

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ

Кафедра програмної інженерії

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: «РОЗРОБКА ДОДАТКУ ГЕНЕРАЦІЇ
СЕРТИФІКАТИВ ЗАСОБАМИ CMS MOODLE»

Виконав: студент 4 курсу, групи 6.1210-2пi
спеціальності 121 інженерія програмного забезпечення
(шифр і назва спеціальності)

освітньої програми програмна інженерія
(назва освітньої програми)

М.С. Бен Мамія

(ініціали та прізвище)

Керівник завідувач кафедри програмної інженерії,
доцент, к.ф.-м.н. Лісняк А.О.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент завідувач кафедри фундаментальної та прикладної
математики, професор, д.т.н. Гребенюк С.М.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет математичний

Кафедра програмної інженерії

Рівень вищої освіти бакалавр

Спеціальність 121 інженерія програмного забезпечення

(шифр і назва)

Освітня програма програмна інженерія

ЗАТВЕРДЖУЮ

Завідувач кафедри програмної
інженерії, к.ф.-м.н., доцент

_____ Лісняк А.О.

(підпис)

“ _____ ” _____ 2023 р.

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

Бен Мамії Мохамед Салаху

(прізвище, ім'я та по-батькові)

1. Тема роботи Розробка додатку генерації сертифікатів засобами CMS Moodle

керівник роботи Лісняк Андрій Олександрович, к.ф.-м.н., доцент

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

затверджені наказом ЗНУ від « 21 » грудня 2023 року № 2180-с

2. Строк подання студентом роботи 03.06.2024 р.

3. Вихідні дані до роботи 1. Постановка задачі.

2. Перелік літератури.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Постановка задачі, аналіз предметної області.

2. Проектування.

3. Реалізація та тестування.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____

презентація за темою доповіді

6. Консультанти розділів роботи

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

7. Дата видачі завдання 25.12.2023 р.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	08.02.2024	
2.	Збір вихідних даних.	15.02.2024	
3.	Обробка методичних та теоретичних джерел.	01.03.2024	
4.	Розробка першого та другого розділу.	12.04.2024	
5.	Розробка третього розділу.	20.05.2024	
6.	Оформлення та нормоконтроль кваліфікаційної роботи бакалавра.	27.05.2024	
7.	Захист кваліфікаційної роботи.	20.06.2024	

Студент _____
(підпис)

М.С. Бен Мамія _____
(ініціали та прізвище)

Керівник роботи _____
(підпис)

А.О. Ліснюк _____
(ініціали та прізвище)

Нормоконтроль пройдено

Нормоконтролер _____
(підпис)

А.В. Столярова _____
(ініціали та прізвище)

РЕФЕРАТ

Кваліфікаційна робота бакалавра «Розробка додатку генерації сертифікатів засобами CMS Moodle»: 73 с., 34 рис., 10 джерел, 3 додатки.

АВТОМАТИЗАЦІЯ, ГЕНЕРАЦІЯ СЕРТИФІКАТІВ, МОДУЛЬ, MOODLE, MYSQL, PHP, UML.

Об'єкт дослідження – додаток генерації сертифікатів, інструменти взаємодії з Moodle.

Мета роботи – розробити додаток генерації сертифікатів.

Методи дослідження – моделювання, проектування, програмний, аналітичний.

У роботі розглянуто процес проектування, реалізації та тестування додатку для генерації сертифікатів. Використано UML для візуалізації системи, зокрема діаграми варіантів використання, діяльності, послідовності та розгортання.

Система надає можливість створювати, редагувати та видаляти шаблони сертифікатів, а також генерувати сертифікати для слухачів курсів.

Таким чином, розроблений додаток для генерації сертифікатів забезпечує автоматизацію процесу створення сертифікатів на базі платформи Moodle. Це дозволяє зменшити час та ресурси, необхідні для цього завдання, та підвищити ефективність освітніх процесів.

SUMMARY

Bachelor's qualifying paper "Development of a Certificate Generation Application Using CMS Moodle": 73 pages, 34 figures, 10 references, 3 supplements.

AUTOMATION, CERTIFICATE GENERATION, MODULE, MOODLE, MYSQL, PHP, UML.

The object of the study is the certificate generation application, tools for interaction with Moodle.

The aim of the study is to develop a certificate generation application.

The methods of research are modeling, design, programming, analytical.

The paper describes the process of designing, implementing, and testing a certificate generation application. UML was employed to visualize the system, including use case diagrams, activities, sequence, and deployment.

The system provides the ability to create, edit, and delete certificate templates, as well as generate certificates for course participants.

Thus, the developed certificate generation application automates the process of creating certificates based on the Moodle platform. This reduction in time and resources required for this task and increase in efficiency of educational processes is a notable benefit of this approach.

ЗМІСТ

Завдання на кваліфікаційну роботу.....	2
Реферат	4
Summary	5
Вступ.....	8
1 Технічне завдання	10
1.1 Терміни та визначення.....	10
1.1.1 Загальні терміни	10
1.1.2 Технічні терміни	10
1.2 Функціональні вимоги.....	11
1.2.1 Призначення і цілі створення системи	11
1.2.2 Загальні функціональні можливості системи	11
1.3 Нефункціональні вимоги.....	12
1.4 Опис предметної області	12
1.5 Опис системи	13
1.6 Огляд інструментів розробки.....	14
1.6.1 Moodle	14
1.6.2 PHP	15
1.6.3 MySQL.....	16
2 Проєктування.....	17
2.1 Використання UML під час розробки системи.....	17
2.2 Діаграма варіантів використання	17
2.2.1 Опис варіантів використання.....	21
2.3 Діаграма діяльності.....	28
2.4 Діаграма послідовності.....	31
2.5 Діаграма розгортання.....	32
3 Реалізація та тестування	34
3.1 Опис інструментів розробки	34

3.2 Основні класи системи	34
3.3 Тестування проєкту.....	35
3.4 Керівництво користувача	37
3.4.1 Рівень підготовки користувача.....	37
3.4.2 Налаштування додатку	37
3.4.3 Створення шаблону сертифікату.....	38
3.4.4 Створення сертифікату.....	44
3.4.5 Отримання сертифікату.....	46
3.4.6 Перевірка сертифікату.....	47
Висновки	49
Перелік посилань.....	50
Додаток А Основні класи системи	51
Додаток Б Тести.....	63
Додаток В Посилання на Git	73

ВСТУП

В сучасному світі сертифікати відіграють важливу роль, будучи підтвердженням кваліфікації та компетенції як в освітній сфері, так і в бізнес-середовищі. Вони можуть відкривати нові можливості для кар'єрного зростання та професійного розвитку. Проте, ручна генерація сертифікатів може бути трудомісткою та неефективною, особливо для організацій, які регулярно видають великі обсяги сертифікатів, наприклад, навчальних закладів або компаній, що проводять стажування. Ця рутинна робота вимагає значних затрат часу та ресурсів.

Тому розробка спеціалізованого сервісу для автоматизованої генерації сертифікатів є надзвичайно актуальною. Такий сервіс дозволить значно оптимізувати та пришвидшити процес створення сертифікатів, заощадивши час та ресурси організацій. Автоматизація цього процесу також допоможе підвищити ефективність та продуктивність роботи, а також забезпечити високу якість та уніфікований вигляд сертифікатів.

Виходячи з цього, було вирішено створити додаток, який був би простий у використанні та адаптивний у налаштуванні.

Актуальність дослідження: актуальність теми полягає в необхідності створення ефективного інструменту для автоматизації генерації сертифікатів, що дозволить зменшити час та ресурси, необхідні для цього завдання.

З огляду на це, можна виділити наступні цілі і задачі нашого дослідження.

Мета: розробити додаток генерації сертифікатів.

Задачі:

- сформулювати вимоги до додатку;
- спроектувати та побудувати архітектуру додатку;
- реалізувати додаток генерації сертифікатів;
- протестувати роботу додатку.

Об'єкт дослідження: процес розробки додатку генерації сертифікатів засобами CMS Moodle.

Предмет дослідження: CMS Moodle.

Методи дослідження: моделювання, проектування, програмний, аналітичний.

Перший розділ присвячено збору та аналізуванню вимог до додатку, огляду інструментарію і опису системи.

У другому розділі розглянуто етапи проектування додатку, наведено список та опис прецедентів, а також побудовано діаграми.

Третій розділ присвячено реалізації та тестуванню роботи додатку, наведено керівництво користувача, яке описує процес роботи з додатком генерації сертифікатів.

1 ТЕХНІЧНЕ ЗАВДАННЯ

1.1 Терміни та визначення

1.1.1 Загальні терміни

Система – модуль генерації сертифікатів на платформі Moodle.

Moodle – це вільне веборієнтоване програмне забезпечення з відкритим вихідним кодом, що дозволяє створювати інтерактивні онлайн-курси та організовувати освітній процес в Інтернеті.

ДВІ – Діаграма Варіантів Використання чи Use Case Diagram.

ДД – Діаграма Діяльності.

ДП – Діаграма Послідовності.

ДР – Діаграма Розгортання.

Користувач – людина, яка зареєстрована у системі та може взаємодіяти з функціоналом.

Адміністратор – користувач, який має повні права на взаємодію з системою.

Автор курсу – користувач, який має повні права на взаємодію з курсом.

Слухач курсу – користувач, який має обмежені права на взаємодію з курсом.

1.1.2 Технічні терміни

MySQL – відкрита система керування реляційними базами даних.

БД – база даних.

Модуль – структурна одиниця Moodle.

1.2 Функціональні вимоги

1.2.1 Призначення і цілі створення системи

Функціональне призначення системи – реалізувати можливість генерування сертифікатів на основі платформи Moodle.

Експлуатаційне призначення системи – система може експлуатуватися користувачами системи.

Мета створення системи – розробка додатку генерації сертифікатів.

1.2.2 Загальні функціональні можливості системи

Система має надавати користувачам такі можливості:

- перевіряти/переглядати/завантажувати сертифікат.

Система має надавати слухачам курсу такі можливості:

- отримання сертифіката.

Система має надавати авторам курсу такі можливості:

- створення/редагування/видалення/налаштування сертифіката;
- перегляд отримувачів сертифіката.

Система має надавати адміністраторам такі можливості:

- створення/редагування/видалення/перегляд шаблону сертифіката;
- додавання елемента сертифікату;
- налаштування параметрів сертифікату;
- завантаження шаблону сертифікату;
- усі права автора курсу.

1.3 Нефункціональні вимоги

Інтерфейс користувача: система повинна коректно відображати його на будь-якому пристрої, забезпечуючи зручність використання для різних користувачів.

Підтримка браузерів: система має підтримувати останні версії популярних браузерів, таких як Mozilla Firefox, Google Chrome, Safari, Microsoft Edge та Opera, гарантуючи сумісність та доступність для широкого кола користувачів.

Вимоги до продуктивності: система повинна завантажувати сторінки не довше ніж за 2 секунди, а генерація кожного сертифіката не має перевищувати 10 секунд, забезпечуючи швидку та ефективну роботу.

Вимоги до безпеки: система повинна обмежувати доступ до своїх функцій лише для авторизованих користувачів, захищаючи дані та функціональність від несанкціонованого використання.

1.4 Опис предметної області

Предметною областю є розробка додатку генерації сертифікатів. Дана система повинна надавати користувачам можливість згенерувати сертифікати, обравши шаблон.

Процес взаємодії з системою залежить від прав користувача. Для слухача курсу все обмежується виконанням вимог отримання сертифікату. Після цього він може згенерувати сертифікат.

Для автора курсу додаються можливості керування елементом «Сертифікат» та взаємодії зі списком отримувачів. Адміністратор в свою чергу має можливість створювати шаблони сертифікатів та підтягувати їх під відповідні елементи на курсах.

Процес генерації сертифікату/ів проходить наступним чином. Слухач курсу після виконання вимог обирає елемент «Сертифікат», система відображає інтерфейс взаємодії з даним елементом. Після цього користувач натискає на кнопку «Переглянути сертифікат», система генерує сертифікат на основі шаблону та відображає користувачі засобами браузера, з цього інтерфейсу користувач може завантажити сертифікати на пристрій.

1.5 Опис системи

В сучасному світі сертифікати відіграють важливу роль, будучи підтвердженням кваліфікації та компетенції як в освітній сфері, так і в бізнес-середовищі. Вони можуть відкривати нові можливості для кар'єрного зростання та професійного розвитку.

Однак ручна генерація сертифікатів може бути трудомісткою та неефективною, особливо для організацій, які регулярно видають великі обсяги сертифікатів, наприклад, навчальних закладів або компаній, що проводять стажування. Ця рутинна робота вимагає значних затрат часу та ресурсів.

Тому розробка спеціалізованого сервісу для автоматизованої генерації сертифікатів є надзвичайно актуальною. Такий сервіс дозволить значно оптимізувати та пришвидшити процес створення сертифікатів, заощадивши час та ресурси організацій. Автоматизація цього процесу також допоможе підвищити ефективність та продуктивність роботи, а також забезпечити високу якість та уніфікований вигляд сертифікатів.

Можливості системи наступні:

- керування шаблонами;
- керування сертифікатами;
- генерація сертифікатів;
- перевірка сертифікатів.

1.6 Огляд інструментів розробки

1.6.1 Moodle

Moodle (Modular Object-Oriented Dynamic Learning Environment) – це потужна система керування навчанням, яка широко використовується у сфері освіти та корпоративного навчання. Moodle є вільним веборієнтованим програмним забезпеченням з відкритим вихідним кодом, що дозволяє створювати інтерактивні онлайн-курси та організувати освітній процес в Інтернеті [6, 9].

Розглянемо основні функції та можливості Moodle [3, 4].

Створення та управління навчальними курсами – викладачі можуть створювати структуровані курси з різноманітними матеріалами, завданнями, тестами та активностями.

Інтерактивне навчання – підтримка різних форматів контенту, таких як текст, відео, аудіо, презентації, а також інтерактивні елементи, такі як форуми, чати, вікторини та опитування.

Оцінювання та відстеження прогресу – викладачі можуть оцінювати роботи студентів, переглядати їх прогрес та активність на курсах, а студенти можуть відстежувати власні результати.

Гнучка система управління користувачами – можливість створювати різні ролі та налаштовувати права доступу для викладачів, студентів та адміністраторів.

Інтеграція з іншими системами – Moodle може інтегруватися з різними зовнішніми системами, такими як системи автентифікації, бази даних та інші додатки через стандартні протоколи обміну даними.

Мобільний доступ – наявність мобільних додатків для Android та iOS, що дозволяє користувачам отримувати доступ до курсів та навчального контенту з мобільних пристроїв [1, 6].

Moodle широко використовується в школах, університетах, корпораціях та різних організаціях по всьому світу для організації дистанційного навчання, змішаного навчання, а також для підтримки традиційних очних курсів та семінарів.

1.6.2 PHP

PHP (Hypertext Preprocessor) – популярна серверна мова програмування, що широко використовується для розробки вебдодатків.

PHP може бути вбудований безпосередньо у HTML-код, що дозволяє легко створювати динамічні вебсторінки. Однією з ключових переваг PHP є її кросплатформеність – вона підтримується на різних операційних системах, включаючи Windows, Linux та macOS [2, 8].

PHP є вільним відкритим програмним забезпеченням, що дозволяє розробникам модифікувати та розповсюджувати його відповідно до ліцензії. Мова має вбудовану підтримку для роботи з різними системами управління базами даних, такими як MySQL, PostgreSQL, Oracle і т.д. Завдяки великій і активній спільноті розробників [8].

PHP забезпечується багатьма бібліотеками, фреймворками та ресурсами для навчання та підтримки [5].

PHP вважається досить ефективною мовою програмування, здатною обробляти великі обсяги запитів та даних [5].

Ще однією перевагою є безкоштовність PHP, що робить її доступною для широкого кола розробників та організацій. Мова широко використовується для створення вебсайтів, систем управління контентом, вебдодатків для електронної комерції, порталів та багатьох інших типів веборієнтованих програм, завдяки простоті синтаксису, кросплатформеності, багатій функціональності та потужній підтримці спільноти.

1.6.3 MySQL

MySQL – це популярна відкрита система керування реляційними базами даних.

MySQL використовує стандартну мову структурованих запитів SQL для створення, маніпулювання та запитів даних. Вона є вільним програмним забезпеченням з відкритим вихідним кодом, що дозволяє користувачам безкоштовно використовувати, модифікувати та розповсюджувати її відповідно до умов ліцензії [7].

MySQL доступна для різних операційних систем, включаючи Windows, Linux, macOS та інші. Вона розроблена з акцентом на високу продуктивність і здатність масштабуватися для обробки великих обсягів даних та підтримки багатьох користувачів одночасно.

MySQL підтримує паралельну обробку запитів за допомогою багатопоточності, що підвищує продуктивність системи. Крім того, MySQL забезпечує функції репліки даних для підвищення доступності та резервного копіювання для забезпечення безпеки даних [7].

MySQL широко використовується у вебдодатках, системах управління контентом, системах електронної комерції, а також у багатьох інших програмах, які вимагають зберігання та керування даними. Її популярність обумовлена високою продуктивністю, надійністю, безкоштовністю та підтримкою багатьох платформ.

2 ПРОЄКТУВАННЯ

2.1 Використання UML під час розробки системи

UML (Unified Modeling Language) є візуальною мовою моделювання, призначеною для специфікації, візуалізації, документування та побудови архітектури об'єктно-орієнтованих програмних систем.

На етапі моделювання вимог діаграми прецедентів використання допомагають візуалізувати функціональні вимоги до системи, взаємодію між акторами (користувачами або зовнішніми системами) та самою системою [10].

Для моделювання аналізу застосовуються діаграми класів, діаграми послідовності та діаграми комунікації, які визначають концептуальну структуру системи, класи та їхні взаємозв'язки. На етапі проєктування системи діаграми компонентів та діаграми розгортання використовуються для візуалізації логічної та фізичної архітектури відповідно [10].

UML діаграми є корисними для документування різних аспектів системи, полегшуючи розуміння та підтримку коду. Більше того, UML забезпечує загальну мову для спілкування між різними зацікавленими сторонами (розробниками, аналітиками, клієнтами) на різних етапах розробки. Деякі інструменти можуть генерувати код безпосередньо зі специфікацій UML, що прискорює процес розробки.

Застосування UML під час розробки системи має свої переваги, включаючи кращу візуалізацію, полегшення комунікації, підвищення якості та зменшення ризиків через поліпшену документацію та моделювання.

2.2 Діаграма варіантів використання

Діаграма варіантів використання (Use Case Diagram) є одним з найважливіших типів діаграм в UML. Вона використовується для графічного

зображення функціональних вимог системи з точки зору зовнішніх акторів, які взаємодіють із системою. Діаграма варіантів використання допомагає візуалізувати та документувати різні сценарії використання системи, а також межі самої системи [10].

На діаграмі варіантів використання зображуються актори та варіанти використання (use cases). Актори представляють ролі або сутності, які взаємодіють з системою, такі як користувачі, зовнішні системи або пристрої. Варіанти використання – це послідовності дій або функції, які система повинна забезпечити для досягнення певної мети. Вони визначають основні вимоги до поведінки системи з точки зору акторів [10].

Діаграма варіантів використання допомагає зрозуміти і проаналізувати вимоги на ранніх етапах розробки системи. Вона забезпечує високорівневе уявлення про функціональність системи та взаємодію з акторами. Діаграма може бути доповнена детальними специфікаціями для кожного варіанту використання, що описують сценарії та можливі альтернативні потоки виконання.

Переваги використання діаграми варіантів використання включають покращену комунікацію між зацікавленими сторонами, ефективніше планування та управління вимогами, а також полегшення тестування та верифікації системи. Діаграма допомагає зосередитися на потребах користувачів і забезпечити, щоб система задовольняла їхні вимоги.

На рисунку 2.1 представлена діаграма варіантів використання додатку генерації сертифікатів.

На діаграмі представлено акторів «Слухач курсу», «Автор курсу» та «Адміністратор», кожен з яких може взаємодіяти з системою згідно на різних рівнях. Також представлено актора «Користувач», який об'єднує спільні прецеденти всіх акторів.

Виділено 1 основний варіант використання – «Створити шаблон сертифікату». Після того, як адміністратор переходить до вкладки «Керування шаблонами» та натискає на кнопку «Створити шаблон», система відображає інтерфейс створення нового шаблону. Після заповнення необхідних полів та

додавання необхідних елементів, користувач натискає на кнопку «Зберегти», система зберігає шаблон сертифікату.

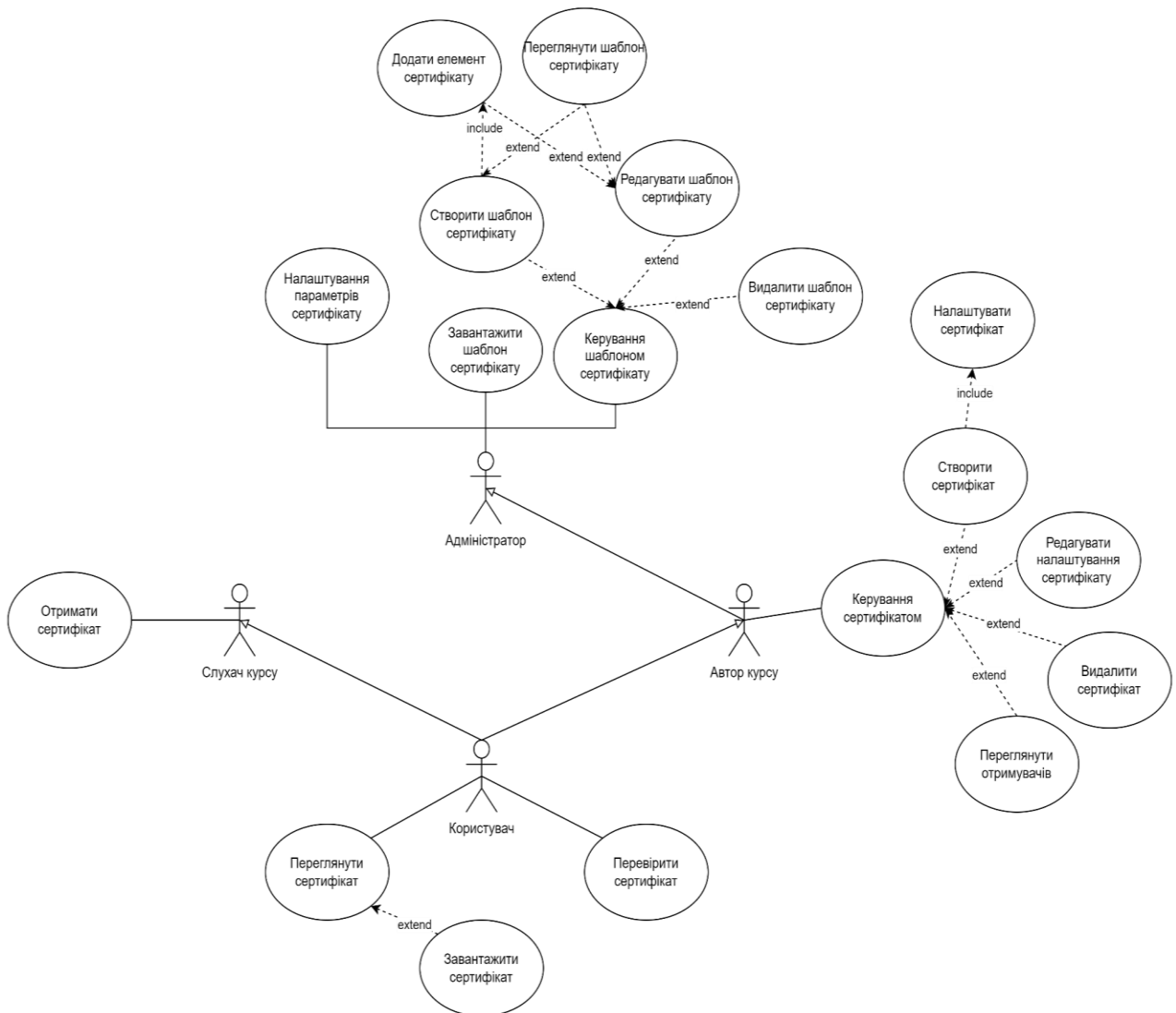


Рисунок 2.1 – Діаграма варіантів використання

Варіанти використання визначають функціональні можливості системи та способи їх використання кінцевими користувачами або зовнішніми акторами. Кожен варіант використання представляє певну послідовність дій, яку може виконати актор для досягнення конкретної мети або результату в рамках взаємодії з системою. Таким чином, варіанти використання відображають різні сценарії використання функціоналу системи, забезпечуючи засоби для моделювання та документування вимог до поведінки системи з точки зору акторів, які нею оперують (див. рис. 2.2 – 2.3).

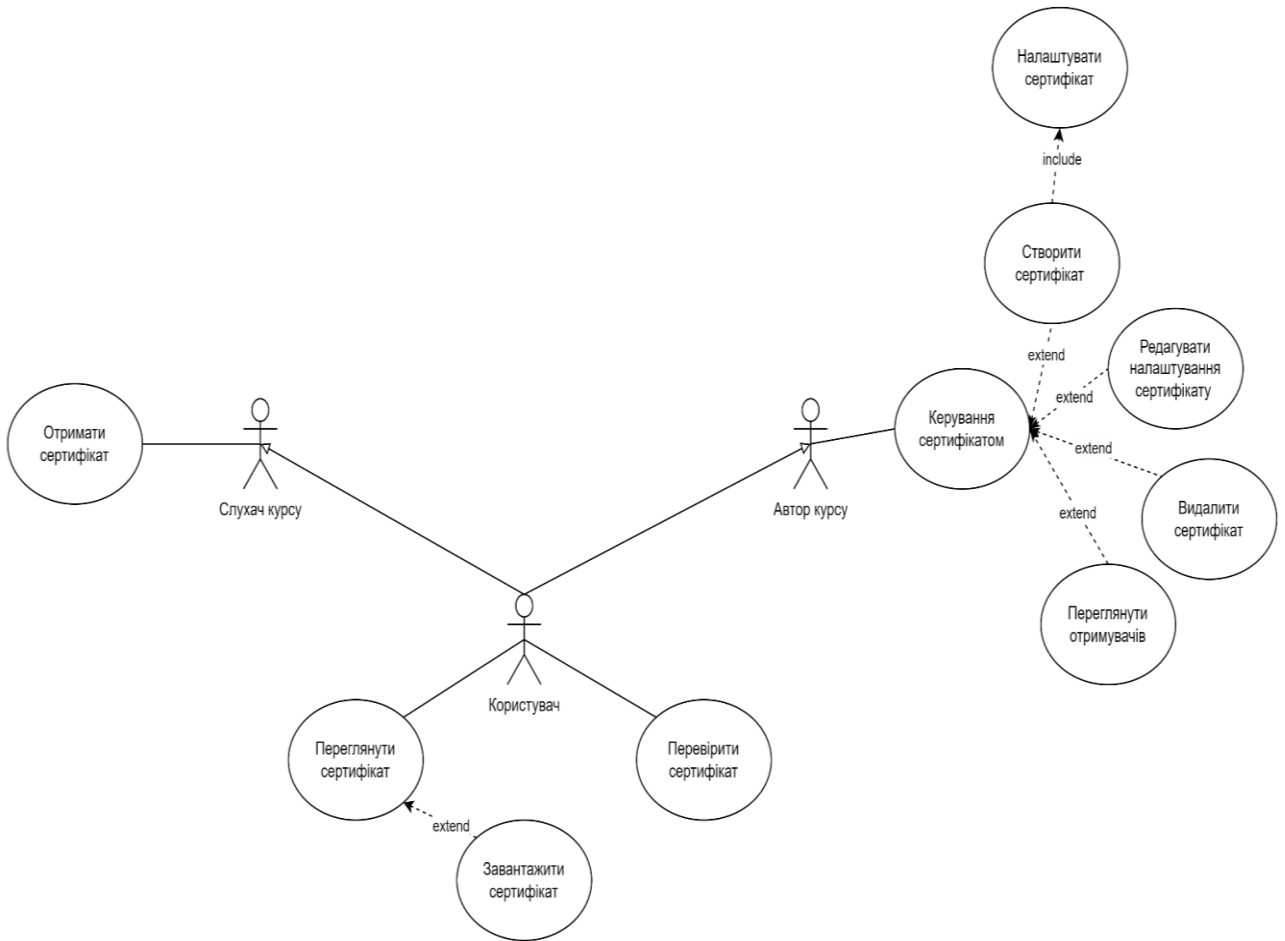


Рисунок 2.2 – Діаграма варіантів використання «Користувач», «Слухач курсу» і «Автор курсу»

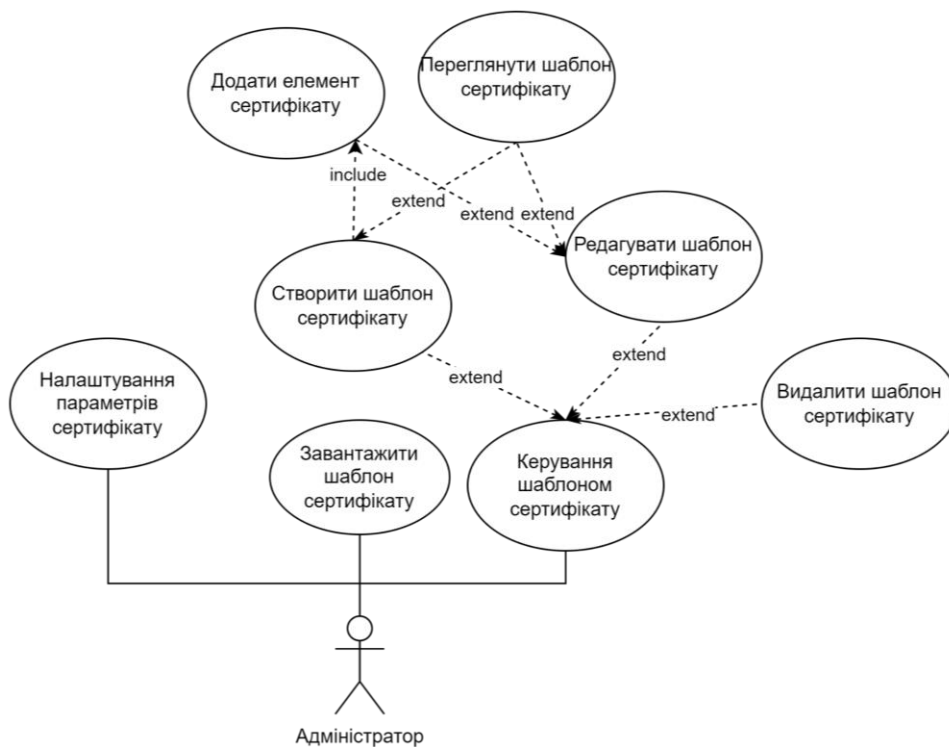


Рисунок 2.3 – Діаграма варіантів використання «Адміністратор»

2.2.1 Опис варіантів використання

Прецедент «Налаштування параметрів сертифікату».

Призначення: даний варіант використання надає можливість адміністратору налаштовувати параметри сертифікату, які будуть задавати поведінку всіх створених сертифікатів.

Основний потік подій: даний варіант використання починає виконуватися, коли адміністратор системи натискає на кнопку «Налаштування сертифікату». Система відображає інтерфейс налаштування сертифікату. Після задання необхідних параметрів, адміністратор натискає на кнопку «Зберегти», система зберігає параметри.

Передумова: перед початком виконання даного варіанта використання адміністратор повинен авторизуватися в системі.

Прецедент «Керування шаблоном сертифікату».

Призначення: даний варіант використання надає можливість адміністратору взаємодіяти з шаблонами сертифікатів.

Основний потік подій: даний варіант використання починає виконуватися, коли адміністратор системи натискає на кнопку «Керувати шаблонами». Система відображає список наявних шаблонів та інструменти взаємодії з ними.

Альтернативний потік подій: якщо жодного шаблону ще не було створено, система повідомляє про це у вигляді порожньої таблиці.

Передумова: перед початком виконання даного варіанта використання адміністратор повинен перейти в налаштування модуля сертифікату.

Прецедент «Створити шаблон сертифікату».

Призначення: даний варіант використання надає можливість адміністратору створювати нові шаблони сертифікатів.

Основний потік подій: даний варіант використання починає виконуватися, коли адміністратор переходить до вкладки «Керування шаблонами» та натискає на кнопку «Створити шаблон», система відображає інтерфейс створення нового

шаблону. Після заповнення полів, що відповідають за назву і розміри сертифікату, і додавання необхідних елементів, користувач натискає на кнопку «Зберегти», система зберігає шаблон сертифікату.

Передумова: перед початком виконання даного варіанта використання адміністратор повинен перейти в налаштування модуля сертифікату.

Вияткова ситуація 1: адміністратор натиснув кнопку «Скасувати» – система відображає сторінку керування шаблонами сертифікатів.

Вияткова ситуація 2: адміністратор ввів невалідні дані – система сповіщає про це, і адміністратор має можливість ввести дані знов.

Вияткова ситуація 3: сервер повернув помилку – система сповіщає про це, і дані не зберігаються.

Прецедент «Додати елемент сертифікату».

Призначення: даний варіант використання надає можливість адміністратору додавати елементи до сертифікату.

Основний потік подій: даний варіант використання починає виконуватися, коли адміністратор системи обирає елемент зі списку і натискає на кнопку «Додати елемент» під час створення або редагування шаблону, система відображає інтерфейс створення та налаштування нового елемента. Після введення необхідних даних, адміністратор натискає на кнопку «Зберегти зміни», система зберігає дані.

Передумова: перед початком виконання даного варіанта використання адміністратор повинен знаходитися на сторінці створення або редагування шаблону сертифікату.

Вияткова ситуація 1: адміністратор натиснув кнопку «Скасувати» – система відображає сторінку створення або редагування шаблону сертифікату.

Вияткова ситуація 2: адміністратор ввів невалідні дані – система сповіщає про це, і адміністратор має можливість ввести дані знов.

Вияткова ситуація 3: сервер повернув помилку – система сповіщає про це, і дані не зберігаються.

Прецедент «Видалити шаблон сертифікату».

Призначення: даний варіант використання надає можливість адміністратору видалити шаблон сертифікату.

Основний потік подій: даний варіант використання починає виконуватися, коли адміністратор системи натискає на піктограму кошика у рядку відповідного шаблону, система видаляє дані.

Передумова: перед початком виконання даного варіанта використання повинно бути створено принаймні один шаблон.

Виняткова ситуація 1: сервер повернув помилку – система сповіщає про це, і шаблон не видаляється.

Прецедент «Редагувати шаблон сертифікату».

Призначення: даний варіант використання надає можливість адміністратору редагувати шаблон сертифікату.

Основний потік подій: даний варіант використання починає виконуватися, коли адміністратор системи натискає на піктограму коліщатка у рядку відповідного шаблону, система відображає інтерфейс редагування шаблону сертифікату. Після внесення змін, адміністратор натискає кнопку «Зберегти», система зберігає оновлені дані та відображає сторінку керування шаблонами сертифікатів.

Передумова: перед початком виконання даного варіанта використання повинно бути створено принаймні один шаблон.

Виняткова ситуація 1: адміністратор натиснув кнопку «Скасувати» – система відображає сторінку керування шаблонами сертифікатів.

Виняткова ситуація 2: адміністратор ввів невалідні дані – система сповіщає про це, і адміністратор має можливість ввести дані знов.

Виняткова ситуація 3: сервер повернув помилку – система сповіщає про це, і дані не зберігаються.

Прецедент «Переглянути шаблон сертифікату».

Призначення: даний варіант використання надає можливість адміністратору переглядати попередній вигляд сертифікату.

Основний потік подій: даний варіант використання починає виконуватися, коли адміністратор системи натискає на кнопку «Збережіть зміни та попередній перегляд» під час створення або редагування шаблону, система відображає сертифікат.

Передумова: перед початком виконання даного варіанта використання адміністратор повинен знаходитися на сторінці створення або редагування шаблону сертифікату.

Прецедент «Завантажити шаблон сертифікату».

Призначення: даний варіант використання надає можливість адміністратору встановлювати шаблон для сертифікатів, що створені на курсі.

Основний потік подій: даний варіант використання починає виконуватися, коли адміністратор переходить до редагування будь-якого сертифікату, створеного на курсі. Система відображає інтерфейс створення сертифікату «з нуля» та пропонує обрати один із наявних шаблонів. Після вибору необхідного шаблону, адміністратор натискає на кнопку «Завантажити», система підтягує шаблон.

Альтернативний потік подій: якщо жодного шаблону ще не було створено, система не пропонує вибір шаблону.

Передумова: перед початком виконання даного варіанта використання повинно бути створено принаймні один шаблон.

Прецедент «Керування сертифікатом».

Призначення: даний варіант використання надає можливість автору курсу та адміністратору взаємодіяти з сертифікатом.

Основний потік подій: даний варіант використання починає виконуватися, коли автор курсу або адміністратор системи натискає на елемент сертифікату на сторінці курсу. Система відображає інтерфейс та інструментарій сертифікату.

Передумова: перед початком виконання даного варіанта використання адміністратор або автор курсу повинен створити елемент сертифікату на сторінці курсу.

Прецедент «Створити сертифікат».

Призначення: даний варіант використання надає можливість адміністратору та автору курсу створювати нові сертифікати.

Основний потік подій: даний варіант використання починає виконуватися, коли адміністратор переходить на сторінку курсу та створює елемент «Сертифікат», система відображає інтерфейс налаштування параметрів сертифікату, а після їх задання і збереження – інтерфейс та інструментарій елемента «Сертифікат».

Передумова: перед початком виконання даного варіанта використання адміністратор або автор курсу повинен знаходитися на сторінці відповідного курсу.

Виняткова ситуація 1: адміністратор або автор курсу натиснув кнопку «Скасувати» – система відображає сторінку курсу.

Виняткова ситуація 2: сервер повернув помилку – система сповіщає про це, і сертифікат не створюється.

Прецедент «Налаштувати сертифікат».

Призначення: даний варіант використання надає можливість адміністратору або автору курсу налаштовувати параметри сертифікату.

Основний потік подій: даний варіант використання починає виконуватися, коли адміністратор або автор курсу створює елемент «Сертифікат» на сторінці курсу. Система відображає інтерфейс налаштування сертифікату. Після задання необхідних параметрів, адміністратор або автор курсу натискає на кнопку «Зберегти», система зберігає параметри.

Передумова: перед початком виконання даного варіанта використання адміністратор або автор курсу повинен знаходитися на сторінці відповідного курсу.

Виняткова ситуація 1: адміністратор або автор курсу натиснув кнопку «Скасувати» – система відображає сторінку курсу.

Прецедент «Видалити сертифікат».

Призначення: даний варіант використання надає можливість адміністратору або автору курсу видалити сертифікат з курсу.

Основний потік подій: даний варіант використання починає виконуватися, коли адміністратор або автор курсу обирає пункт «Видалити» у меню редагування елемента «Сертифікат». Система видаляє дані.

Передумова: перед початком виконання даного варіанта використання повинно бути створено принаймні один сертифікат на сторінці відповідного курсу.

Вияткова ситуація 1: сервер повернув помилку – система сповіщає про це, і сертифікат не видаляється.

Прецедент «Редагувати сертифікат».

Призначення: даний варіант використання надає можливість адміністратору або автору курсу редагувати сертифікат.

Основний потік подій: даний варіант використання починає виконуватися, коли адміністратор або автор курсу натискає на пункт «Редагувати сертифікат» або «Редагувати параметри. Відповідно до наданих прав цим користувачам доступний різний набір налаштувань. Після внесення змін, адміністратор або автор курсу натискає на кнопку «Зберегти», система зберігає оновлені дані та відображає сторінку керування сертифікатом.

Передумова: перед початком виконання даного варіанта використання повинно бути створено принаймні один сертифікат.

Вияткова ситуація 1: адміністратор або автор курсу натиснув кнопку «Скасувати» – система відображає сторінку керування сертифікатом.

Вияткова ситуація 2: сервер повернув помилку – система сповіщає про це, і дані не зберігаються.

Прецедент «Переглянути отримувачів».

Призначення: даний варіант використання надає можливість адміністратору або автору курсу переглядати список отримувачів сертифікату.

Основний потік подій: даний варіант використання починає виконуватися, коли адміністратор або автор курсу переходить до інтерфейсу сертифікату, система відображає список отримувачів.

Передумова: перед початком виконання даного варіанта використання повинно бути створено принаймні один сертифікат.

Прецедент «Отримати сертифікат».

Призначення: даний варіант використання надає можливість слухачу курсу отримати сертифікат.

Основний потік подій: даний варіант використання починає виконуватися, коли слухач курсу виконує вимоги для отримання сертифікату та натискає на елемент «Сертифікат» на сторінці курсу. Система відображає інтерфейс сертифікату.

Передумова: перед початком виконання даного варіанта використання повинно бути створено принаймні один сертифікат та виконані вимоги для його отримання.

Вияткова ситуація 1: слухач курсу не відповідає вимогам отримання сертифікату – система відображає вимоги та не дозволяє користувачу перейти до інтерфейсу сертифікату.

Прецедент «Перевірити сертифікат».

Призначення: даний варіант використання надає можливість користувачу перевірити підлітність сертифікату.

Основний потік подій: даний варіант використання починає виконуватися, коли користувач напряму переходить за посиланням перевірки сертифікату або обирає цю опцію в інтерфейсі відповідного сертифікату. Система відображає поле для введення номеру сертифікату. Після того як користувач ввів номер та натиснув на кнопку «Перевірити», система відображає повідомлення про наявність сертифікату з таким номером в базі даних.

Вияткова ситуація 1: номер сертифікату не знайдено – система відображає повідомлення «Не підтверджено».

Прецедент «Переглянути сертифікат».

Призначення: даний варіант використання надає можливість користувачу переглянути сертифікат відповідно до наданих прав.

Основний потік подій: даний варіант використання починає виконуватися, коли користувач переходить до інтерфейсу сертифікату відповідного курсу. Система відображає кнопку «Переглянути сертифікат» для слухачів курсу та

список отримувачів для адміністратора та автора курсу з можливістю переглянути сертифікат кожного зі списку.

Передумова: перед початком виконання даного варіанта використання повинно бути створено принаймні один сертифікат та виконані вимоги для його отримання.

Вияткова ситуація 1: слухач курсу не відповідає вимогам отримання сертифікату – система відображає вимоги та не дозволяє користувачу перейти до інтерфейсу сертифікату.

Прецедент Завантажити сертифікат».

Призначення: даний варіант використання надає можливість користувачу завантажити сертифікат.

Основний потік подій: даний варіант використання починає виконуватися, коли користувач натискає на кнопку «Переглянути сертифікат» для слухачів курсу або натискає на піктограму завантаження відповідного сертифікату зі списку отримувачів для адміністратора та автора курсу. Система відображає сертифікат засобами браузера та надає можливість його завантажити на пристрій користувача

Передумова: перед початком виконання даного варіанта використання повинно бути створено принаймні один сертифікат та виконані вимоги для його отримання.

2.3 Діаграма діяльності

Діаграма діяльності (Activity Diagram) в UML – це тип діаграми потоків керування, яка візуалізує послідовність кроків у певному процесі або алгоритмі, включаючи можливі розгалуження на підпроцеси, паралельну обробку та з'єднання окремих потоків виконання [10].

На діаграмі діяльності процес зображується у вигляді серії вузлів дій, з'єднаних стрілками, які показують порядок виконання. Діаграма починається з

початкового вузла і завершується кінцевим вузлом. Розгалуження потоку керування відображаються за допомогою ромбоподібних вузлів рішень і різних переходів з умовами.

Діаграми діяльності широко використовуються для моделювання бізнес-процесів, потоків робіт, алгоритмів та послідовностей операцій. Вони допомагають візуалізувати та краще зрозуміти логіку процесу, виявити можливі проблеми або вузькі місця.

Ці діаграми також корисні для документування, аналізу та оптимізації процесів, а також для комунікації між командами аналітиків, розробників та зацікавлених сторін під час проектування систем.

Наведемо діаграму діяльності, що описує модель поведінки варіанта використання «Створити шаблон сертифікату». Діаграма представлена на рисунках 2.4 – 2.6.

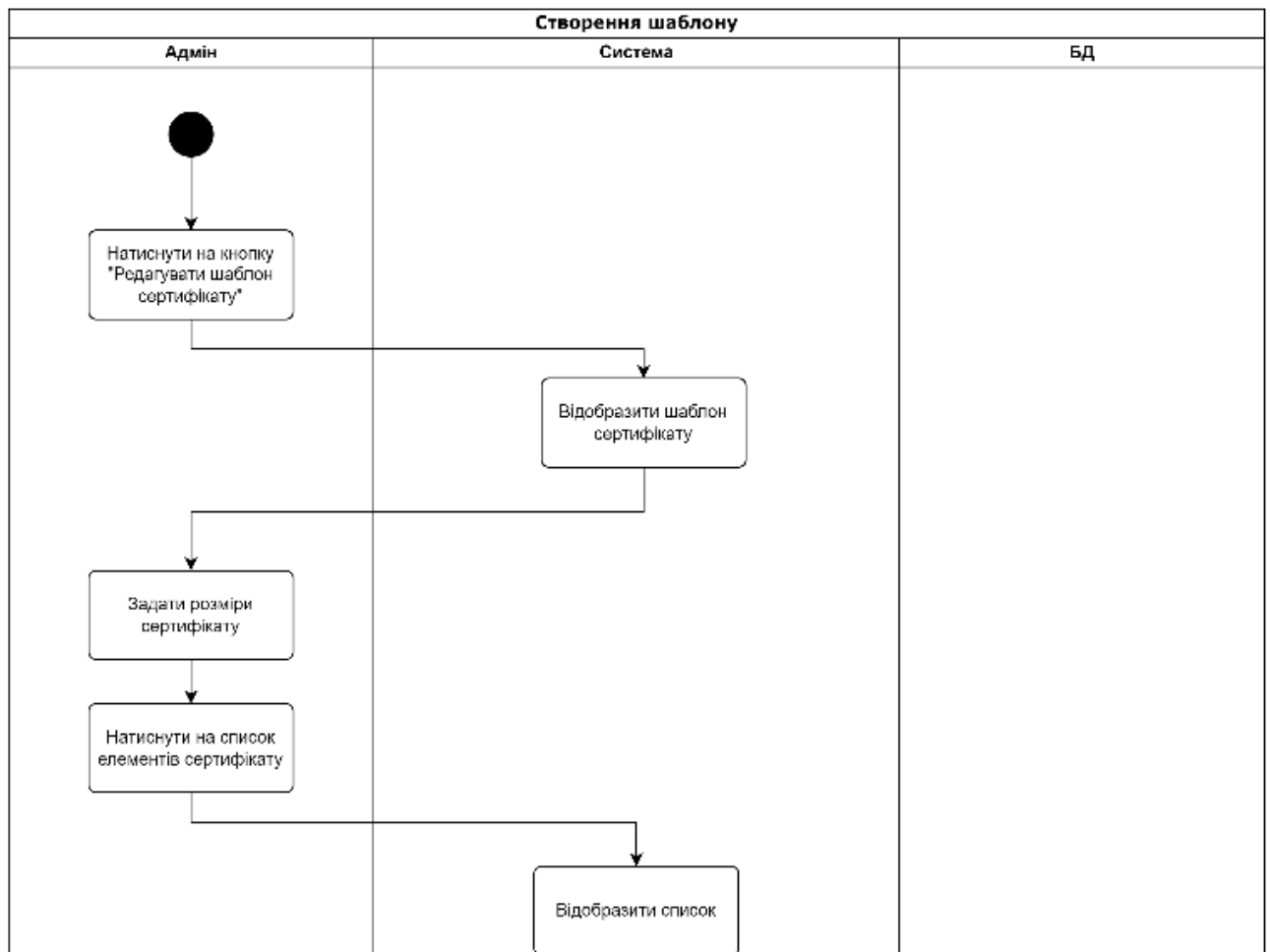


Рисунок 2.4 – Діаграма діяльності (1 частина)

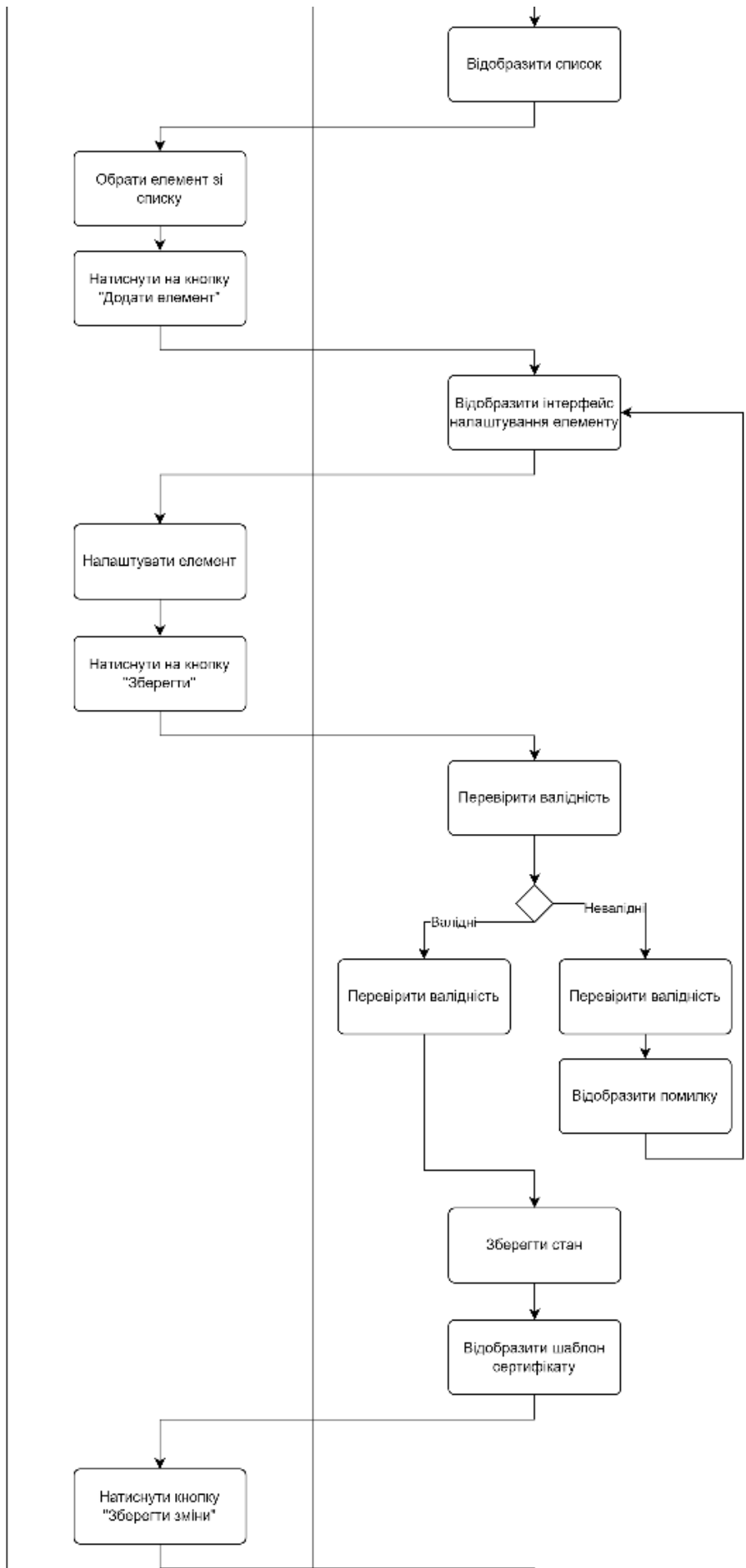


Рисунок 2.5 – Діаграма діяльності (2 частина)

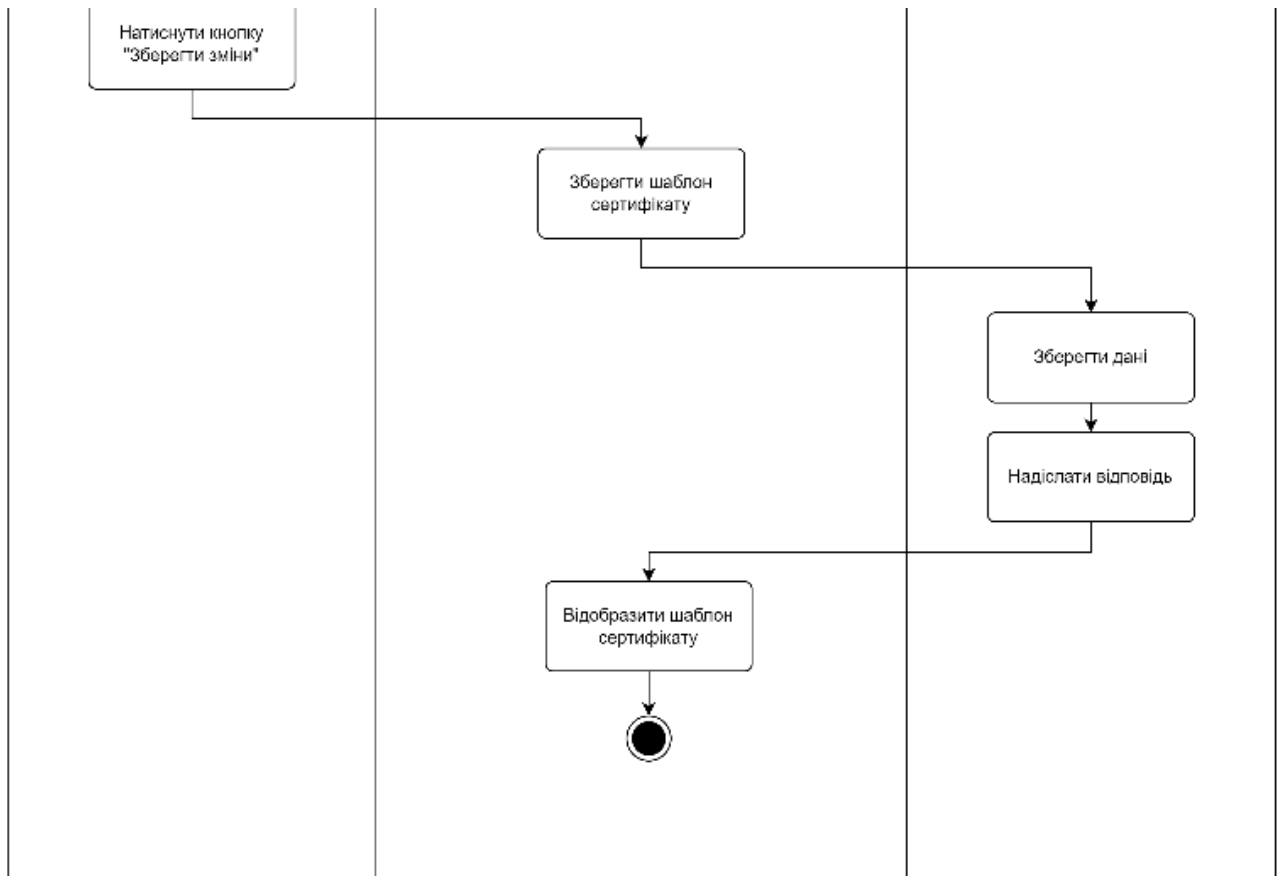


Рисунок 2.6 – Діаграма діяльності (3 частина)

2.4 Діаграма послідовності

Діаграма послідовності в UML – це різновид діаграми взаємодії, яка використовується для візуалізації послідовності повідомлень, що обмінюються між об’єктами або компонентами системи протягом певного сценарію виконання. Вона показує, як об’єкти взаємодіють один з одним у часі для реалізації певної функціональності [10].

Ці діаграми часто використовуються на етапах аналізу та проектування об’єктно-орієнтованих систем, а також під час розробки та документування складних сценаріїв взаємодії. Вони є цінним засобом для комунікації між розробниками, аналітиками та іншими зацікавленими сторонами.

На рисунку 2.7 описана діаграма послідовності прецедента «Створити шаблон сертифікату».

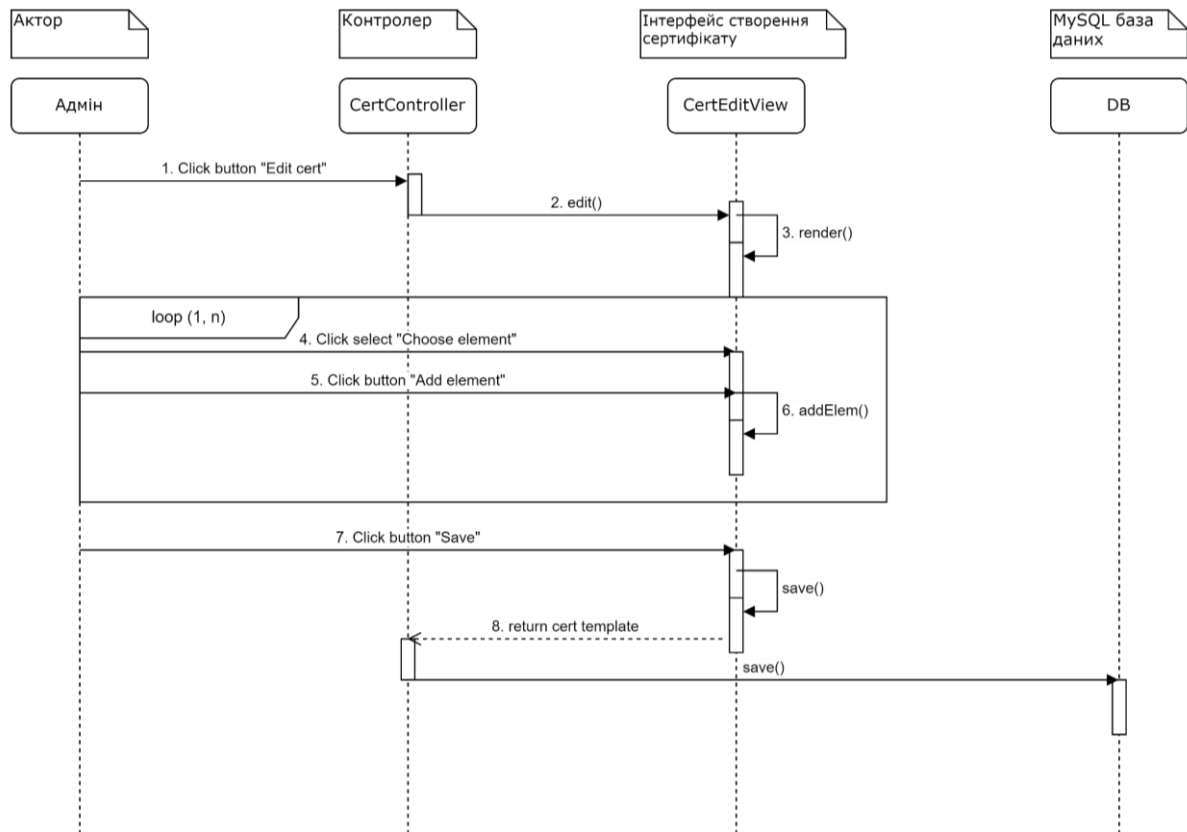


Рисунок 2.7 – Діаграма послідовності

2.5 Діаграма розгортання

Діаграма розгортання (Deployment Diagram) в UML призначена для візуалізації фізичної архітектури системи та відображення залежностей між апаратними та програмними компонентами. Вона показує, як артефакти програмного забезпечення розгортаються на різних вузлах, таких як сервери, пристрої або інші фізичні компоненти. Діаграма розгортання допомагає зрозуміти топологію системи та взаємозв'язки між її складовими частинами з точки зору фізичного розміщення [10].

Діаграма розгортання забезпечує високорівневе бачення системи з точки зору розподілу її компонентів на різних вузлах. Вона допомагає планувати та документувати процес розгортання, а також виявляти потенційні проблеми, пов'язані з масштабованістю, продуктивністю або безпекою.

На рисунку 2.8 наведено діаграму розгортання системи.

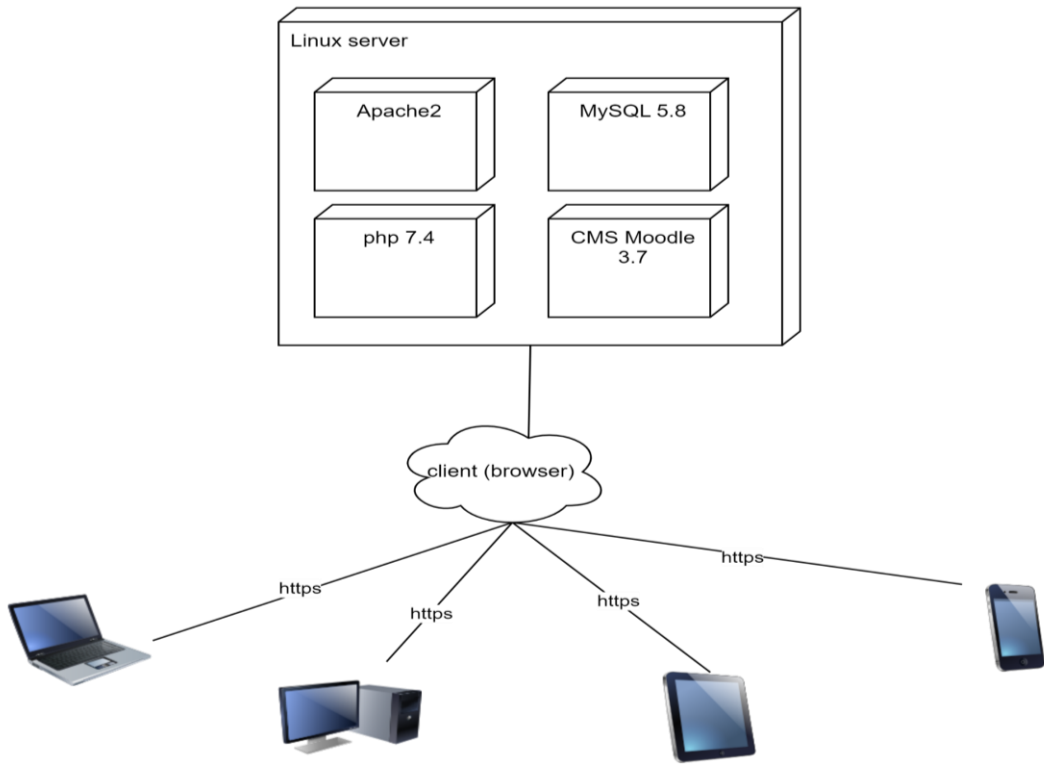


Рисунок 2.8 – Діаграма розгортання

3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ

3.1 Опис інструментів розробки

Для реалізації було використано платформу Moodle та серверну мову програмування php, для зберігання даних – відкрита система управління базами даних MySQL.

Також при розробці використовувалися вбудовані бібліотеки платформи Moodle, на кшталт тестування або роботи із зображеннями.

3.2 Основні класи системи

Оскільки основним прецедентом системи є «Створення шаблону сертифікату», то основними класами системи будуть ті, що надають можливість створювати, змінювати, видаляти та переглядати дані.

Нижче приведено приклад функції генерації унікального коду сертифікату (рис. 3.1).

```
public static function generate_code() {  
    global $DB;  
  
    $uniquecodefound = false;  
    $code = random_string(10);  
    while (!$uniquecodefound) {  
        if (!$DB->record_exists('customcert_issues', array('code' => $code))) {  
            $uniquecodefound = true;  
        } else {  
            $code = random_string(10);  
        }  
    }  
  
    return $code;  
}
```

Рисунок 3.1 – Приклад функції

Головними класами є:

- `certificate` – забезпечує функціональність, необхідну для діяльності, пов'язаної з сертифікатами;
- `template` – відображає шаблон користувацького сертифікату;
- `element` – абстрактний клас для елементів сертифікату;
- `element_helper` – містить корисні функції, пов'язані з елементами.

Детально ознайомитися з кодом основних класів можна в додатку А, а з повним кодом проєкту в додатку В.

3.3 Тестування проєкту

Юніт-тестування (`unit testing`) – це процес тестування окремих модулів/компонентів програмного забезпечення на коректність їх функціонування.

Метою юніт-тестування є перевірка правильності роботи найменших частин програми (функцій, методів, класів) з метою зменшення ймовірності помилок у системі в цілому.

У Moodle використовується фреймворк `PHPUnit` для написання та запуску юніт-тестів [6, 9].

Для запуску юніт-тестів у локальному середовищі розробки Moodle передбачено спеціальний інструмент і командний рядок.

Це дозволяє автоматизувати процес тестування й інтегрувати його в цикл безперервної інтеграції та постійного розгортання (`CI/CD`).

Приклади такого тестування наведено на рисунках 3.2 – 3.3.

Детально ознайомитися з прикладами тестування можна в додатку Б, а з повним кодом проєкту, який містить всі тести – в додатку В.

```

/**
 * Tests we are returning the correct course id for an element in a course customcert activity.
 */
public function test_get_courseid_element_in_course_certificate() {
    global $DB;

    // Create a course.
    $course = $this->getDataGenerator()->create_course();

    // Create a custom certificate in the course.
    $customcert = $this->getDataGenerator()->create_module('customcert', array('course' => $course->id));

    // Get the template to add elements to.
    $template = $DB->get_record('customcert_templates', array('contextid' => context_module::instance($customcert->cmid)->id));
    $template = new \mod_customcert\template($template);

    // Add a page to the template.
    $pageid = $template->add_page();

    // Add an element to this page.
    $element = new \stdClass();
    $element->name = 'Test element';
    $element->element = 'testelement';
    $element->pageid = $pageid;
    $element->sequence = \mod_customcert\element_helper::get_element_sequence($element->pageid);
    $element->timecreated = time();
    $element->id = $DB->insert_record('customcert_elements', $element);

    // Confirm the correct course id is returned.
    $this->assertEquals($course->id, \mod_customcert\element_helper::get_courseid($element->id));
}

```

Рисунок 3.2 – Тестування element_helper

```

public function test_delete_issue_no_login() {
    global $DB;

    // Create a course.
    $course = $this->getDataGenerator()->create_course();

    // Create a custom certificate in the course.
    $customcert = $this->getDataGenerator()->create_module('customcert', ['course' => $course->id]);

    // Create two users.
    $student1 = $this->getDataGenerator()->create_user();
    $student2 = $this->getDataGenerator()->create_user();

    // Enrol them into the course.
    $this->getDataGenerator()->enrol_user($student1->id, $course->id);
    $this->getDataGenerator()->enrol_user($student2->id, $course->id);

    // Issue them both certificates.
    $i1 = \mod_customcert\certificate::issue_certificate($customcert->id, $student1->id);
    $i2 = \mod_customcert\certificate::issue_certificate($customcert->id, $student2->id);

    $this->assertEquals(2, $DB->count_records('customcert_issues'));

    // Try and delete without logging in.
    $this->expectException('require_login_exception');
    \mod_customcert\external::delete_issue($customcert->id, $i2);
}

```

Рисунок 3.3 – Тестування вебсервісів

3.4 Керівництво користувача

3.4.1 Рівень підготовки користувача

Для ефективного використання сайту користувач повинен мати відповідні навички та досвід. Користувачу необхідні базові знання з використання комп'ютера та веббраузера. Ознайомлення з інструкціями для користувачів допоможе легше розібратися у функціоналі сайту.

3.4.2 Налаштування додатку

При інтеграції додатку до платформи Moodle користувач може налаштувати його параметри, такі як: базові значення для всіх сертифікатів, дозвіл перевірки сертифікатів, тощо. Всі налаштування детально описані, що дозволяє легко зрозуміти призначення кожного пункту (рис. 3.4).

Налаштування власного сертифіката

Дозволити перевірку всіх сертифікатів
customcert | verify/certificate

Показати позиції X і Y
customcert | showposxy

Перевірити сертифікат
customcert | verify/certificate

Керування шаблонами
customcert | manage/templates

Завантажити зображення
customcert | upload/images

Типово: Ні

Якщо це налаштування ввімкнено, будь-яка людина (включаючи користувачів, які не увійшли в систему) може віддати посилання "http://moodle.local/mod/customcert/verify_certificate.php", щоб перевірити будь-який сертифікат на сайті, а не перейти до посилання для підтвердження для кожного сертифіката.

Примітка - це стосується лише сертифікатів, для яких у налаштуваннях сертифіката встановлено значення "Так" у пункті "Дозволити кожному перевірити сертифікат".

Типово: Ні

Це покаже позицію X і Y під час редагування елемента, що дозволяє користувачеві точно вказати місце розташування.

Це не обов'язково, якщо ви плануєте виключно використовувати інтерфейс переглядання для цієї мети.

[Перевірити сертифікат](#)

Це посилання перенесе вас на новий екран, де ви зможете перевірити сертифікати на сайті

[Керування шаблонами](#)

Це посилання приведе вас до нового екрану, на якому ви зможете керувати шаблонами, які використовуються користувальницькою діяльністю сертифікатів на курсах.

[Завантажити зображення](#)

Це посилання перенесе вас на новий екран, де ви зможете завантажувати зображення. Зображення, завантажені за допомогою цього методу, будуть доступні для всіх користувачів, які можуть створити сертифікат.

Типові значення для налаштування діяльності

Значення встановлені тут визначають типові установки, які будуть використовуватися при створенні вами нової діяльності. Ви також можете встановити, які параметри налаштування діяльності будуть вважатися розширеними.

Надіслати ел. пошту студентам
customcert | email/students

Надіслати ел. пошту вчителям
customcert | email/teachers

Надіслати ел. пошту іншим
customcert | email/others

Дозволити кожному перевірити сертифікат
customcert | verify

Типово: Ні

Якщо встановлено, це надішле студентам копію сертифіката, коли вона стане доступною.

Типово: Ні

Якщо це буде встановлено, буде надіслано вчителю копію сертифіката, коли вона стане доступною.

Типово: Порожнє

Якщо це буде встановлено, буде надіслано електронні адреси, вказані тут (розділені комою), з копією сертифіката, коли вона стане доступною.

Типово: Ні

Це налаштування дозволяє кожному, хто має посилання на підтвердження сертифіката (включаючи користувачів, які не увійшли в систему), перевірити сертифікат.

Рисунок 3.4 – Налаштування додатку

3.4.3 Створення шаблону сертифікату

Натиснувши на пункт «Керування шаблонами», користувач переходить до сторінки створення та керування шаблонами. Якщо жодного шаблону ще не було створено, то система відображає повідомлення «Нема чого показувати». Для створення нового шаблону, користувачу необхідно натиснути на кнопку «Створити шаблон» (рис. 3.5).

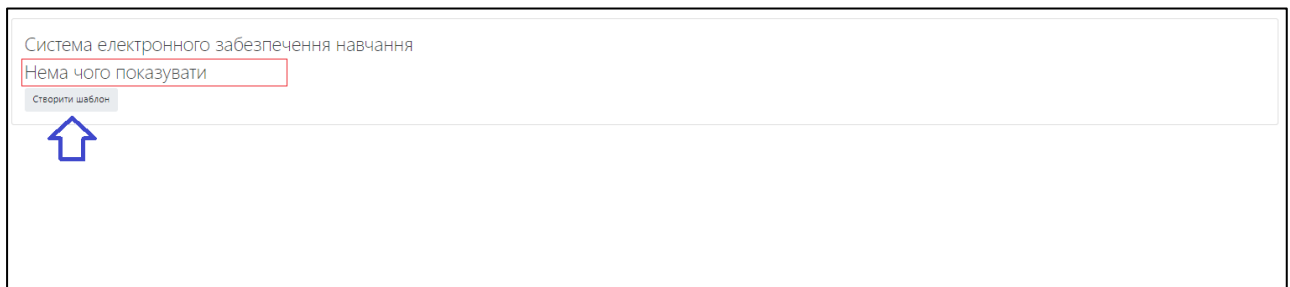


Рисунок 3.5 – Інтерфейс керування шаблонами

Система відображає інтерфейс створення шаблону, який можна логічно розділити на дві частини: основна інформація (назва, розміри, поля) і елементи (рис. 3.6).

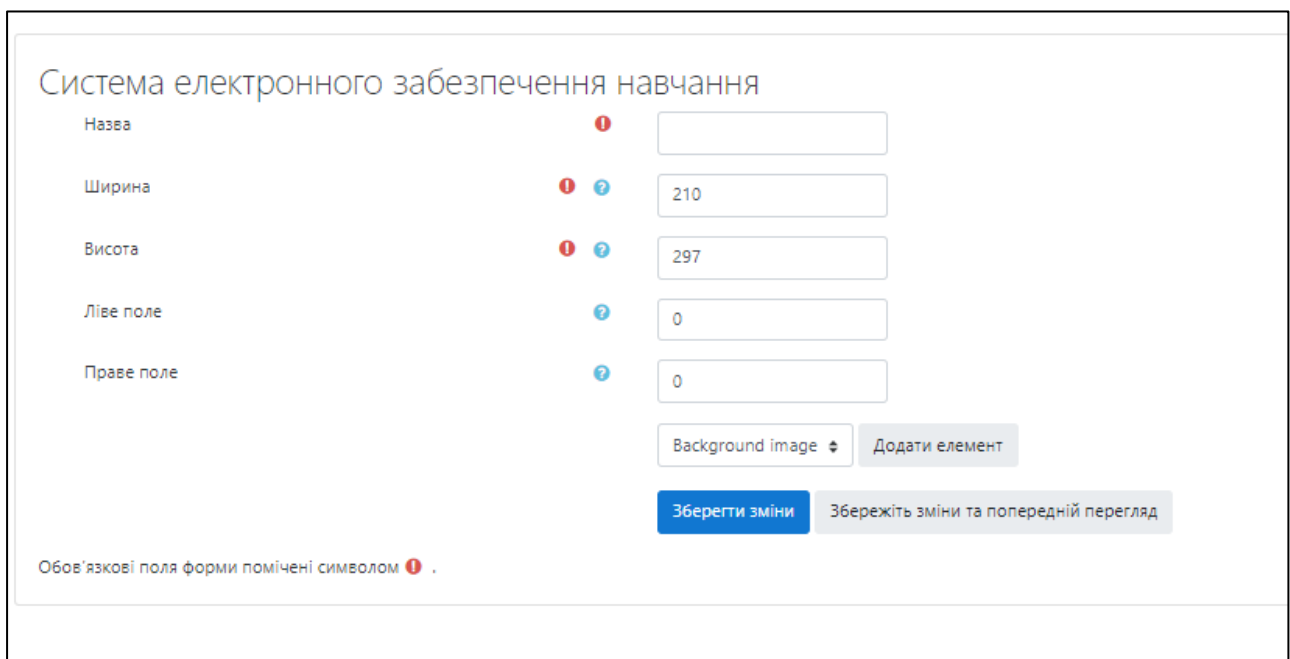
The screenshot shows the 'Create template' form in the 'Система електронного забезпечення навчання'. It contains several input fields: 'Назва' (Name) with a red error icon, 'Ширина' (Width) with a value of 210, 'Висота' (Height) with a value of 297, 'Ліве поле' (Left margin) with a value of 0, and 'Праве поле' (Right margin) with a value of 0. There are also buttons for 'Background image' (with a plus icon), 'Додати елемент' (Add element), 'Зберегти зміни' (Save changes), and 'Збережіть зміни та попередній перегляд' (Save changes and preview). A footer note states: 'Обов'язкові поля форми помічені символом i' (Required form fields are marked with the i symbol).

Рисунок 3.6 – Інтерфейс створення шаблону

Шаблон пропонує багато структурних елементів, які умовно можна поділити на наступні категорії: взаємодія із зображеннями, взаємодія із текстом та взаємодія із полями користувачів та курсів (рис. 3.7).

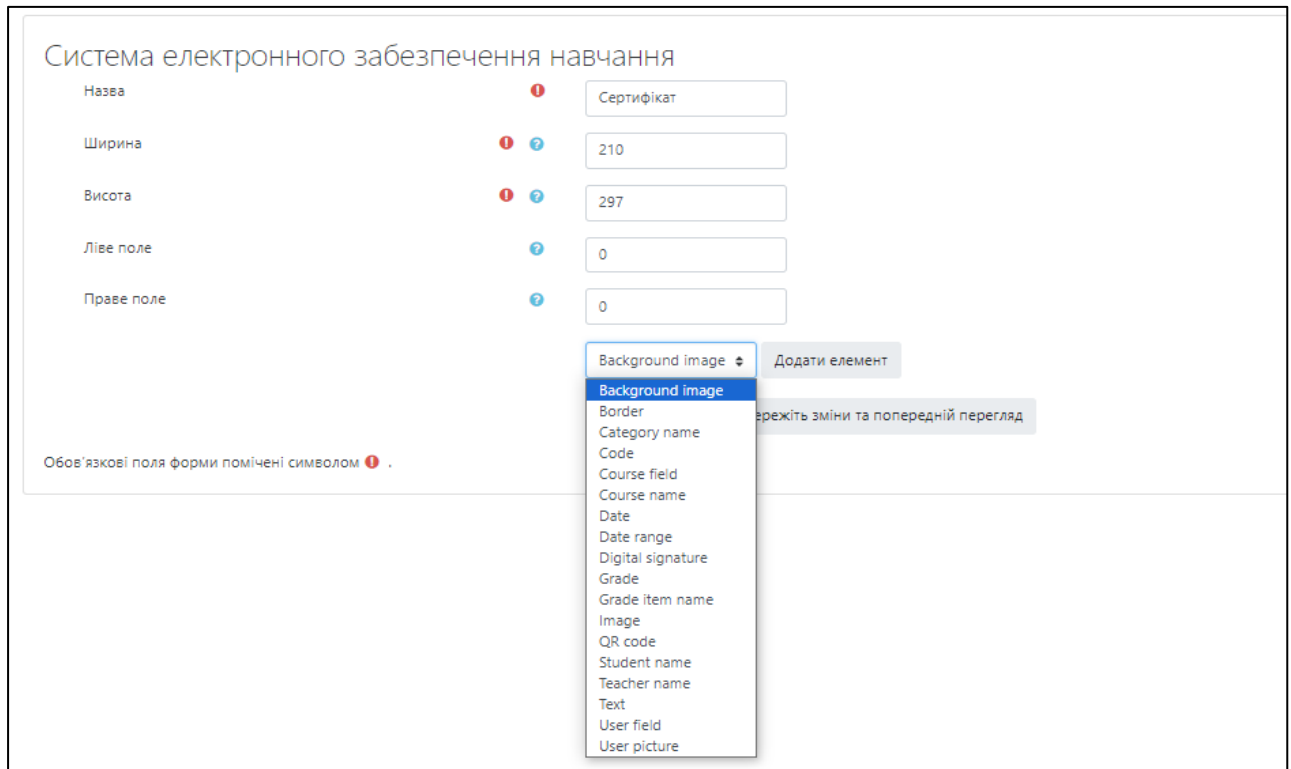


Рисунок 3.7 – Елементи шаблону

Опишемо детально кожен з елементів:

- Background image – відповідає за зображення фону;
- Border – відповідає за товщину границь;
- Category name – дозволяє використовувати категорію курсу на якому розташовано сертифікат;
- Code – генерує унікальний код з 10-ти символів;
- Course field/User field – дозволяє використовувати поля курсів (терміни, видимість, тощо) та користувачів (ПІБ, email, факультет, тощо);
- Course name/Student name/Teacher name – дозволяє отримати назву курсу/ПІБ отримувача сертифікату/ПІБ викладача курсу;
- Date/Date range – надає можливість взаємодіяти з датами курсу або його структурних одиниць. Також є можливість задавати дату власноруч;

- Grade/Grade item name – надає можливість взаємодіяти з оцінками курсу або його структурних одиниць;
- Image – відповідає за вставку зображення;
- Text – відповідає за вставку тексту.

Інтерфейс створення кожного з елементів має свій набір полів та налаштувань. На рисунку 3.8 наведено інтерфейс створення елемента «Background image».

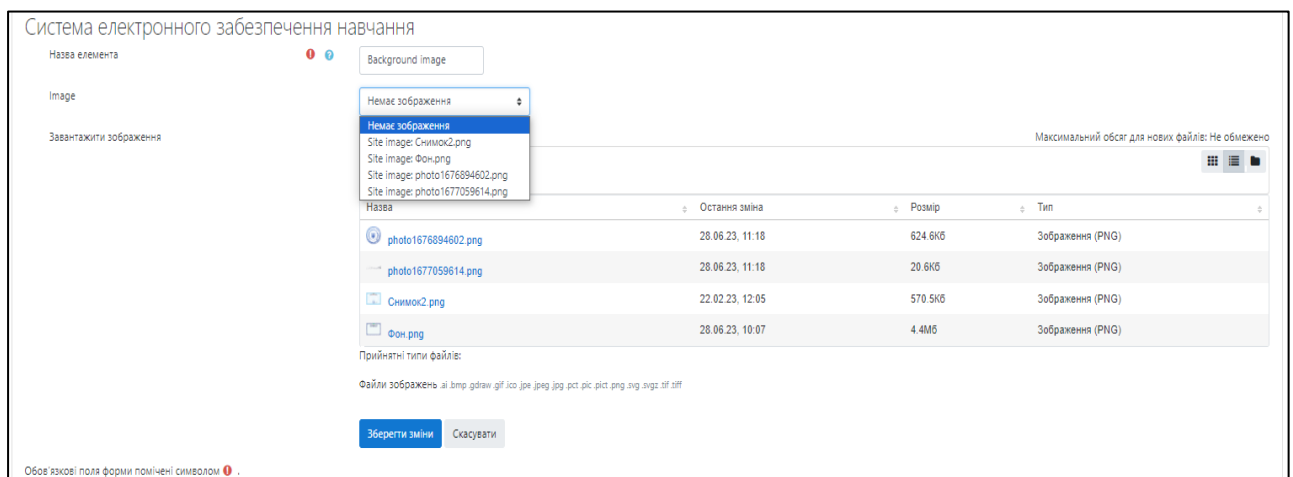


Рисунок 3.8 – Інтерфейс елемента «Background image»

Після того як користувач додав та налаштував елемент, він з'являється в інтерфейсі створення шаблону (рис. 3.9), де з ним можна взаємодіяти (редагувати, видалити, перемістити, перейменувати).

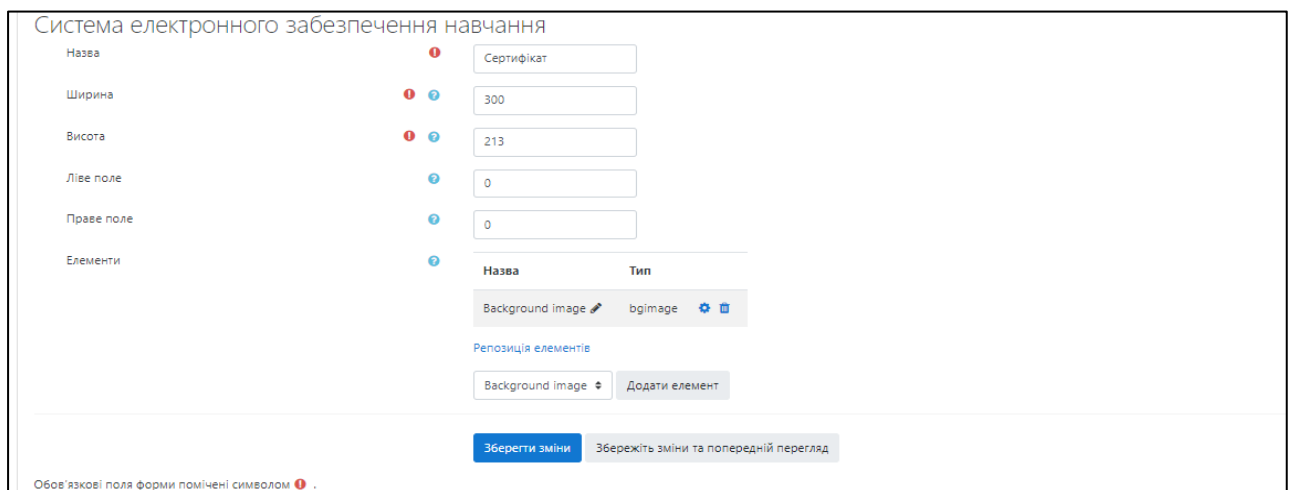


Рисунок 3.9 – Елемент «Background image»

Також щоб впевнитися, що елемент працює належним чином, користувач може використовувати функцію передперегляду (рис. 3.10).

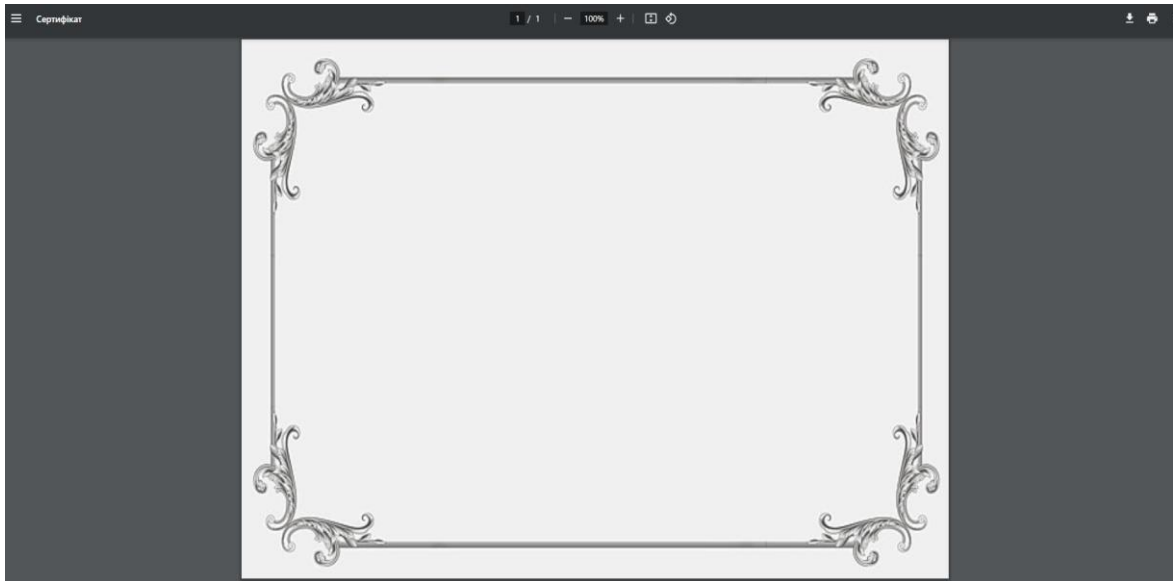


Рисунок 3.10 – Передперегляд сертифікату

Розглянемо взаємодію з елементом «Text», процес створення та налаштування схожі з попереднім елементом, за виключенням параметрів та полів. Для текстових елементів притаманні властивості розміру, кольору та шрифту тексту. Більш детально ознайомитися з інтерфейсом текстових елементів можна на рисунку 3.11.

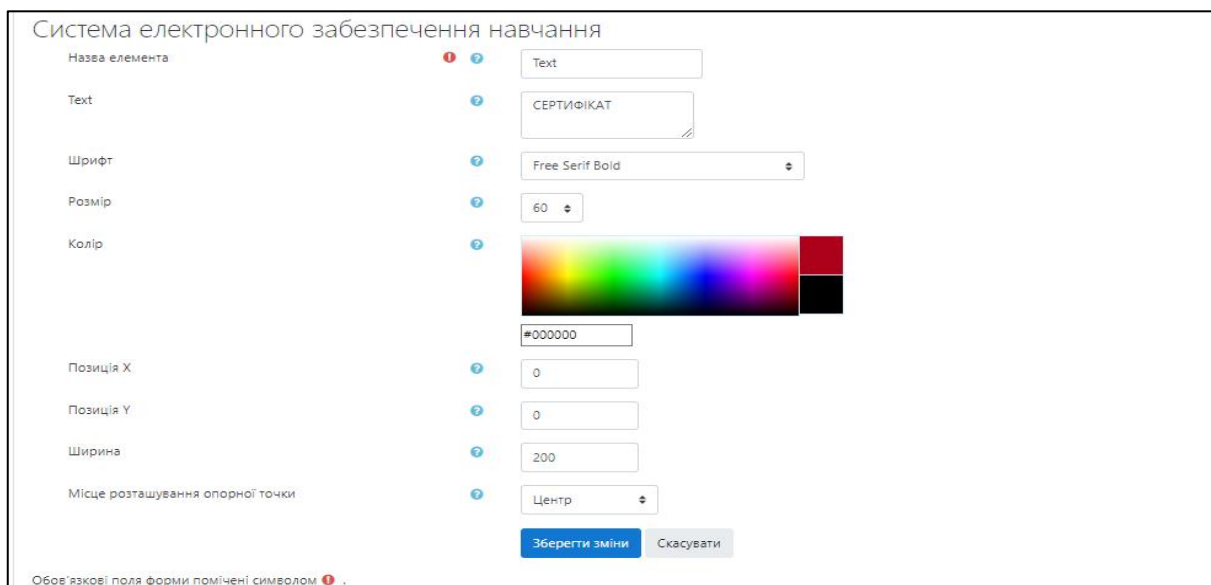


Рисунок 3.11 – Інтерфейс елемента «Text»

Після збереження змін, користувач також може перевірити відображення сертифікату. Оскільки при створенні елементу не було вказано параметрів розташування, то його буде розміщено у лівому верхньому куті (рис. 3.12).

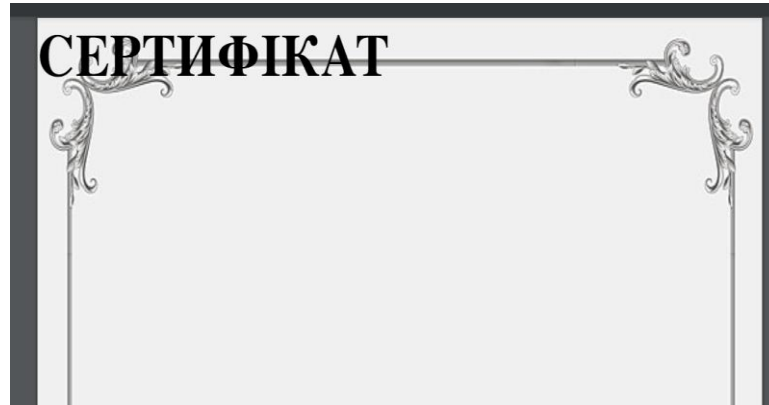


Рисунок 3.12 – Передперегляд з елементом «Text»

Для позиціонування елементу можна внести зміни у його налаштування, вказавши параметри розташування. Але це не дуже зручний варіант, краще скористатися функцією репозиції елементів, яка доступна з інтерфейсу створення шаблону. Для цього необхідно натиснути на пункт «Репозиція елементів». Система відображає інтерфейс, на якому, за допомогою функції переміщення елементів, можна зручно та швидко позиціонувати елементи відносно один одного (рис. 3.13).

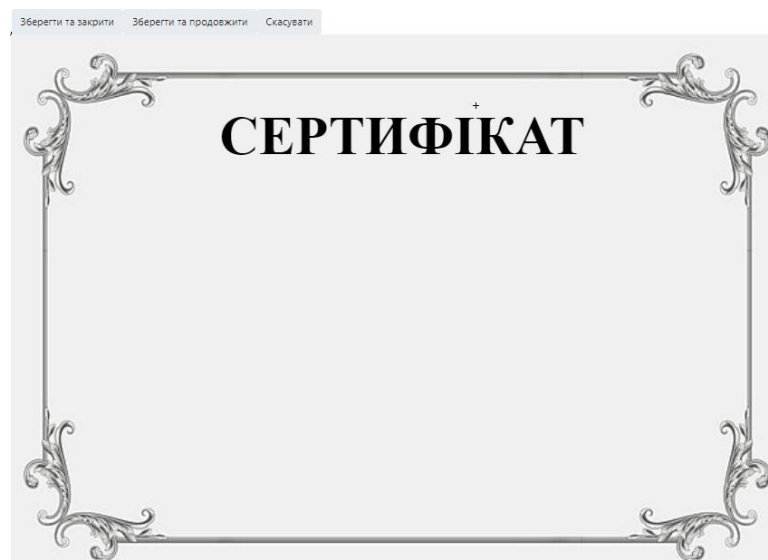


Рисунок 3.13 – Репозиція елементів

За позицію відповідає символ «+», тобто, якщо необхідно розташувати елемент по центру, то потрібно орієнтуватися саме по символу, а не значенню. Після збереження змін, користувач може знов використати функцію передперегляду, щоб впевнитися, що елемент розташовано правильно (рис. 3.14).

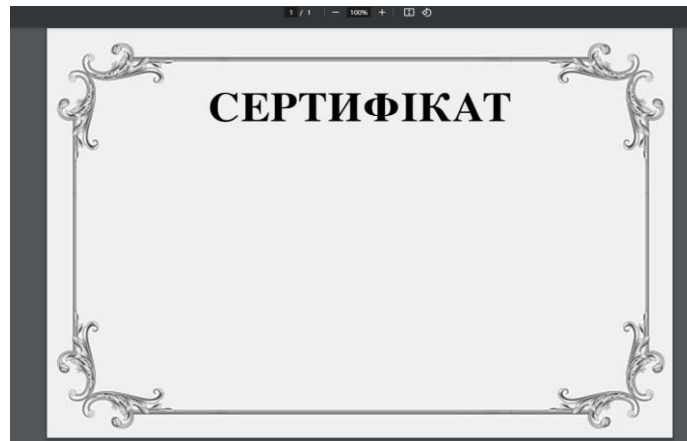


Рисунок 3.14 – Передперегляд після репозиції

Невід’ємною частиною будь-якого сертифікату є унікальний номер. Цей функціонал надає елемент «Code», який генерує код з 10-ти символів (рис. 3.15 – 3.16).

Система електронного забезпечення навчання

Назва елемента	<input type="text" value="Code"/>
Шрифт	<input type="text" value="Free Serif Bold Italic"/>
Розмір	<input type="text" value="30"/>
Колір	<input type="text" value="#000000"/>
Позиція X	<input type="text" value="150"/>
Позиція Y	<input type="text" value="80"/>
Ширина	<input type="text" value="200"/>
Місце розташування опорної точки	<input type="text" value="Центр"/>

Зберегти зміни Скасувати

Обов'язкові поля форми помічені символом **i**.

Рисунок 3.15 – Інтерфейс елементу «Code»



Рисунок 3.16 – Передперегляд з елементом «Code»

Для подальшого тестування створимо ще два елемента «Student name» і «Course name». Вони будуть необхідні для перевірки правильності підтягування назв поточного курсу та користувача (рис. 3.17).

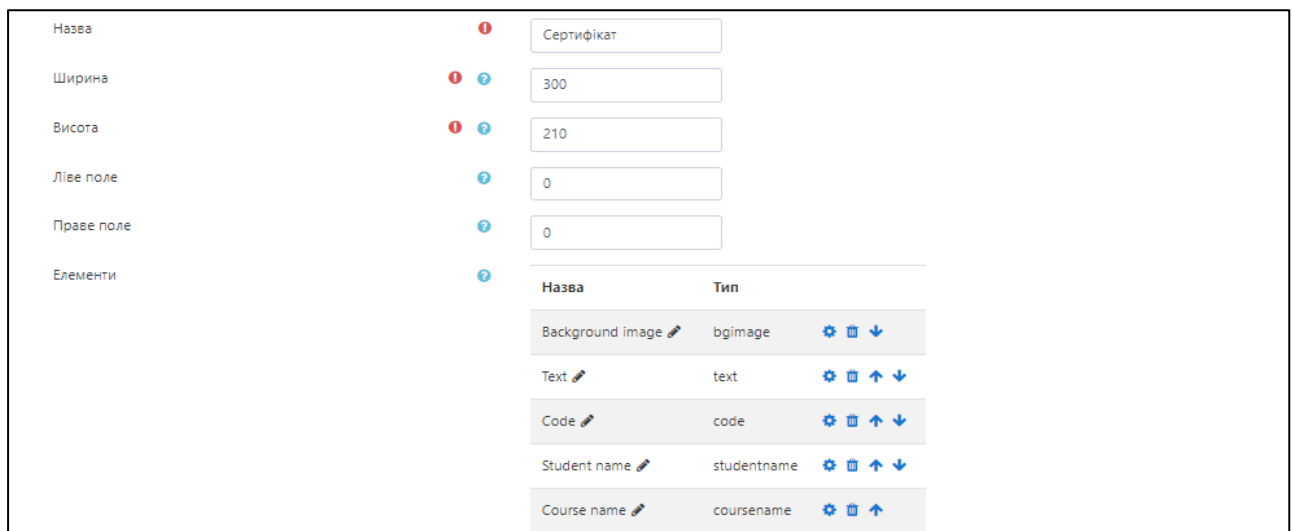


Рисунок 3.17 – Елементи «Student name» і «Course name»

3.4.4 Створення сертифікату

Щоб створити сертифікат, користувачу необхідно перейти до будь-якого курсу, де в нього є права «Автор курсу», та обрати компонент «Сертифікат» серед інших компонентів курсу. Сертифікат з'явиться на сторінці, як показано на рисунку 3.18.

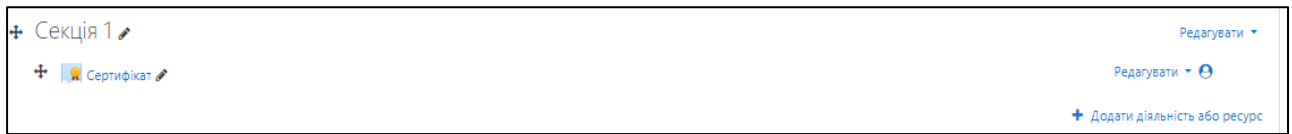


Рисунок 3.18 – Компонент «Сертифікат»

Далі необхідно перейти до налаштувань сертифікату та обрати один із наявних шаблонів. Залежно від налаштування прав у системі, ця дія може бути доступна виключно адміністраторам (рис. 3.19 – 3.20).

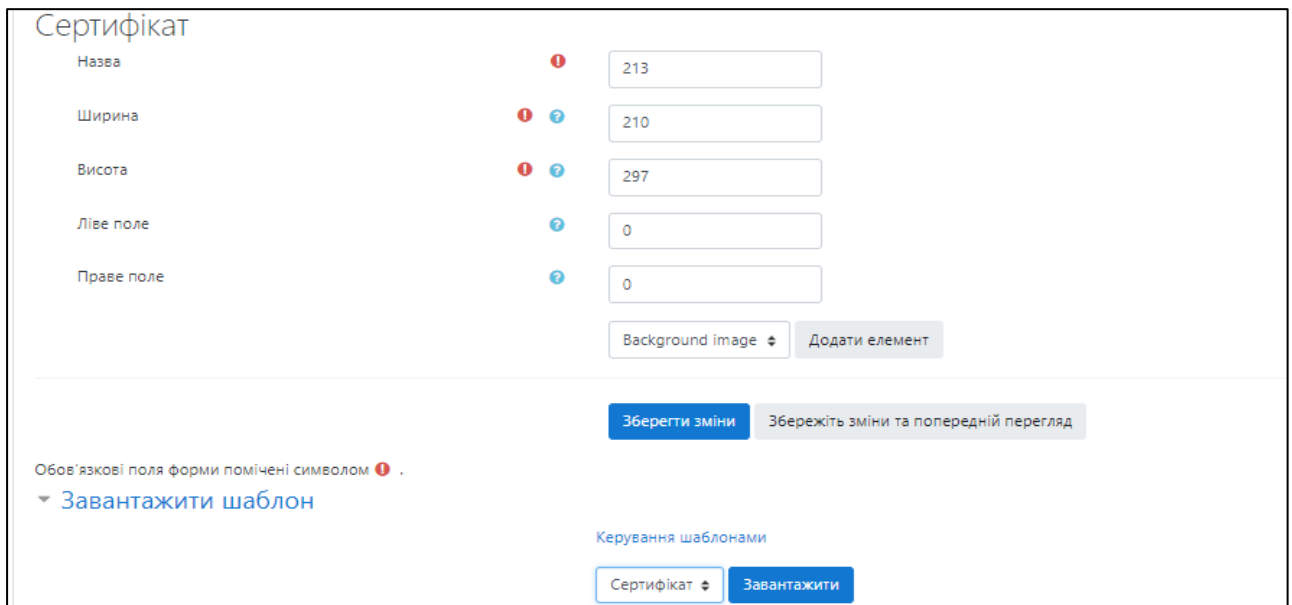


Рисунок 3.19 – Інтерфейс налаштування сертифікату

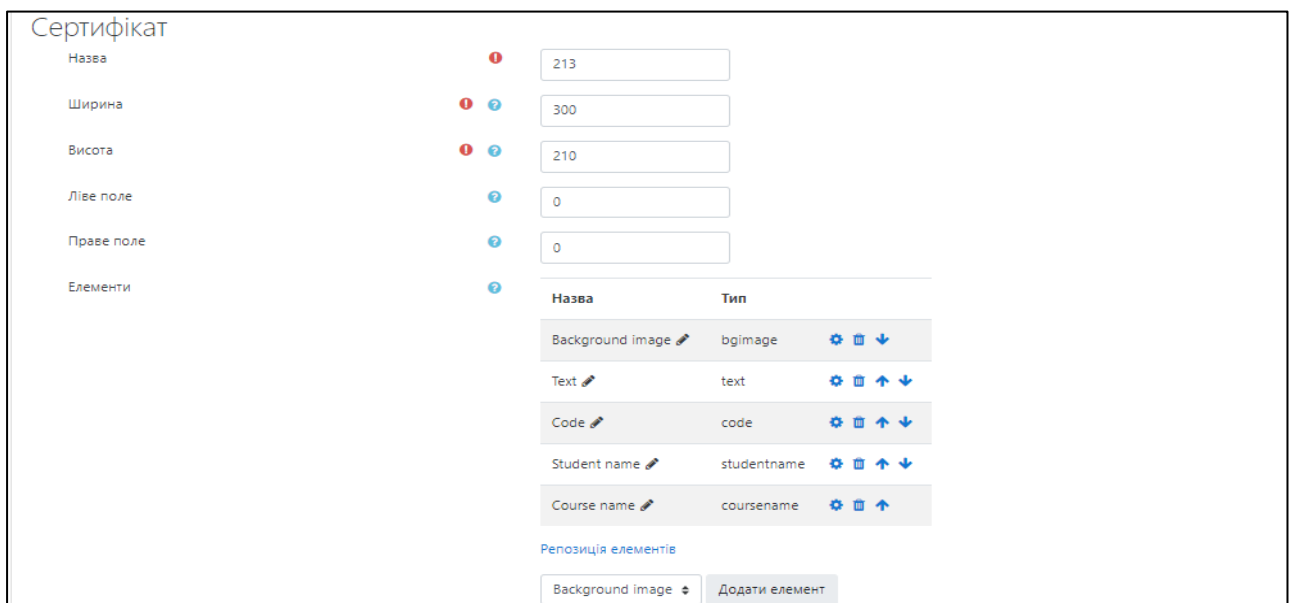


Рисунок 3.20 – Підтягнутий шаблон

Також при завантаженні або зміні шаблону, система буде запитувати у користувача, що він впевнений у виборі та що наявні налаштування будуть втрачені (рис. 3.21).

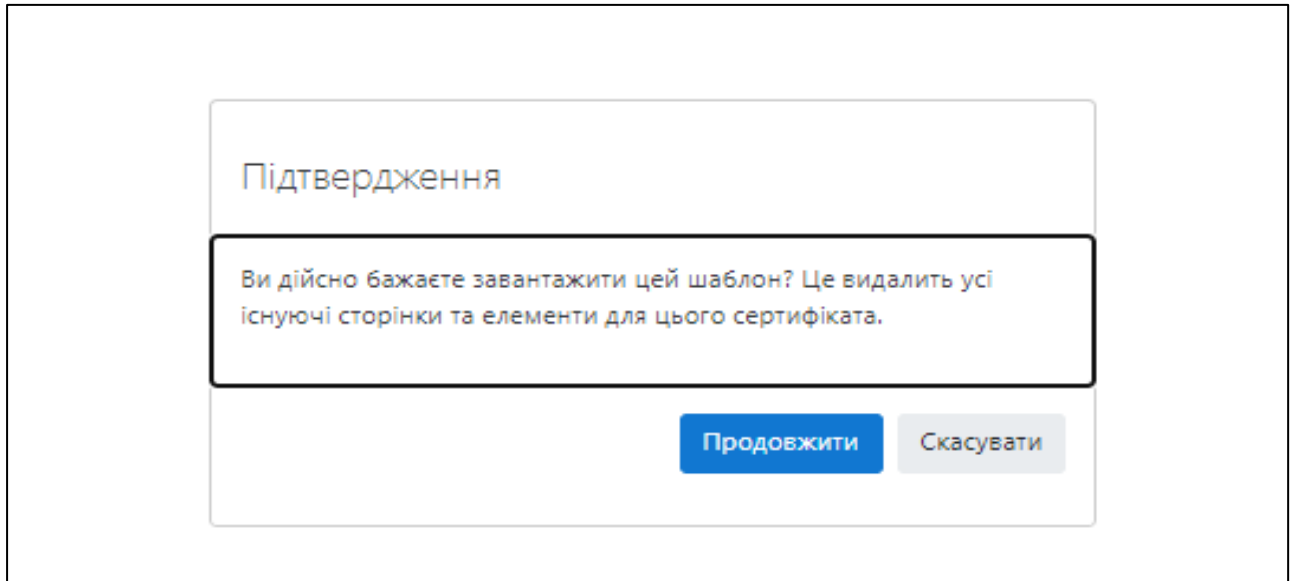


Рисунок 3.21 – Попередження

3.4.5 Отримання сертифікату

Кожний слухач курсу може отримати сертифікат за умови виконання вимог, якщо вони є. Для цього його повинно бути зараховано на відповідний курс з правами слухача курсу (студента) (рис. 3.22).

Вибрати	Прізвище / Ім'я	Електронна пошта	Заклад	Ролі	Групи	Останній вхід на курс	Статус
<input type="checkbox"/>	test test	test@test.test		Студент	Тест 1	20 години 49 хв	Активний

Рисунок 3.22 – Слухач курсу

Для того щоб отримати сертифікат, користувач натискає на кнопку «Переглянути сертифікат» в інтерфейсі компонента «Сертифікат». Система відображає його засобами браузера (рис. 3.23).



Рисунок 3.23 – Сертифікат слухача курсу

Також при виконанні вимог отримання сертифіката, автору курсу автоматично стає доступен список слухачів курсу, які виконали вимоги (рис. 3.24).

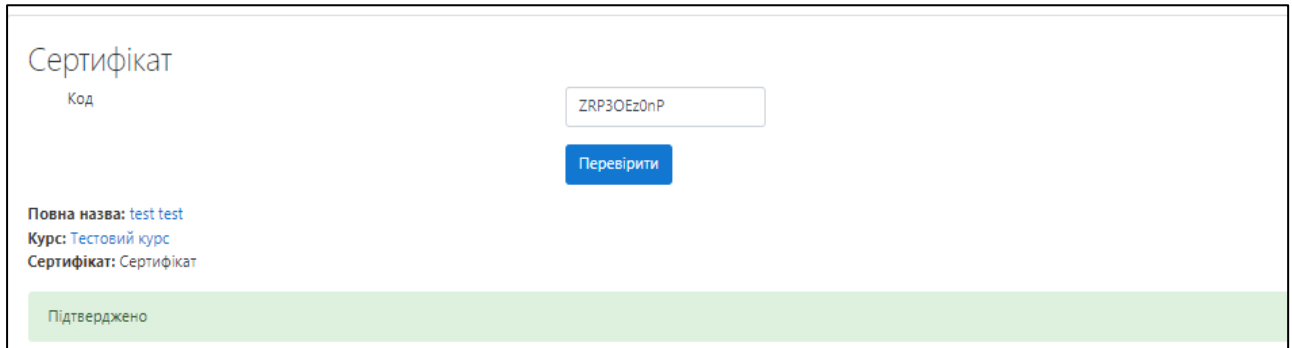
Прізвище / Ім'я	Електронна пошта	Заклад	Нагороджений	Код	Файл
test test	test@test.test		Thursday 2 May 2024 12:57 PM	ZRP3OEz0nP	Download

Рисунок 3.24 – Список слухачів курсу, які отримали сертифікат

3.4.6 Перевірка сертифікату

Оскільки у кожного сертифікату при створенні генерується унікальний номер, то відповідно його можна перевірити, щоб впевнитись, що він дійсний. Для цього необхідно перейти за посиланням, яке доступне з кожного компонента

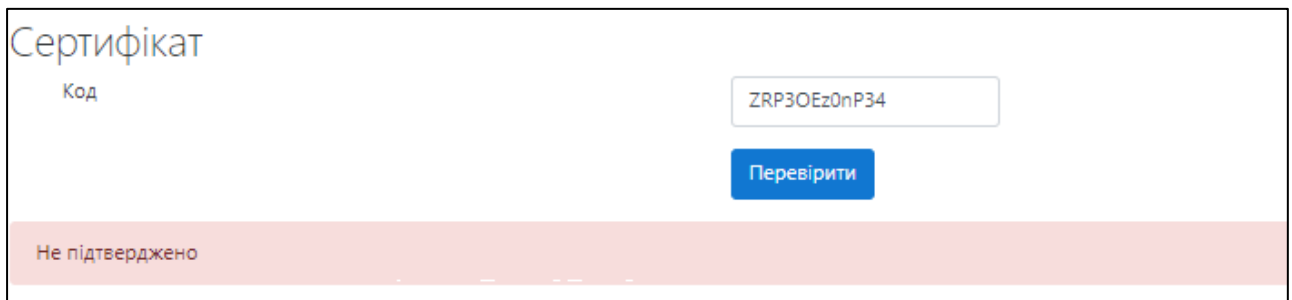
«Сертифікат», ввести номер у відповідне поле та натиснути на кнопку «Перевірити». Альтернативно сертифікат можна перевірити, якщо при його створенні використовувався елемент «QR code». Якщо сертифікат дійсний, то система відображає інформацію про курс та власника (рис. 3.25).



The screenshot shows a web interface for certificate verification. At the top left, the word "Сертифікат" (Certificate) is displayed. Below it, the label "Код" (Code) is positioned above a text input field containing the alphanumeric string "ZRP3OEz0nP". A blue button labeled "Перевірити" (Verify) is located below the input field. To the left of the button, the following information is displayed: "Повна назва: test test" (Full name: test test), "Курс: Тестовий курс" (Course: Test course), and "Сертифікат: Сертифікат" (Certificate: Certificate). At the bottom of the interface, a light green horizontal bar contains the text "Підтверджено" (Confirmed).

Рисунок 3.25 – Дійсний сертифікат

Якщо сертифікат відсутній в системі, то система відображає повідомлення «Не підтверджено» (рис. 3.26).



The screenshot shows a web interface for certificate verification. At the top left, the word "Сертифікат" (Certificate) is displayed. Below it, the label "Код" (Code) is positioned above a text input field containing the alphanumeric string "ZRP3OEz0nP34". A blue button labeled "Перевірити" (Verify) is located below the input field. At the bottom of the interface, a light red horizontal bar contains the text "Не підтверджено" (Not confirmed).

Рисунок 3.26 – Сертифікат не підтверджено

ВИСНОВКИ

У ході виконання даної роботи було розроблено систему автоматизованої генерації сертифікатів засобами CMS Moodle. Основні функціональні можливості системи включають створення, редагування та видалення шаблонів сертифікатів, а також генерацію сертифікатів для слухачів курсів. Система відповідає сучасним вимогам до вебдодатків, включаючи адаптивний інтерфейс, сумісність з популярними веббраузерами, високу продуктивність та забезпечення безпеки даних користувачів.

У відповідності з метою кваліфікаційної роботи було розроблено додаток генерації сертифікатів з використаннями наступних технологій:

- Moodle для управління курсами та користувачами;
- PHP для реалізації серверної частини додатку;
- MySQL для зберігання даних.

У відповідності з поставленими задачами були виконані наступні етапи створення додатку:

- сформовані вимоги до системи (функціональні та нефункціональні (інтерфейс, кросбраузерність, безпека, продуктивність)), а також приведено огляд інструментів розробки і предметної області;
- спроектована та побудована структура системи (побудовані діаграми прецедентів, діяльності, послідовності і розгортання; надано детальний опис кожного з прецедентів);
- реалізовано додаток генерації сертифікатів (наведено керівництво користувача та структуру проєкту);
- протестована робота системи.

ПЕРЕЛІК ПОСИЛАНЬ

1. Büchner A. Moodle 4 Administration – Fourth Edition: An administrator’s guide to configuring, securing, customizing, and extending Moodle. Birmingham : Packt Publishing, 2022. 640 p.
2. Daubois A., Windler C. Clean Code in PHP: Expert tips and best practices to write beautiful, human-friendly, and maintainable PHP. Birmingham : Packt Publishing, 2022. 264 p.
3. Ellis B. C. Introduction to Moodle 3.9+ Plugin Development: A comprehensive introduction to plugin development for Moodle, the world’s leading learning environment, for PHP developers. London : Mukudu Publishing, 2021. 280 p.
4. Koo J. Moodle for Learning Management System (LMS): A Practical and Visual Guidebook of Administrator and Instructor for Distance Education. Michigan : Independently published, 2020. 319 p.
5. Mitchell D. PHP Blueprint: An Essential Beginners Guide to Learn the Realms of PHP From A-Z. David Mitchell, 2023. 132 p.
6. Moodle Developer Documentation. URL: <https://moodledev.io/docs> (дата звернення: 06.03.2024).
7. MySQL Documentation. URL: <https://dev.mysql.com/doc/> (дата звернення: 08.03.2024).
8. PHP Documentation. URL: <https://www.php.net/docs.php> (дата звернення: 23.03.2024).
9. Postlewaite D. Moodle in Minutes: A Practical Introduction for New Online Educators. Michigan : Independently published, 2020. 64 p.
10. Unhelkar B. Software Engineering with UML. Boca Raton : Auerbach Publications, 2020. 426 p.

ДОДАТОК А

Основні класи системи

```
class certificate {

    /**
     * @var string the print protection variable
     */
    const PROTECTION_PRINT = 'print';

    /**
     * @var string the modify protection variable
     */
    const PROTECTION_MODIFY = 'modify';

    /**
     * @var string the copy protection variable
     */
    const PROTECTION_COPY = 'copy';

    /**
     * @var int the number of issues that will be displayed on each page in the report
     * If you want to display all customcerts on a page set this to 0.
     */
    const CUSTOMCERT_PER_PAGE = '50';

    /**
     * Handles setting the protection field for the customcert
     *
     * @param \stdClass $data
     * @return string the value to insert into the protection field
     */
    public static function set_protection($data) {
        $protection = array();

        if (!empty($data->protection_print)) {
```

```

    $protection[] = self::PROTECTION_PRINT;
}
if (!empty($data->protection_modify)) {
    $protection[] = self::PROTECTION_MODIFY;
}
if (!empty($data->protection_copy)) {
    $protection[] = self::PROTECTION_COPY;
}

// Return the protection string.
return implode(' ', $protection);
}

/**
 * Handles uploading an image for the customcert module.
 *
 * @param int $draftitemid the draft area containing the files
 * @param int $contextid the context we are storing this image in
 * @param string $filearea indentifies the file area.
 */
public static function upload_files($draftitemid, $contextid, $filearea = 'image') {
    global $CFG;

    // Save the file if it exists that is currently in the draft area.
    require_once($CFG->dirroot . '/lib/filelib.php');
    file_save_draft_area_files($draftitemid, $contextid, 'mod_customcert', $filearea, 0);
}

/**
 * Return the list of possible fonts to use.
 */
public static function get_fonts() {
    global $CFG;

    require_once($CFG->libdir . '/pdf/lib.php');

    $arrfonts = [];
    $pdf = new \pdf();

```

```

$fontfamilies = $pdf->get_font_families();
foreach ($fontfamilies as $fontfamily => $fontstyles) {
    foreach ($fontstyles as $fontstyle) {
        $fontstyle = strtolower($fontstyle);
        if ($fontstyle == 'r') {
            $filenamewoextension = $fontfamily;
        } else {
            $filenamewoextension = $fontfamily . $fontstyle;
        }
        $fullpath = \TCPDF_FONTS::_getfontpath() . $filenamewoextension;
        // Set the name of the font to null, the include next should then set this
        // value, if it is not set then the file does not include the necessary data.
        $name = null;
        // Some files include a display name, the include next should then set this
        // value if it is present, if not then $name is used to create the display name.
        $displayname = null;
        // Some of the TCPDF files include files that are not present, so we have to
        // suppress warnings, this is the TCPDF libraries fault, grrr.
        @include($fullpath . '.php');
        // If no $name variable in file, skip it.
        if (is_null($name)) {
            continue;
        }
        // Check if there is no display name to use.
        if (is_null($displayname)) {
            // Format the font name, so "FontName-Style" becomes "Font Name - Style".
            $displayname = preg_replace("/([a-z])([A-Z])/", "$1 $2", $name);
            $displayname = preg_replace("/([a-zA-Z])-([a-zA-Z])/", "$1 - $2", $displayname);
        }

        $arrfonts[$filenamewoextension] = $displayname;
    }
}
ksort($arrfonts);

return $arrfonts;
}

```

```

/**
 * Return the list of possible font sizes to use.
 */
public static function get_font_sizes() {
    // Array to store the sizes.
    $sizes = array();

    for ($i = 1; $i <= 200; $i++) {
        $sizes[$i] = $i;
    }

    return $sizes;
}

/**
 * Get the time the user has spent in the course.
 *
 * @param int $courseid
 * @param int $userid
 * @return int the total time spent in seconds
 */
public static function get_course_time($courseid, $userid = 0) {
    global $CFG, $DB, $USER;

    if (empty($userid)) {
        $userid = $USER->id;
    }

    $logmanager = get_log_manager();
    $readers = $logmanager->get_readers();
    $enabledreaders = get_config('tool_log', 'enabled_stores');
    if (empty($enabledreaders)) {
        return 0;
    }
    $enabledreaders = explode(',', $enabledreaders);

    // Go through all the readers until we find one that we can use.
    foreach ($enabledreaders as $enabledreader) {

```

```

$reader = $readers[$enabledreader];
if ($reader instanceof \logstore_legacy\log\store) {
    $logtable = 'log';
    $coursefield = 'course';
    $timefield = 'time';
    break;
} else if ($reader instanceof \core\log\sql_internal_table_reader) {
    $logtable = $reader->get_internal_log_table_name();
    $coursefield = 'courseid';
    $timefield = 'timecreated';
    break;
}
}

// If we didn't find a reader then return 0.
if (!isset($logtable)) {
    return 0;
}

$sql = "SELECT id, $timefield
        FROM { {$logtable} }
        WHERE userid = :userid
        AND $coursefield = :courseid
        ORDER BY $timefield ASC";
$params = array('userid' => $userid, 'courseid' => $courseid);
$totaltime = 0;
if ($logs = $DB->get_recordset_sql($sql, $params)) {
    foreach ($logs as $log) {
        if (!isset($login)) {
            // For the first time $login is not set so the first log is also the first login.
            $login = $log->$timefield;
            $lasthit = $log->$timefield;
            $totaltime = 0;
        }
        $delay = $log->$timefield - $lasthit;
        if ($delay > ($CFG->sessiontimeout * 60)) {
            // The difference between the last log and the current log is more than
            // the timeout Register session value so that we have found a session!

```

```

        $login = $log->$timefield;
    } else {
        $totaltime += $delay;
    }
    // Now the actual log became the previous log for the next cycle.
    $lasthit = $log->$timefield;
}

return $totaltime;
}

return 0;
}

/**
 * Returns a list of issued customcerts.
 *
 * @param int $customcertid
 * @param bool $groupmode are we in group mode
 * @param \stdClass $cm the course module
 * @param int $limitfrom
 * @param int $limitnum
 * @param string $sort
 * @return array the users
 */
public static function get_issues($customcertid, $groupmode, $cm, $limitfrom, $limitnum, $sort = "") {
    global $DB;

    // Get the conditional SQL.
    list($conditionssql, $conditionsparams) = self::get_conditional_issues_sql($cm, $groupmode);

    // If it is empty then return an empty array.
    if (empty($conditionsparams)) {
        return array();
    }

    // Add the conditional SQL and the customcertid to form all used parameters.
    $allparams = $conditionsparams + array('customcertid' => $customcertid);

```



```

// Return the issues.
$extrafields = get_extra_user_fields(\context_module::instance($cm->id));
$ufields = \user_picture::fields('u', $extrafields);
$sql = "SELECT $ufields, ci.id as issueid, ci.code, ci.timecreated
        FROM {user} u
        INNER JOIN {customcert_issues} ci
            ON u.id = ci.userid
        WHERE u.deleted = 0
        AND ci.customcertid = :customcertid
        $conditionssql";
if ($sort) {
    $sql .= "ORDER BY " . $sort;
} else {
    $sql .= "ORDER BY " . $DB->sql_fullname();
}

return $DB->get_records_sql($sql, $allparams, $limitfrom, $limitnum);
}

/**
 * Returns the total number of issues for a given customcert.
 *
 * @param int $customcertid
 * @param \stdClass $cm the course module
 * @param bool $groupmode the group mode
 * @return int the number of issues
 */
public static function get_number_of_issues($customcertid, $cm, $groupmode) {
    global $DB;

    // Get the conditional SQL.
    list($conditionssql, $conditionsparams) = self::get_conditional_issues_sql($cm, $groupmode);

    // If it is empty then return 0.
    if (empty($conditionsparams)) {
        return 0;
    }
}

```

```

// Add the conditional SQL and the customcertid to form all used parameters.
$allparams = $conditionsparams + array('customcertid' => $customcertid);

// Return the number of issues.
$sql = "SELECT COUNT(u.id) as count
      FROM {user} u
      INNER JOIN {customcert_issues} ci
      ON u.id = ci.userid
      WHERE u.deleted = 0
      AND ci.customcertid = :customcertid
      $conditionssql";
return $DB->count_records_sql($sql, $allparams);
}

/**
 * Returns an array of the conditional variables to use in the get_issues SQL query.
 *
 * @param \stdClass $cm the course module
 * @param bool $groupmode are we in group mode ?
 * @return array the conditional variables
 */
public static function get_conditional_issues_sql($cm, $groupmode) {
    global $DB, $USER;

    // Get all users that can manage this customcert to exclude them from the report.
    $context = \context_module::instance($cm->id);
    $conditionssql = "";
    $conditionsparams = array();

    // Get all users that can manage this certificate to exclude them from the report.
    $certmanagers = array_keys(get_users_by_capability($context, 'mod/customcert:manage', 'u.id'));
    $certmanagers = array_merge($certmanagers, array_keys(get_admins()));
    list($sql, $params) = $DB->get_in_or_equal($certmanagers, SQL_PARAMS_NAMED, 'cert');
    $conditionssql .= "AND NOT u.id $sql \n";
    $conditionsparams += $params;

    if ($groupmode) {
        $canaccessallgroups = has_capability('moodle/site:accessallgroups', $context);

```

```

$currentgroup = groups_get_activity_group($cm);

// If we are viewing all participants and the user does not have access to all groups then return
nothing.
if (!$currentgroup && !$canaccessallgroups) {
    return array("", array());
}

if ($currentgroup) {
    if (!$canaccessallgroups) {
        // Guest users do not belong to any groups.
        if (isguestuser()) {
            return array("", array());
        }

        // Check that the user belongs to the group we are viewing.
        $usersgroups = groups_get_all_groups($cm->course, $USER->id, $cm->groupingid);
        if ($usersgroups) {
            if (!isset($usersgroups[$currentgroup])) {
                return array("", array());
            }
        } else { // They belong to no group, so return an empty array.
            return array("", array());
        }
    }
}

$groupusers = array_keys(groups_get_members($currentgroup, 'u.*'));
if (empty($groupusers)) {
    return array("", array());
}

list($sql, $params) = $DB->get_in_or_equal($groupusers, SQL_PARAMS_NAMED, 'grp');
$conditionssql .= "AND u.id $sql ";
$conditionsparams += $params;
}
}

return array($conditionssql, $conditionsparams);

```

```

}

/**
 * Get number of certificates for a user.
 *
 * @param int $userid
 * @return int
 */
public static function get_number_of_certificates_for_user($userid) {
    global $DB;

    $sql = "SELECT COUNT(*)
            FROM {customcert} c
            INNER JOIN {customcert_issues} ci
            ON c.id = ci.customcertid
            WHERE ci.userid = :userid";
    return $DB->count_records_sql($sql, array('userid' => $userid));
}

/**
 * Gets the certificates for the user.
 *
 * @param int $userid
 * @param int $limitfrom
 * @param int $limitnum
 * @param string $sort
 * @return array
 */
public static function get_certificates_for_user($userid, $limitfrom, $limitnum, $sort = "") {
    global $DB;

    if (empty($sort)) {
        $sort = 'ci.timecreated DESC';
    }

    $sql = "SELECT c.id, c.name, co.fullname as coursename, ci.code, ci.timecreated
            FROM {customcert} c
            INNER JOIN {customcert_issues} ci

```

```

        ON c.id = ci.customcertid
    INNER JOIN {course} co
        ON c.course = co.id
    WHERE ci.userid = :userid
    ORDER BY $sort";
return $DB->get_records_sql($sql, array('userid' => $userid), $limitfrom, $limitnum);
}

/**
 * Issues a certificate to a user.
 *
 * @param int $certificateid The ID of the certificate
 * @param int $userid The ID of the user to issue the certificate to
 * @return int The ID of the issue
 */
public static function issue_certificate($certificateid, $userid) {
    global $DB;

    $issue = new \stdClass();
    $issue->userid = $userid;
    $issue->customcertid = $certificateid;
    $issue->code = self::generate_code();
    $issue->emailed = 0;
    $issue->timecreated = time();

    // Insert the record into the database.
    return $DB->insert_record('customcert_issues', $issue);
}

/**
 * Generates a 10-digit code of random letters and numbers.
 *
 * @return string
 */
public static function generate_code() {
    global $DB;

    $uniquecodefound = false;

```

```
$code = random_string(10);  
while (!$uniquecodefound) {  
    if (!$DB->record_exists('customcert_issues', array('code' => $code))) {  
        $uniquecodefound = true;  
    } else {  
        $code = random_string(10);  
    }  
}  
  
return $code;  
}  
}
```

ДОДАТОК Б

Тести

```

class mod_customcert_element_helper_testcase extends advanced_testcase {

    /**
     * Test set up.
     */
    public function setUp() {
        $this->resetAfterTest();
    }

    /**
     * Tests we are returning the correct course id for an element in a course customcert activity.
     */
    public function test_get_courseid_element_in_course_certificate() {
        global $DB;

        // Create a course.
        $course = $this->getDataGenerator()->create_course();

        // Create a custom certificate in the course.
        $customcert = $this->getDataGenerator()->create_module('customcert', array('course' => $course->id));

        // Get the template to add elements to.
        $template = $DB->get_record('customcert_templates', array('contextid' =>
context_module::instance($customcert->cmid)->id));
        $template = new \mod_customcert\template($template);

        // Add a page to the template.
        $pageid = $template->add_page();

        // Add an element to this page.
        $element = new \stdClass();
        $element->name = 'Test element';
        $element->element = 'testelement';
    }
}

```

```

$element->pageid = $pageid;
$element->sequence = \mod_customcert\element_helper::get_element_sequence($element->pageid);
$element->timecreated = time();
$element->id = $DB->insert_record('customcert_elements', $element);

// Confirm the correct course id is returned.
$this->assertEquals($course->id, \mod_customcert\element_helper::get_courseid($element->id));
}

/**
 * Tests we are returning the correct course id for an element in a site template.
 */
public function test_get_courseid_element_in_site_template() {
    global $DB, $SITE;

    // Add a template to the site.
    $template = \mod_customcert\template::create('Site template', context_system::instance()->id);

    // Add a page to the template.
    $pageid = $template->add_page();

    // Add an element to this page.
    $element = new \stdClass();
    $element->name = 'Test element';
    $element->element = 'testelement';
    $element->pageid = $pageid;
    $element->sequence = \mod_customcert\element_helper::get_element_sequence($element->pageid);
    $element->timecreated = time();
    $element->id = $DB->insert_record('customcert_elements', $element);

    // Confirm the correct course id is returned.
    $this->assertEquals($SITE->id, \mod_customcert\element_helper::get_courseid($element->id));
}

/**
 * Tests we are returning the correct course module id for an element in a course customcert activity.
 */
public function test_get_context_element_in_course_certificate() {

```



```

global $DB;

// Create a course.
$course = $this->getDataGenerator()->create_course();

// Create a custom certificate in the course.
$customcert = $this->getDataGenerator()->create_module('customcert', array('course' => $course->id));

// Get the template to add elements to.
$template = $DB->get_record('customcert_templates', array('contextid' =>
context_module::instance($customcert->cmid)->id));
$template = new \mod_customcert\template($template);

// Add a page to the template.
$pageid = $template->add_page();

// Add an element to this page.
$element = new \stdClass();
$element->name = 'Test element';
$element->element = 'testelement';
$element->pageid = $pageid;
$element->sequence = \mod_customcert\element_helper::get_element_sequence($element->pageid);
$element->timecreated = time();
$element->id = $DB->insert_record('customcert_elements', $element);

// Confirm the correct course module id is returned.
$this->assertEquals(context_module::instance($customcert->cmid),
    \mod_customcert\element_helper::get_context($element->id));
}

/**
 * Tests we are returning the correct course module id for an element in a site template.
 */
public function test_get_context_element_in_site_template() {
    global $DB;

    // Add a template to the site.
    $template = \mod_customcert\template::create('Site template', context_system::instance()->id);

```

```

// Add a page to the template.
$pageid = $template->add_page();

// Add an element to this page.
$element = new \stdClass();
$element->name = 'Test element';
$element->element = 'testelement';
$element->pageid = $pageid;
$element->sequence = \mod_customcert\element_helper::get_element_sequence($element->pageid);
$element->timecreated = time();
$element->id = $DB->insert_record('customcert_elements', $element);

// Confirm the correct course module id is returned.
$this->assertEquals(context_system::instance(),
\mod_customcert\element_helper::get_context($element->id));
}

/**
 * Test we return the correct grade items in a course.
 */
public function test_get_grade_items() {
    global $DB;

    // Create a course.
    $course = $this->getDataGenerator()->create_course();

    // Create a few gradeable items.
    $assign1 = $this->getDataGenerator()->create_module('assign', array('course' => $course->id));
    $assign2 = $this->getDataGenerator()->create_module('assign', array('course' => $course->id));
    $assign3 = $this->getDataGenerator()->create_module('assign', array('course' => $course->id));

    // Create a manual grade item.
    $gi = $this->getDataGenerator()->create_grade_item(['courseid' => $course->id]);

    // Create a category grade item.
    $gc = $this->getDataGenerator()->create_grade_category(['courseid' => $course->id]);
    $gc = $DB->get_record('grade_items', ['itemtype' => 'category', 'iteminstance' => $gc->id]);

```

```

// Create an item attached to an outcome.
$outcome = $this->getDataGenerator()->create_grade_outcome(['courseid' => $course->id, 'shortname'
=> 'outcome']);
$go = $this->getDataGenerator()->create_grade_item(
    [
        'courseid' => $course->id,
        'outcomeid' => $outcome->id
    ]
);

// Confirm the function returns the correct number of grade items.
$gradeitems = \mod_customcert\element_helper::get_grade_items($course);
$this->assertCount(6, $gradeitems);
$this->assertArrayHasKey($assign1->cmid, $gradeitems);
$this->assertArrayHasKey($assign2->cmid, $gradeitems);
$this->assertArrayHasKey($assign3->cmid, $gradeitems);
$this->assertArrayHasKey('gradeitem:' . $gi->id, $gradeitems);
$this->assertArrayHasKey('gradeitem:' . $gc->id, $gradeitems);
$this->assertArrayHasKey('gradeitem:' . $go->id, $gradeitems);
}

/**
 * Test we return the correct grade information for an activity.
 */
public function test_get_mod_grade_info() {
    // Create a course.
    $course = $this->getDataGenerator()->create_course();

    // Set that we want 3 decimals to display.
    grade_set_setting($course->id, 'decimalpoints', 3);

    // Create two users.
    $student1 = $this->getDataGenerator()->create_user();
    $student2 = $this->getDataGenerator()->create_user();

    // Enrol them into the course.
    $this->getDataGenerator()->enrol_user($student1->id, $course->id);
    $this->getDataGenerator()->enrol_user($student2->id, $course->id);

```

```

// Create a gradeable item.
$assign = $this->getDataGenerator()->create_module('assign', array('course' => $course->id));

// Give a grade to the student.
$gi = grade_item::fetch(
    [
        'itemtype' => 'mod',
        'itemmodule' => 'assign',
        'iteminstance' => $assign->id,
        'courseid' => $course->id
    ]
);
$datagrade = 50;
$time = time();
$grade = new grade_grade();
$grade->itemid = $gi->id;
$grade->userid = $student1->id;
$grade->rawgrade = $datagrade;
$grade->finalgrade = $datagrade;
$grade->rawgrademax = 100;
$grade->rawgrademin = 0;
$grade->timecreated = $time;
$grade->timemodified = $time;
$grade->insert();

// Check that the user received the grade.
$grade = \mod_customcert\element_helper::get_mod_grade_info(
    $assign->cmid,
    GRADE_DISPLAY_TYPE_PERCENTAGE,
    $student1->id
);

$this->assertEquals($assign->name, $grade->get_name());
$this->assertEquals('50.00000', $grade->get_grade());
$this->assertEquals('50.000 %', $grade->get_displaygrade());
$this->assertEquals($time, $grade->get_dategraded());

// Check that the user we did not grade has no grade.

```

```

$grade = \mod_customcert\element_helper::get_mod_grade_info(
    $assign->cmid,
    GRADE_DISPLAY_TYPE_PERCENTAGE,
    $student2->id
);
$this->assertEquals($assign->name, $grade->get_name());
$this->assertEquals(null, $grade->get_grade());
$this->assertEquals('-', $grade->get_displaygrade());
$this->assertEquals(null, $grade->get_dategraded());

grade_get_setting($course->id, null, null, true);
}

/**
 * Test we return the correct grade information for a course.
 */
public function test_get_course_grade_info() {
    // Create a course.
    $course = $this->getDataGenerator()->create_course();

    // Set that we want 3 decimals to display.
    grade_set_setting($course->id, 'decimalpoints', 3);

    // Create two users.
    $student1 = $this->getDataGenerator()->create_user();
    $student2 = $this->getDataGenerator()->create_user();

    // Enrol them into the course.
    $this->getDataGenerator()->enrol_user($student1->id, $course->id);
    $this->getDataGenerator()->enrol_user($student2->id, $course->id);

    // Get the course item.
    $coursegradeitem = grade_item::fetch_course_item($course->id);

    $datagrade = 50;
    $time = time();
    $grade = new grade_grade();
    $grade->itemid = $coursegradeitem->id;

```

```

$grade->userid = $student1->id;
$grade->rawgrade = $datagrade;
$grade->finalgrade = $datagrade;
$grade->rawgrademax = 100;
$grade->rawgrademin = 0;
$grade->timecreated = $time;
$grade->timemodified = $time;
$grade->insert();

// Check that the user received the grade.
$grade = \mod_customcert\element_helper::get_course_grade_info(
    $course->id,
    GRADE_DISPLAY_TYPE_PERCENTAGE,
    $student1->id
);

$this->assertEquals(get_string('coursetotal', 'grades'), $grade->get_name());
$this->assertEquals('50.00000', $grade->get_grade());
$this->assertEquals('50.000 %', $grade->get_displaygrade());
$this->assertEquals($time, $grade->get_dategraded());

// Check that the user we did not grade has no grade.
$grade = \mod_customcert\element_helper::get_course_grade_info(
    $course->id,
    GRADE_DISPLAY_TYPE_PERCENTAGE,
    $student2->id
);

$this->assertEquals(get_string('coursetotal', 'grades'), $grade->get_name());
$this->assertEquals(null, $grade->get_grade());
$this->assertEquals('-', $grade->get_displaygrade());
$this->assertEquals(null, $grade->get_dategraded());

grade_get_setting($course->id, null, null, true);
}

/**
 * Test we return the correct grade information for a grade item.
 */

```

```

public function test_get_grade_item_info() {
    // Create a course.
    $course = $this->getDataGenerator()->create_course();

    // Set that we want 3 decimals to display.
    grade_set_setting($course->id, 'decimalpoints', 3);

    // Create two users.
    $student1 = $this->getDataGenerator()->create_user();
    $student2 = $this->getDataGenerator()->create_user();

    // Enrol them into the course.
    $this->getDataGenerator()->enrol_user($student1->id, $course->id);
    $this->getDataGenerator()->enrol_user($student2->id, $course->id);

    // Create a manual grade item.
    $gi = $this->getDataGenerator()->create_grade_item(['itemname' => 'Grade item yo', 'courseid' =>
$course->id]);

    // Give a grade to the student.
    $gi = grade_item::fetch(['id' => $gi->id]);
    $datagrade = 50;
    $time = time();
    $grade = new grade_grade();
    $grade->itemid = $gi->id;
    $grade->userid = $student1->id;
    $grade->rawgrade = $datagrade;
    $grade->finalgrade = $datagrade;
    $grade->rawgrademax = 100;
    $grade->rawgrademin = 0;
    $grade->timecreated = $time;
    $grade->timemodified = $time;
    $grade->insert();

    // Check that the user received the grade.
    $grade = \mod_customcert\element_helper::get_grade_item_info(
        $gi->id,
        GRADE_DISPLAY_TYPE_PERCENTAGE,

```

```
    $student1->id
);

$this->assertEquals('Grade item yo', $grade->get_name());
$this->assertEquals('50.00000', $grade->get_grade());
$this->assertEquals('50.000 %', $grade->get_displaygrade());
$this->assertEquals($time, $grade->get_dategraded());

// Check that the user we did not grade has no grade.
$grade = \mod_customcert\element_helper::get_grade_item_info(
    $gi->id,
    GRADE_DISPLAY_TYPE_PERCENTAGE,
    $student2->id
);
$this->assertEquals('Grade item yo', $grade->get_name());
$this->assertEquals(null, $grade->get_grade());
$this->assertEquals('-', $grade->get_displaygrade());
$this->assertEquals(null, $grade->get_dategraded());

grade_get_setting($course->id, null, null, true);
}
}
```


ДОДАТОК В

Посилання на Git

URL: https://bitbucket.org/s_var_og/moodlecert/src/master/.