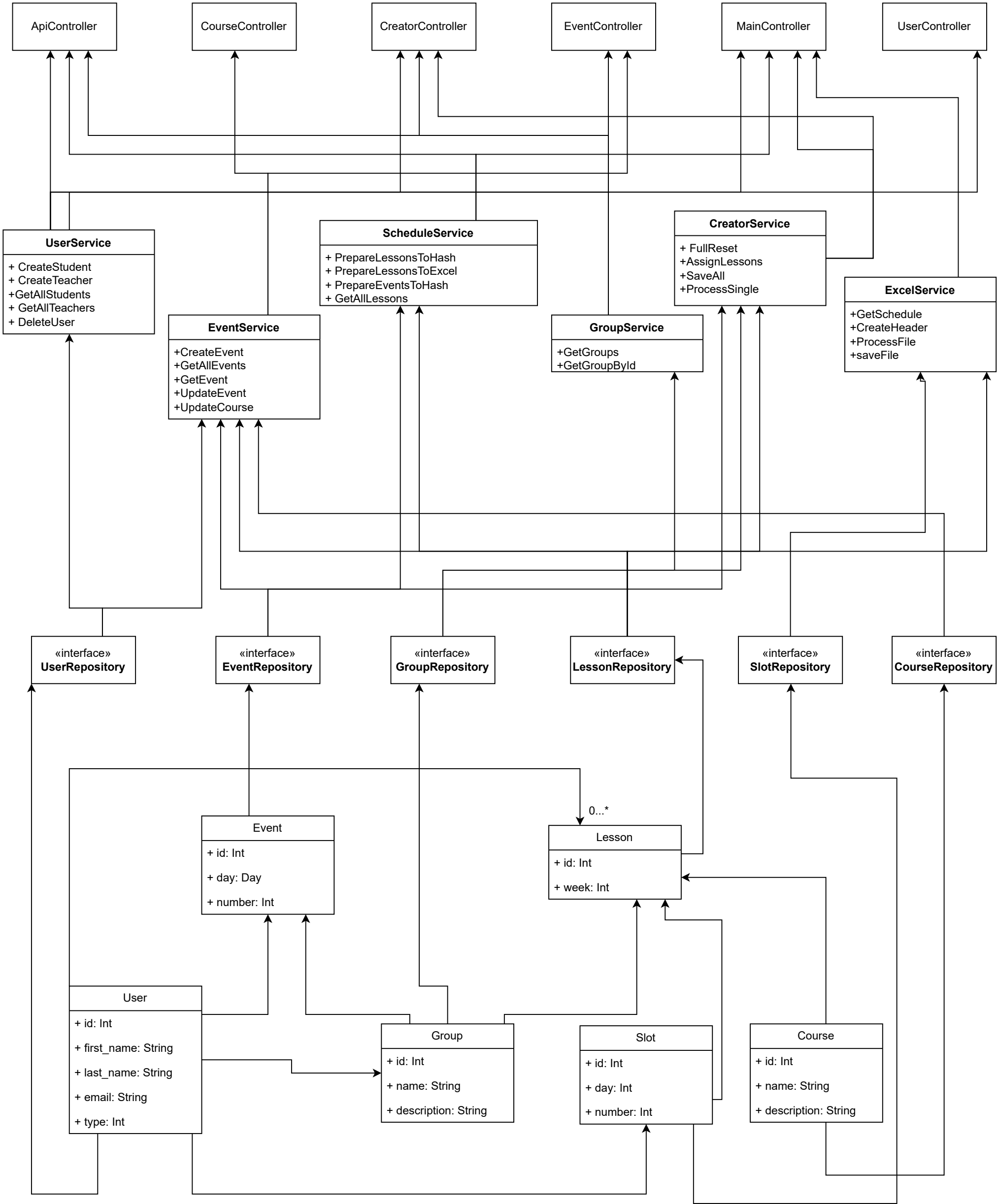
Виконавець:

Антон ГРИГОР'ЄВ

Київ – 2023



					КПІ.ІТ-9308.045440.06.99.СБД								
					Схема бази даних				Лит.		Арк.	Аркуші	
Зм.	Арк.	№ докум.	Підп.	Дата									1
Розроб.		Григор'єв А.С.											
Перев.		Носов К.С.											
									Аркуш		Аркуші		
					Програмне забезпечення для організації навчання в дистанційному режимі				КПІ ім.Ігоря Сікорського Кафедра ІПІ гр. ІП\Т-9Х				
Н. Кон.		Піщук К.І.											
Затв.		Жаріков Е.В.											



					КПІ.ІТ-9308.045440.07.99.СБД							
					Схема структурна класів програмного забезпечення			Лит.		Арк.	Аркушів	
Зм.	Арк.	№ докум.	Підп.	Дата								1
Розроб.	Григор'єв А.С.											
Перев.	Носов К.С.											
Т. Кон.												
					Програмне забезпечення для організації навчання в дистанційному режимі			Аркуш		Аркушів		
Н. Кон.	Піщук К.І.							КПІ ім.Ігоря Сікорського Кафедра ІПІ гр. ІП\Т-9Х				
Затв.	Жаріков Е.В.											



					КПІ.ІТ-9308.045440.08.99.СС				
					Схема структурна варіантів використання				
Зм.	Арк.	№ докум.	Підп.	Дата					
Розроб.		Григор'єв А.С.							
Перев.		Носов К.С.							
Т. Кон.									
Н. Кон.		Піщук К.І.							
Затв.		Жаріков Е.В.							
					Програмне забезпечення для організації навчання в дистанційному режимі				
					КПІ ім.Ігоря Сікорського Кафедра ІПІ гр. ІП\Т-9Х				
					Лит.		Арк.		Аркушів
									1
					Аркуш			Аркушів	